المعالجات الصغرية
والبرمجة بلغة الأسمبلي

وقل رب زدني علما
جامعة دمشق
Damascus University

وقل رب زدني علما
جامعة دمشق
Damascus University

منشورات جامعة دمشق
كلية الهندسة
الميكانيكية والكهربائية



المعالجات الصغرية والبرمجة بلغة الأسمبلي

الدكتور أحمد خضور
أستاذ مساعد في قسم الهندسة الطبية

جامعة دمشق
١٤٣٥ ـ ١٤٣٦ هـ
٢٠١٤ ـ ٢٠١٥ م

وقل رب زدني علما
جامعة دمشق
Damascus University

الفهرس	الصفحة

مقدمة الكتاب	13
الجزء الأول: مقدمة عامة	15
الفصل الأول: الإشارات الحيوية	19
١-١ أنواع الإشارات الحيوية	19
١-٢ اقتباس (تحصيل) الإشارات الحيوية	21
١-٣ قياس الإشارات الحيوية	27
١-٤ تجهيزات قياس الإشارات الحيوية	30
الفصل الثاني: الحاسوب الرقمي	33
٢-١ طرق الولوج للحاسوب	36
٢-٢ التركيبة العامة للحواسيب الصغرية	37
٢-٣ مفهوم الذاكرة	40
٢-٣-١ أنواع الذواكر	42
٢-٣-٢ زيادة طول كلمة الذاكرة	47
٢-٣-٣ زيادة حجم الذاكرة	48
٢-٣-٤ تصميم كتلة ذاكرة	50
٢-٣-٥ الفراغ في المجال العنوني	56
٢-٤ المعالجات الصغرية	59
٢-٤-١ تطور المعالجات الصغرية	61
٢-٤-٢ أنماط عمل المعالجات	63
٢-٤-٣ قولطية المعالج	63
٢-٤-٤ معمارية المعالج	64

٢-٤-٥ تبادل البيانات بين المعالج والأجزاء الأخرى	65
٢-٤-٦ معالجات RISC و CISC	66
٢-٤-٧ المعالجات متعددة النوى	67
٢-٤-٨ برمجة المعالجات	68
٢-٥ أسئلة متعددة الاجابات	69
الفصل الثالث: محولات الإشارات والربط البيني مع الحاسوب	71
٣-١ الخصائص العامة للمضخم العملياتي ودارته الأساسية	72
٣-٢ الحساسات والمبدلات	80
٣-٣ محولات الإشارة	83
٣-٣-١ محولات الإشارة الرقمية التشابهية D/A	84
٣-٣-٢ ربط المحول D/A مع الحواسيب الصغرية	87
٣-٣-٣ محولات الإشارة التشابهية الرقمية A/D	88
٣-٣-٣-١ طرق تحويل الإشارة التمثيلية إلى قيمة رقمية	89
٣-٣-٤ ربط المحولات A/D مع الحواسيب الصغرية	91
الجزء الثاني: المعالجات الصغرية	93
الفصل الرابع: المعالج الصغري Intel 8080A	99
٤-١ التركيبة الداخلية للمعالج I8080A	99
٤-٢ التركيبة الخارجية للمعالج I8080A	103
٤-٣ التزامن لدى المعالج I8080A	106
٤-٤ صيغ المعطيات والتعليمات لدى المعالج 8080A	108
الفصل الخامس: دارات الدعم غير المبرمجة للمعالج 8080A	111
٥-١ الدارة 8224 (Clock generator)	115
٥-٢ الدارة 8228/8238 (System controller and bus driver)	117

٥-٣ الدارة 8205 (High Speed 1 out 8 Binary Decoder) 118
٥-٤ الدارة 8212 (8 bit I/O port) 119
٥-٥ الدارة 8216/8226 (4-bit Bidirectional Buffer/Latch) 120
٥-٦ الدارة 8282/8283 (Octal Latch) 121
٥-٧ الدارة 8286/8287 (Octal Bus Transceiver) 122
٥-٨ الدارة 8214 (Interrupt Controller) 123
٥-٩ أمثلة حول تصميم أنظمة مايكرو كومبيوتر باستخدام المعالج 8080A 124
٥-١٠ أسئلة متعددة الاجابات 134

الفصل السادس: دارات الدعم المبرمجة للمعالج 8080A 135
٦-١ النقل المتوازي للمعطيات 135
٦-١-١ دارة الدعم المبرمجة 8255A للنقل التفرعي للمعطيات 138
٦-٢ النقل التسلسلي للمعطيات 147
٦-٢-١ دارة الدعم المبرمجة 8251A للنقل التسلسلي للمعطيات 150
٦-٣ دارة الدعم المبرمجة 8253/8254 للتوقيت/العد 159

الفصل السابع: المعالج الصغري 8085A 165
٧-١ التركيبة الداخلية للمعالج 8085A 165
٧-٢ إشارات المعالج 8085A 168
٧-٣ دارات الدعم الخاصة بالمعالج 8085A 171
٧-٣-١ الدارة 8155/8156 172
٧-٣-٢ الدارة 8355 178
٧-٣-٣ الدارة 8755 181
٧-٤ أمثلة حول تصميم أنظمة مايكرو كومبيوتر باستخدام المعالج 8085A 183
٧-٥ أسئلة متعددة الاجابات 192

الفصل الثامن: معالجات 16 bits Intel 195

٨-١ العائلة 8086 Family 195
٨-٢ الأنظمة متعددة المعالجات 196
٨-٣ المعالج الصغري 8086/8088 199
٨-٣-١ أشكال (صيغ) عمل المعالج الصغري 8086 199
٨-٣-١-١ متحكم الممر 8288 200
٨-٣-٢ التركيبة الداخلية للمعالج 8086 201
٨-٣-٢-١ وحدة التنفيذ EU 203
٨-٣-٢-٢ وحدة ربط الممر BIU 206
٨-٣-٣ إشارات المعالج الصغري 8086 207
٨-٣-٤ مقاطعات المعالج الصغري 8086 215
٨-٤ تقطيع الذاكرة 216
٨-٤-١ حساب العنوان الفيزيائي 218
٨-٥ نظام التزامن لدى المعالج الصغري 8086 219
٨-٦ اسئلة متعددة الاجابات 222
الفصل التاسع: وحدات متممة للمعالجات الصغرية 225
٩-١ المقاطعات Interrupts 225
٩-٢ أنماط عمل المعالج الصغري 8086 227
٩-٣ وحدة الفاصلة العشرية 229
٩-٤ وحدة الذاكرة المخبئية 232
٩-٥ وحدة نقل المعطيات بالولوج الذاكري المباشر 236
٩-٦ أسئلة متعددة الاجابات 238

الفصل العاشر: دارات الدعم للمعالج 8086 239
١٠-١ مولد النبضات الساعية 8284A 239
١٠-٢ المتحكم بأولوية المقاطعة 8259A 242
١٠-٣ المتحكم بالـ DMA الدارة Intel 8237 Controller 244
١٠-٤ المعالج الحسابي 8087 247
١٠-٥ محكم الممر Bipolar Bus Arbiter 8289 250
الفصل الحادي عشر: تصميم مايكرو كمبيوتر باستخدام 8086 Family 253
١١-١ تقنيات ترميز عناوين الذاكرة 255
١١-٢ ارتباط تصميم الحاسوب الصغري بنمط عمل المعالج 260
١١-٢-١ تصميم منظومة حاسوب صغري باستخدام المعالج 8086 بالنموذج الاصغري 261
١١-٢-٢ تصميم منظومة حاسوب صغري باستخدام المعالج 8086 بالنموذج الأعظمي 266
١١-٣ أسئلة متعددة الاجابات 274
الفصل الثاني عشر: معالجات صغرية بمحيطيات متكاملة 275
١٢-١ المعالج الصغري 80186/80188 275
١٢-٢ المعالج الصغري Intel 80286 282
١٢-٢-١ مفهوم النمط الحقيقي للمعالج 286 287
١٢-٢-٢ مفهوم النمط المحمي للمعالج 286 288
الفصل الثالث عشر: معالجات صغرية 32 Bits 289
١٣-١ المعالج الصغري 80386 289
١٣-١-١ النمط الحقيقي للمعالج 386 296

١٣-١-٢ النمط المحمي للمعالج 386	297
١٣-٢ المعالج الصغري 80486	298
١٣-٣ المعالج الصغري 80586	302
الفصل الرابع عشر: معالجات Pentium	303
١٤-١ تطور معالجات Pentium	303
١٤-٢ المعماريات المصغرة المستخدمة بمعالجات Pentium	309
١٤-٣ التركيبة الداخلية لمعالج Pentium الأساسي	311
١٤-٤ إشارات معالج Pentium الأساسي	315
١٤-٥ إصدارات معالج Pentium	320
١٤-٦ مقارنة بين المعالجات	323
الجزء الثالث: البرمجة بلغة الأسمبلي	327
الفصل الخامس عشر: المفاهيم الأساسية للبرمجة بلغة الأسمبلي	331
١٥-١ أنماط العنونة لدى المعالج 8086	333
١٥-٢ تركيب شيفرة الآلة لتعليمات المعالج 8086	335
١٥-٣ توجيهات تسمية العناوين والمعطيات	338
١٥-٤ تعليمات المعالج 8086	346
١٥-٥ اسئلة متعددة الاجابات	354
الفصل السادس عشر: توصيف تعليمات نقل المعطيات والتعليمات الحسابية والمنطقية	355
١٦-١ التعليمات الحسابية	355
١٦-١-١ تعليق حول التعليمات الحسابية للمعالج 8086	361
١٦-٢ تعليمات نقل المعطيات	362
١٦-٣ تعليمات معالجة البت	366

الفصل السابع عشر: توصيف تعليمات نقل البرنامج والتحكم بالمعالج 373

١٧–١ تعليمات النقل غير الشرطية 376

١٧–٢ تعليمات النقل الشرطية 377

١٧–٣ تعليمات المقاطعة 384

١٧–٣–١ المقاطعة 21H 385

١٧–٣–٢ المقاطعة 10H 387

١٧–٤ تعليمات التحكم التكرارية 392

الفصل الثامن عشر: البرمجة البنوية وأدوات تطوير برامج لغة الأسمبلي 395

١٨–١ تنفيذ بعض البنى البرمجية الأساسية بلغة الأسمبلي 398

١٨–٢ الإجراءات بلغة الأسمبلي 402

١٨–٢–١ المكدس في المعالج 8086 404

١٨–٣ الماكرو Macro 405

١٨–٤ تعليمات التحكم للمعالج 8086 407

١٨–٥ أدوات تطوير برامج لغة الأسمبلي 410

١٨–٦ معالجة المعطيات 413

١٨–٦–١ تعليمات السلاسل 413

١٨–٦–٢ التحويل بين أشكال مختلفة للمعطيات 416

١٨–٧ أسئلة متعددة الاجابات 421

الجزء الرابع: الأنظمة المدمجة 423

الفصل التاسع عشر: المتحكمات الصغرية 431

١٩–١ المتحكم Intel 8051 434

١٩–١–١ معمارية الذاكرة في المتحكم 8051 443

١٩–١–٢ تنظيم الذاكرة في المتحكم 8051 444

١٩-٢ متحكمات AVR	449
١٩-٢-١ المتحكم ATmega16	450
١٩-٢-١-١ ميزات المتحكم ATmega16	452
١٩-٢-١-٢ تركيبة المتحكم ATmega16	453
١٩-٣ متحكمات PIC	463
١٩-٣-١ المتحكم PIC 16F84A	464
١٩-٣-١-١ مزايا المتحكم PIC 16F84A	466
١٩-٣-١-٢ مسجلات المتحكم PIC16F84A	467
١٩-٣-٢ المتحكم PIC16F877A/874A	472
١٩-٣-٢-١ تنظيم الذاكرة لدى المتحكم PIC16F877A	477
١٩-٤ تطبيقات المتحكمات	479
الفصل العشرون: برمجة المتحكمات	483
٢٠-١ برمجة المتحكمات الصغرية بلغة عالية المستوى	489
الجزء الخامس: القسم العملي	499
المراجع	525
ملحق١: مصطلحات علمية	527
ملحق ٢: إشارات المعالج Pentium 510/60, 567/66	537
ملحق٣: قائمة بالأشكال الواردة في الكتاب	539
ملحق٤: قائمة بالجداول الواردة في الكتاب	547

Damascus University

مقدمة الكتاب

نتعرف بهذا الكتاب على المعالجات الصغرية التي أصبحنا نصادفها في كل مجال من مجالات الحياة. إن مقرري الدارات المنطقية والمعالجات الصغرية يشكلان الأساس للتعريف بالبنية العامة للحواسيب الصغرية.

بدأت المعالجات الصغرية بالظهور مع مطلع العقد السابع من القرن الماضي، وأخذت تتطور سريعاً لدرجة أن ما هو جديد اليوم في هذا المجال يصبح بعد عام على الأكثر قديم، لذلك سنكتفي بالإشارة إلى المفاهيم الأساسية المتعلقة بالمعالجات الصغرية واستخداماتها في الحياة العملية.

وبالنظر إلى ترافق ظهور المتحكمات الصغرية مع المعالجات الصغرية، سنحاول في هذا الكتاب الإشارة أيضاً إلى مفهوم الأنظمة المدمجة التي تعتمد في تركيبتها العامة على المتحكمات الصغرية.

بالطبع لا يهمنا من دراسة المعالجات والمتحكمات الصغرية معرفة كيفية تصنيعها، وإنما معرفة استخدام ما هو مصنع منها وصولا إلى التعريف بالمكونات الأساسية للحاسوب الصغري (الشخصي)، وهذا يتطلب التعريف ببعض دارات الدعم للمعالجات الصغرية حتى نتعرف كيفية وصلها مع ما يلزم من مكونات أخرى متممة لبناء منظومة متكاملة تصلح لأمور المعالجة والتحكم.

ولتكملة التعريف بالتركيبة العامة للحواسيب الصغرية وآلية عملها في قيادة بعض المنظومات، يلزم التعريف أيضاً ببعض الجوانب المتعلقة ببرمجة المعالجات باللغات المناسبة لتكوين تصور متكامل عن استخدامها في مختلف التطبيقات.

يركّز هذا الكتاب الموجه لطلاب الهندسة الطبية بجامعة دمشق على الجانب التطبيقي لاطلاع الطلبة على كيفية تصميم منظومة حاسوب صغري، وسيتم عرض الموضوعات بشكل يجعل من هذا الكتاب مساعد للمهتم في مجال المعالجات والمتحكمات الصغرية.

قسم الكتاب إلى خمسة أجزاء، وكل جزء يضم فصلين أو أكثر، بحيث إن العدد الإجمالي لفصول الكتاب وصل إلى عشرين فصلا، لتغطي مختلف الجوانب النظرية والعملية المتعلقة بالمعالجات والمتحكمات الصغرية.

يمثل الجزء الأول المكون من ثلاثة فصول مقدمة عامة للتذكير ببعض المفاهيم الأساسية المتعلقة بالحاسوب الصغري وبواحد من المجالات المهمة لتطبيقاتها، بحيث إن الفصل الأول يذكّر ببعض المفاهيم المتعلقة بالإشارات الحيوية ومعالجتها.

أما الفصل الثاني فيذكّر بالتركيبة العامة للحاسوب الرقمي، وبعض المفاهيم المتعلقة به، وبمكوناته الأساسية وبالأخص الذاكرة لتعرف آلية تصميم كرت الذاكرة المناسب لربطه بالمعالج عند تشكيل حاسوب صغري.

ويذكّر الفصل الثالث ببعض المفاهيم والعناصر (الدارات) المتعلقة بتضخيم الإشارات الحيوية على مختلف أنواعها، والدارات الأساسية لمضخم العمليات، إضافة إلى بعض الأنواع من الحساسات والمبدلات ومحولات الإشارة بنوعيها من تمثيلي إلى رقمي A/D ومن رقمي إلى تمثيلي D/A.

ويدور الجزء الثاني من الكتاب حول بعض المعالجات الصغرية من شركة إنتل ابتداء من المعالجات بطول كلمة 8 bit مرورا بمعالجات بطول كلمة 16 bit، وصولا إلى المعالجات المستخدمة في الوقت الحاضر كالبنتيوم بإصداراته المختلفة.

وقد أتى هذا الجزء باثني عشر فصلا ليغطي مجمل المفاهيم المتعلقة بالمعالجات الصغرية وكيفية استخدامها لتصميم تركيبة محددة للحاسوب الصغري.

يغطي الجزء الثالث كيفية برمجة المعالجات الصغرية بلغة الأسمبلي وقد أفرد لهذا الجزء أربعة فصول للتعريف بكيفية كتابة برنامج متكامل بهذه اللغة.

أما الجزء الرابع فقد خصص لتغطية المفاهيم الأساسية المتعلقة بالمتحكمات الصغرية، وبخاصة المتحكمات بطول كلمة 8 bit من أهم الشركات المنتجة لها.

الجزء الخامس والأخير يضم مجموعة من الأمثلة حول كتابة بعض البرامج بلغة الأسمبلي لتغطية الجزء العملي للمعالجات الصغرية.

الجزء الأول: مقدمة عامة

ويضم ثلاثة فصول

الأول: الإشارات الحيوية

الثاني: الحاسوب الرقمي

الثالث: محولات الإشارات والربط البيني مع الحاسوب

Damascus University


وقل رب زدني علما
جامعة دمشق
Damascus University

ظهر المعالج الصغري Microprocessor نتيجة التطور التكنولوجي والحاجة لدارات الكترونية تكاملية قابلة للبرمجة والتعديل من قبل المستخدم. وقد ظهر مع كل معالج مجموعة من دارات الدعم Support Circuits المتممة له، حيث إن كل معالج يشكل مع جملة دارات الدعم الخاصة به ما يعرف بال Computer Set.

يتناول هذا الجزء بعض المفاهيم المتعلقة بالهندسة الطبية، تحديدا مفهوم الإشارات الحيوية، أنواعها الأساسية وأشكال قياسها وتجهيزات التقاطها كمقدمة عامة، للإشارة إلى دور المعالجات والمتحكمات الصغرية في هذا المجال، خاصة وان المهندسين الطبيين يعمدون إلى إيجاد الطرائق المناسبة لتحصيل الإشارات الحيوية لتحليلها وتقديمها بالشكل المناسب إلى الطبيب الذي يقع على عاتقه استنباط المعلومات الهامة والمفيدة منها.

يعد الإنسان المصدر الأساسي للإشارات الحيوية Bio-Signals التي تدل على حالته، ويمكن تعريف مجمل الإشارات الحيوية الصادرة من أعضاء وأنظمة مختلفة لجسم الإنسان بأنها تسجيل زمني أو مكاني لنشاط حيوي يتميز بمجموعة محددات، حيث تصاحب مجمل الأنشطة الحيوية تبدلات في الوسط التي تحدث فيها.

أتت موضوعات هذا الجزء التي تبدو للوهلة الأولى بعيدة عن موضوع الكتاب كمدخل للتعريف بواحد من أهم مجالات استخدام المعالجات والمتحكمات الصغرية في معالجة الإشارات الحيوية، وإظهار المعلومات التي تعكسها هذه الإشارات لاستخدامها من قبل العاملين في هذا المجال، خاصة وان الإشارات الحيوية تمثل بشكل عام مستودعاً من المعلومات عن جسم الإنسان محكم الإغلاق وأن المعالجات والمتحكمات الصغرية توفر مفاتيح الدخول إلى ذلك المستودع.

تتعدد المصادر التي تنشأ منها الإشارات الحيوية من الأعضاء المختلفة لجسم الإنسان مثل القلب ، الدماغ ، الرئتين ، الجلد وغيرها لتحمل في طياتها المعلومات الطبية المهمة والدالة على وظيفة تلك الأعضاء الناشئة منها، كما هو الحال في الفعالية الكهربائية للقلب، أو جريان الدم في الأوعية الدموية المختلفة، أو مرور الهواء من خلال المجاري التنفسية أو حركة القفص الصدري أثناء التنفس أو التراكيز الجزيئية للأوكسجين وثاني أكسيد الكاربون في الدم وغيرها.

يركز الفصل الأول من هذا الجزء بشكل أساسي على عرض بعض المفاهيم المتعلقة بالإشارات الحيوية، خاصة وأن التطور الهندسي والتقاني لتجهيزات تحصيل والتقاط الإشارات الحيوية ساعد في زيادة دقة قياس هذه الإشارات وإمكانية دراستها وتحليلها باستخدام المعالجات الصغرية.

وخصص الفصل الثاني للتعريف ببعض المكونات الأساسية للحاسوب الصغري كأداة أساسية في المعالجة والمراقبة، وقد تم فيه الإشارة إلى المفاهيم الأساسية المتعلقة بالمعالجات الصغرية كمكون أساسي بالنسبة لأي حاسوب وكعنصر أساسي في معظم التجهيزات الطبية الحديثة.

أما الفصل الثالث فقد خصص للتذكير ببعض العناصر التي يأتي استخدامها في منظومة ما قبل استخدام المعالج الصغري في تحصيل الإشارات من الوسط الخارجي وتهيئتها للشكل المناسب لإدخالها إلى الحاسوب بقصد المعالجة. وقد يستخدم البعض من هذه العناصر لإخراج نتائج المعالجة بالحاسوب بالشكل المناسب للاستفادة من المعلومات التي تنطوي عليها والتحكم بالجملة المقادة من خلالها.

الفصل الأول: الإشارات الحيوية

قد نتعامل في كثير من التطبيقات مع إشارات ضعيفة مثل الإشارات الطبية الحيوية Biomedical Signals الصادرة عن جسم الإنسان، بالتالي سنحاول في هذا الفصل التعرف باختصار على مفهوم الإشارات الحيوية (أنواعها، أشكال تحصيلها وطرق قياسها). تعدّ الإشارات الحيوية واحدة من الظواهر المهمة جداً التي يعتمد عليها في تحديد حالة الجسم الفيزيولوجية والسريرية مما يسهم بشكل أساسي في تشخيص العديد من الحالات المرضية التي تنتاب الإنسان.

يمكن اعتبار الإشارة الحيوية كل تسجيل للتبدلات التي تطرأ على العضو الصادرة عنه، وتقدم هذه التسجيلات معلومات مهمة للطبيب عن الحالة الفيزيولوجية التي صاحبتها. يتطلب الحصول على الإشارات الحيوية المرور بمجموعة من المراحل، تبدأ بعملية

*- التحصيل (الاقتباس) Acquisition: التي تتم باستخدام أساليب تقنية (حساسات، مبدلات) تختلف باختلاف نوع الإشارة.

*- التحليل Analysis: وهي التضخيم Amplification ، الترشيح Filtering، التبديل الرقمي Digitization، المعالجة Processing .

*- التخزين Storage : وهي عمليات تقليدية تتم بأساليب تقنية أيضاً تبدأ بالدارات الالكترونية البسيطة وتنتهي بأكثر التقنيات تعقيداً (الحواسيب).

١-١ أنواع الإشارات الحيوية

يعتمد في تقسيم الإشارات الحيوية على أسبابها الفيزيولوجية، وباختلاف أنواع الإشارات الحيوية وطبيعتها الفيزيائية وخواصها فقد تكون تلك الإشارات بشكل أساسي: كهربائية، مغناطيسية، كيمائية، ميكانيكية، صوتية أو حتى ضوئية.

وفيما يلي نستعرض باختصار مفهوم كل من هذه الإشارات الأساسية للوصول إلى مجالات قياساتها وتردداتها.

أولا: الإشارات الكهربائية الحيوية Bioelectric Signals

تحدث في الخلايا العضلية والعصبية حوادث كهر وكيميائية عند وصول منبه ما إلى عتبة التنبيه، وتسبب هذه الحوادث انتقالات للشوارد من الوسط خارج الخلوي إلى داخل الخلوي وبالعكس، وتصاحب هذه الانتقالات انتشارات موجهة لإثارة كهربية، تولد حولها حقلا كهربائياً ينتشر في النسج الحيوية المحيطة.

يمكن تحصيل الإثارة الكهربائية الحيوية باستخدام المساري الكهربائية Electrode توضيعها على سطح الجلد. ومن الأمثلة على هذا النوع من الإشارات ما يتم تحصيله في أجهزة: ECG (ElectroCardioGram) تخطيط القلب الكهربائي،

EMG (ElectroMyoGram) تخطيط العضلات الكهربائي،

EEG (ElectroEncephaloGram) تخطيط الدماغ الكهربائي.

ثانياً: الإشارات المغناطيسية الحيوية Biomagnetic Signals

عند دراسة أعضاء حيوية مختلفة (الدماغ، القلب، الرئتين ... الخ) يلاحظ أن نشاطها مصحوب بحقول مغناطيسية ذات شدات ضعيفة جداً، ولذلك يُعمد فقط إلى دراسة الإشارات المغناطيسية الموافقة لإشارات كهربائية تقف وراءها حوادث فيزيولوجية مهمة. ويتم تحصيل هذه الإشارات بوساطة أنواع من الحساسات المغناطيسية فائقة الدقة، تسمى SQUID (Superconducting Quantum Interference Device) أي تجهيزات التداخل الكوانتية فائقة التوصيل وهي مستخدمة في كثير من الأجهزة، مثل:

MCG (MagnetoCardiogGaph) تخطيط القلب المغناطيسي، و

MEG (MagnetoEncephaloGraphy) تخطيط الدماغ المغناطيسي

ثالثاً: الإشارات الكيميائية الحيوية Biochemical Signals

وتنطوي هذه الإشارات على تبدلات في تراكيز العوامل الكيميائية في الجسم (كالغازات، الشوارد، الأملاح)، حيث يمكن تسجيل تبدلات تراكيز الشوارد كالكالسيوم والبوتاسيوم، أو التغيرات في ضغوط الأكسجين وثاني أكسيد الكربون في الأطراف (الضغوط الجزئية Partial Pressures)، مما يعطي معلومات عن تغير سوية تراكيزها في الدم.

رابعاً: الإشارة الميكانيكية الحيوية Biomechanical Signals

تقوم الأجهزة الحيوية بالعديد من الوظائف الميكانيكية كالحركة والجهود والضغوط والجريانات. ففياس الضغط الدموي مثلاً هو قياس لتغيرات شدة وجهة تلك القوى المؤثرة في جدران الأوعية الدموية، ويمكن تسجيل هذه التغيرات بشكل موجي كتابع للزمن، فمن الشكل العام لتخطيط تغيرات الضغط الدموي يمكننا تحديد الضغط الانقباضي وذلك من خلال التموجات الصغيرة العليا للإشارة، أما التموجات الدنيا من الإشارة فهي تحمل معنى للضغط الانبساطي الذي تدخل فيه البطينات القلبية إلى وضع الراحة.

خامساً: الإشارات الصوتية الحيوية Bioacoustical Signals

ويمكن اعتبار هذه الإشارات كمجموعة فرعية من الإشارات الميكانيكية، إذ تتضمن تلك الإشارات الميكانيكية الناجمة عن الاهتزازات المتكررة، حيث تصدر العديد من الأعضاء الحيوية المتحركة إشارات صوتية أثناء عملها، فمثلاً يصاحب عمل القلب نبضات صوتية يمكن الاستماع إليها باستخدام سماعة Stethoscope ، ويمكن تحديد الحالة الصحية للقلب بالاستفادة من هذه الأصوات، كما تولد المفاصل والعضلات وأوعية الجهاز الدوري إشارات صوتية تنتقل في الأوساط المحيطة بها، مما يمكّن من قياسها باستخدام مبدلات صوتية متوضعة على الجلد كالميكروفونات .

سادساً: الإشارات البصرية الحيوية Biooptical Signals

وهي الإشارات التي يتم تحصيلها بناءً على ردود فعلية ضوئية (عاكسة، ماصة، مرسلة) في الأنظمة الحيوية، وتحدث هذه الإشارات إما بشكل طبيعي أو عند تعرض الجسم بشكل ما إلى حزمة إشعاعية خارجية بهدف رصد متحول ذي مدلول حيوي.

١-٢ تحصيل (اقتباس) الإشارات الحيوية

تنشأ الإشارات الحيوية في الخلايا والأعصاب وعضلات الجسم، وتقوم الجملة العصبية بعد ذلك بعملية نقل ومعالجة المعلومات من وإلى العضو المقتبسة منه الإشارة. تؤدي الإشارات الحيوية المقتبسة بشكل صحيح دور كبير في الحكم طبياً فيما إذا كان عضو ما في الجسم البشري سليماً أم مريضاً. وقد أدى التطور التقاني والهندسي عبر الزمن دوراً مهماً في

عملية تحصيل الإشارات الحيوية التي تمتاز بأمور عدة منها:

1- توفير معلومات عن أطوار الإثارة التي تحدث في العضلات والأعصاب.

2- إمكانية إظهار تغيرات مباشرة فيزيولوجية في هذه الإشارات.

3- إمكانية تمييز وظيفة عضو ما وذلك باختيار مكان التحصيل ووسيلته المناسبة.

4- عدم تشويه الحقل الكهرحيوي بفقدان أو نقص القدرة أثناء نقله في سلسلة التحصيل

5- عدم تشويه مواصفات الإشارة الحيوية وتغييرها في أثناء سيرها في سلسلة التحصيل

6- التوصل إلى فصل بين الإشارة الحيوية المفيدة وإشارات الضجيج المتراكبة عليها.

7- إمكانية استقطاب إشارات حيوية تحمل معلومات تشخيصية مهمة على الرغم من بعدها عن مكان نشوئها بشكل غير جراحي في معظم الأحيان، مباشرة عن سطح الجسم، ومما يؤدي إلى تخفيف مخاطر كبيرة (العدوى والتقاط الأمراض). ومن أهم التحصيلات التي نصادفها للإشارات الحيوية، نشير هنا إلى:

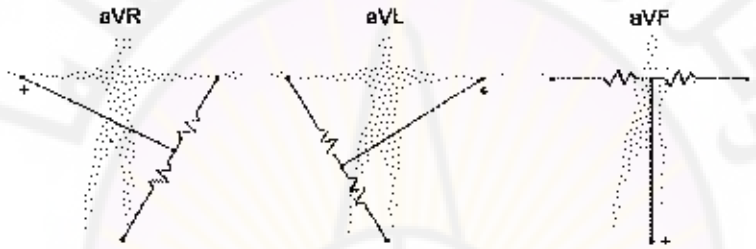
a) تحصيل إشارة القلب ECG

يعدّ القلب من أهم أجزاء الدورة الدموية، ويقوم فيها بوظيفة مضخة التشغيل المبقية على الحياة. ولكي يتم ضمان عمل هذه المضخة على نحو فعّال يجب أن يكون هناك تناسق وتجانس زماني ومكاني لعمل النسج العضلية الفاعلة لكلٍ من الأذينين والبطينين. يقوم بهذه المهمة نظام نقل الإثارة بالقلب الذي يقوم بإثارة أو توليد تحريض منطقه عضلية في القلب تسبب عملية التقلص الميكانيكية لهذه العضلة. يأخذ مخطط كهربائية القلب ECG أشكالا عدة، وذلك حسب نوع التقاط الحقل الكهربائي للقلب. ويمكن تصنيف أنظمة تحصيل مخطط القلب الكهربي ECG إلى أنظمة جراحية وأنظمة غير جراحية. ففي الأنظمة غير الجراحية يتم وضع إلكترودات الالتقاط على السطح الخارجي للجسم ومن ثم إدخال المعلومات المقتبسة إلى منظومة حاسوبية بقصد التشخيص والإظهار. ومن أهم الطرق المعروفة بهذه الأنظمة لتحصيل الإشارة الحيوية وتحويلها إلى إشارة كهربائية نعرض:

*- طريقة التحصيل حسب غولدبرغ Goldberg

وهي طريقة أحادية القطبية Unipolar تعتمد على ترتيب نقاط التحصيل بحيث يتم الكشف عن تغير الجهد في نقطة واحدة وذلك بالنسبة لنقطة مرجعية لا تتأثر خواصها بالنشاط الكهربائي في أثناء طور انقباض القلب، وتقسم هذه الطريقة إلى ثلاث حالات هي:

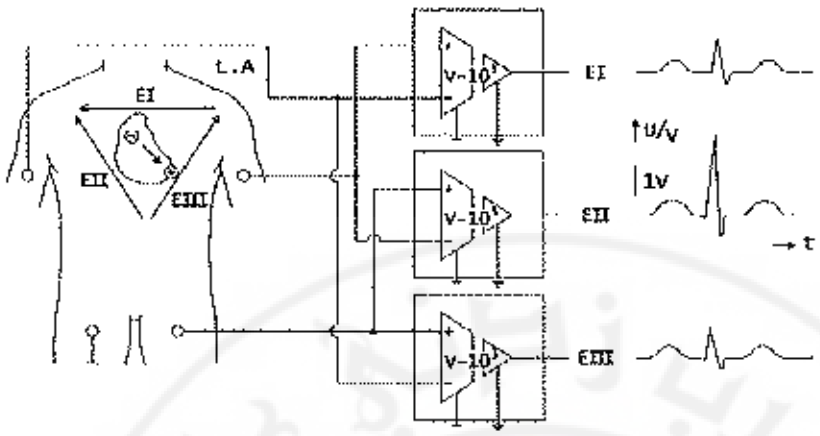
التحصيل aVR على الذراع اليمنى، التحصيل aVL على الذراع اليسرى، التحصيل aVF على الرجل اليسرى. يظهر الشكل1-1 هذه الطريقة بحالاتها الثلاث.



الشكل1-1: إشارة تخطيط القلب حسب طريقة غولدبرغ ECG

*- طريقة التحصيل حسب أينتهوفن Einthoven

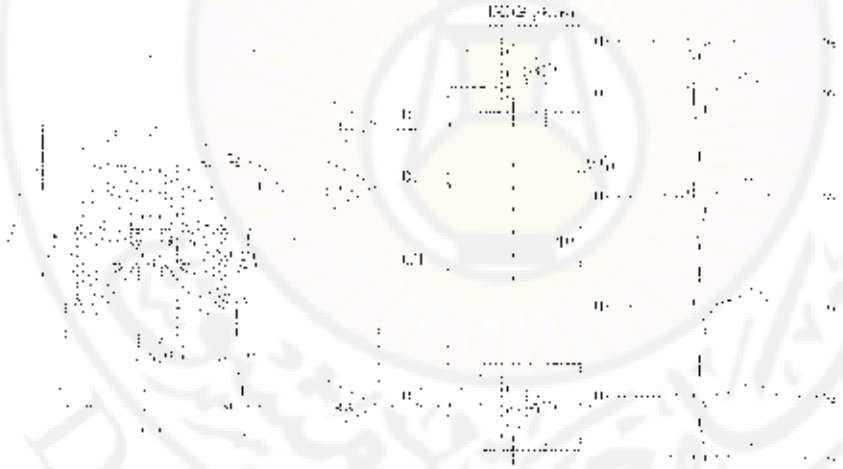
وهي طريقة ثنائية القطبية Bipolar تعتمد على أنه يمكن عَد الفعاليات الكهربائية للقلب بشكل ثنائيات أقطاب، وفيها يتم ترتيب نقاط التحصيل بحيث يتم الكشف عن تغير الجهد بين نقطتين تقع كل منهما في المستوى الأمامي للجسم. وتقسم هذه الطريقة إلى ثلاث حالات من التحصيلات هي: التحصيلI (LEAD I)، التحصيل II التحصيل III. ويظهر الشكل2-1 طريقة التحصيل حسب أينتهوفن في حالاتها الثلاث.



الشكل2-1: إشارة تخطيط القلب ECG حسب طريقة أينتهوفن

*** - طريقة التحصيل حسب ويلسون Wilson**

يبين الشكل 3-1 طريقة ويلسون في تحصيل إشارة القلب الكهربية (طريقة التحصيل الصدرية من V1 وحتى V6)



الشكل3-1: طريقة ويلسون في تحصيل إشارة الـ ECG

تعتمد هذه الطريقة على التحصيل بالكشف عن تغير الجهد في 6 نقاط صدرية مختلفة وبترتيب معين وتسمى نقاط التحصيل من V1 إلى V6 حسب ترتيبها على الصدر. يتم القياس على الالكترود المحدد بالنسبة لنقطة مرجعية لجميع النقاط V1 وحتى V6 والمشكلة

من وصل الكترود الذراع الأيمن مع الكترود الساق اليسرى بحيث لا تتأثر خواص النقطة المرجعية بالنشاط الكهربائي للقلب بشكل ملحوظ، وبذلك يمكن اعتبار الإشارة المقاسة هي إشارة التغير الكهربائي للقلب على الالكترود المعني بعملية القياس.

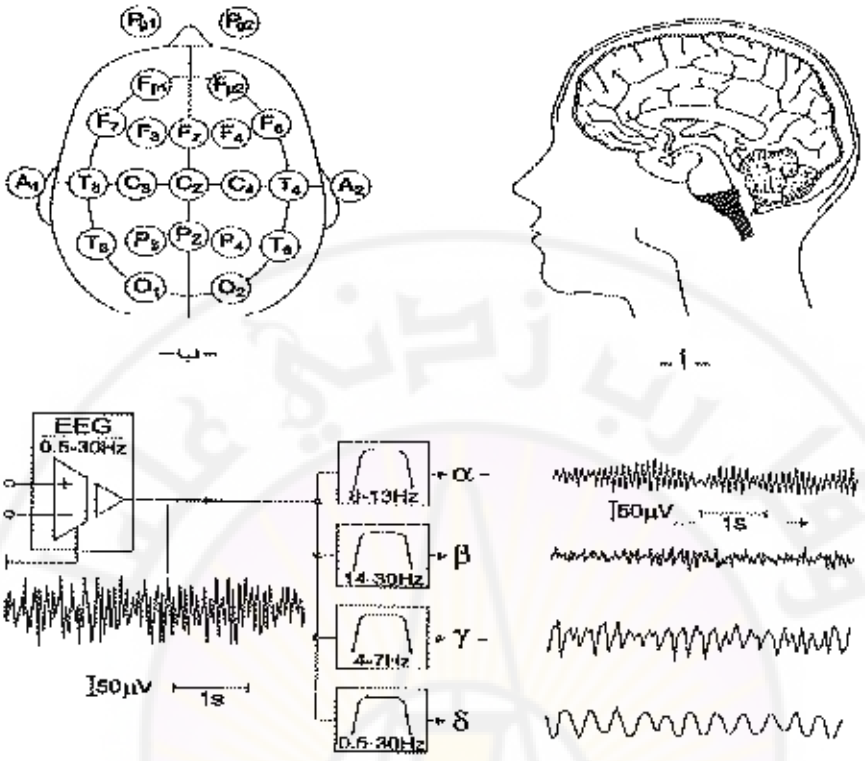
b) تخطيط كهربائية العضلات EMG

تؤدي العضلات وظائف حركية متعددة من خلال تقلصها وانبساطها، وتسبب آلية تقلص العضلات فعاليات كهربائية يمكن تسجيلها.

لا يمكن إعطاء أنظمة تحصيل إشارات العضلات الكهربائية صفة القياسية بسبب تنوع العضلات وأخذها أشكالا وأطوالا مختلفة، فحسب العضلة المراد تحصيل إشارتها يؤخذ في الحسبان نوع الالتقاط وكذلك قيم الإلكترودات المراد استخدامها ومكان توضعها. ويتم عموماً استخدام الإلكترودات السطحية لتحصيل الفعالية الكهربائية الكلية للعضلة، حيث تقوم هذه الإلكترودات بنقل مجموع الطاقة الممثلة للحقل الكهربائي الناتج من العضلة.

c) تخطيط الدماغ EEG

إن مصدر إشارة الدماغ هو المخ الذي تتم فيه جميع عمليات المعالجة وجميع ردّات الفعل اللاإرادية إضافة إلى جميع أطوار الإثارة لجميع أجزاء الدماغ والألياف. هناك أعداد مختلفة من أنظمة التحصيل التي تختلف فيما بينها بعدد الإلكترودات وموضعها على جلد الرأس. ويظهر الشكل 4-1 نظام التحصيل العالمي المعروف بـ 10-20 والمؤلف من 23 نقطة تحصيل. يتم ترتيب الإلكترودات في جميع الأنظمة بحيث تغطي المراكز الرئيسية للمخ. عملياً يتم أخذ من 8 إلى 16 تحصيلا باستمرار وعرضها وتخزينها.



الشكل4-1: أ- المخ منبع لإشارة الدماغ

ب- نظام الالتقاط العالمي 10-20

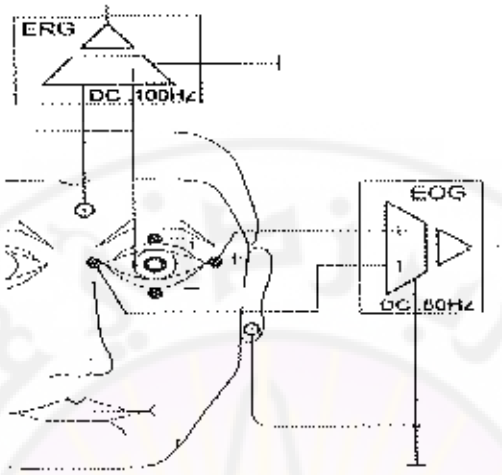
ج- إشارة EEG وتقسيمها إلى أربعة أنواع من الأمواج

d) تخطيط العين EOG" ERG

(Electro – Oculogram" Electro– Retinogram)

يوجد في محيط العين منابع متعددة ومختلفة للفعاليات الكهربائية التي تؤدي إلى ما يعرف بإشارة العين الكهربائية EOG، وإشارة الشبكية الكهربائية ERG. لتحصيل إشارة العين الكهربائية EOG يتم استخدام ترتيبه ثنائية القطبية حيث يتم فيها وضع إلكترودات مداخل المضخم على زوايا الجفون كما هو مبين في الشكل 5-1، في حين يتم وضع الالكترود المرجعي على حلمة الأذن (القرط). أما تحصيل ERG فيتم بوضع الكترود من نوع خاص على القرنية (على شكل نصف كرة على القرنية) أما الالكترود الثاني فيتم تثبيته

على الجبهة والالكترود المرجعي يتم تثبيته على حلمة الأذن. ولتحريض إشارة الشبكة الكهربائية ERG يتم استخدام إشارات ضوئية قصيرة وذات شدة قوية في عملية الإثارة.



الشكل5-1: نظام تحصيل إشارة الـ EOG وإشارة ERG

١-٣ قياس الإشارات الحيوية

تمثل القياسات الحيوية Bio – Measurements في الأساس عملية تحصيل والتقاط لجميع الإشارات الحيوية المتولدة في الجسم ومن ثم نقلها ومعالجتها، وإظهارها. وتعّد الإشارات الحيوية ذات مواصفات كهربائية من حيث التردد والشكل والاستطاعة التي يمكن الحصول عليها إما مباشرة من على السطح الخارجي للجسم أو من أحد أعضائه أو الأجزاء الداخلية فيه.

ويمكن أيضاً الحصول على الإشارات الكهربائية لجسم الإنسان بطريقة غير مباشرة عن طريق المبدلات Transducers أو الحساسات Sensors التي تُحوّل الفعاليات الميكانيكية أو الكيماوية أو الحرارية إلى قيم كهربائية يمكن معالجتها وإظهارها. تتطلب دراسة الإشارات الحيوية في بعض الحالات إثارة العضو بوساطة وحدات إثارة Stimulators ومن ثم قياس رد الفعل العضوي على هذه الإثارة وتحليله، كما هو الحال في بعض قياسات الفعالية الكهربائية للعضلات. تشمل القياسات الحيوية جميع الإشارات الحيوية المتولدة في الجسم البشري، ويمكن تصنيفها حسب المقدار المقاس في عدة أنواع:

1- قياسات كهربية

تعد هذه القياسات من أهم القياسات الحيوية التي تُجرى للإنسان، وذلك لأهميتها التشخيصية. وتعد من الإشارات ذات التوليد المباشر التي تنشأ في الخلايا والأعصاب وعضلات الجسم، ويدخل ضمن هذا النوع من القياسات كل من التحصيلات الأربعة التي سبق وتمت الإشارة لها، أي ECG, EMG, EEG, EOG" ERG.

2- قياسات ميكانيكية

تعتمد هذه القياسات على المعايير الأساسية للميكانيك مثل الكتلة والمسافة والزمن التي يمكن إرجاع جميع القيم الأخرى إليها، فمثلا يمكن قياس المسافة عن طريق قياس تغير السعة أو تحريض حقل مغنطيسي أو تغير مقاومة أو قياس فرق زمن المسار بين نبضة مرسلة وانعكاس هذه النبضة عن الجسم. أما السرعة فما هي إلا مشتق المسافة التي يمكن قياسها عن طريق قياس مقدار تغير التردد بين الإشارة المرسلة والمستقبلة كما في أثر دوبلر (قياس سرعة جريان الدم). كما تضم هذه القياسات كلا من قياسات القوة والضغط التي تعتمد مبدأ تغير الشكل الهندسي لعنصر القياس.

3- قياسات حرارية

هناك نوعان من قياسات حرارة جسم الإنسان.

الأول: قياس حرارة سطح الجسم أو الجلد.

والثاني: قياس الحرارة الجوفية أو ما يعرف بالحرارة الداخلية لأعضاء الجسم. وتعتمد القياسات في كلتا الحالتين على تغير الخصائص الفيزيائية للمواد وأحجامها بالنسبة لدرجة الحرارة، ويمكن أن يكون هذا التغير على شكل تمدد لسائل متعلق بدرجة الحرارة مثل الزئبق، أو على شكل تغير في المقاومة الكهربائية بالنسبة لدرجة الحرارة.

٤- قياسات كيميائية

تعتمد هذه القياسات مثل القياسات التي تُجرى على الدم لمعرفة مواصفاته، أو لقياس درجة الحموضة pH، أو قياس نسبة الأكسجين على مبدلات تنتج تياراً كهربائياً نتيجة تفاعل كيميائي في دارة قياس خارجية تحتوي على معدنين مختلفين في محلول كهرليتي.

ويبين الجدول 1 – 1 أهم الإشارات الحيوية المقاسة.

الجدول 1 – 1: أهم الإشارات الحيوية المقاسة مع مجالات قياساتها وترددانه.

نوع القياس	المجال الحيوي	التردد [Hz]
جريان الدم	1 – 300 mL/s	0 – 20
ضغط الدم (مم زئبقي)	0 – 400 mmHg	0 – 50
ناتج القلب (ل / دقيقة)	4 – 20 L/min	0 – 20
إشارة القلب الكهربائية ECG	0.5 – 4 μV	0.05 – 50
إشارة الدماغ الكهربائية EEG	5 – 300 μV	0.5 – 150
إشارة العضلات الكهربائية EMG	0.1 – 5 mV	0 – 10000
إشارة العين الكهربائية EOG	0.01 – 3.5 mV	0 – 100
إشارة الشبكية الكهربائية ERG	0 – 900 μV	0 – 50
درجة الحموضة PH	3 – 13	0 – 1
الضغط الجزئي لثاني أوكسيد الكربون	40–100 mmHg	0 – 2
الضغط الجزئي لغاز الأوكسجين	30–100 mmHg	0 – 2
جريان التنفس (ل/دقيقة)	0 – 600	0 – 40
تردد التنفس (مرة/دقيقة)	2 – 50	0.1 – 10
الحرارة (م)	32 – 40 Co	0 – 0.1

١-٤ تجهيزات قياس الإشارات الحيوية

لقياس الإشارات الحيوية، يجب أن يمتاز الجهاز أو المنظومة المستخدمة لذلك بميزات خاصة تمكنه من تحقيق الغرض المطلوب بشكل صحيح ودقيق، لذلك فإن مثل تلك الأجهزة تتألف وبشكل أساسي من أربعة مكونات رئيسية هي:

*- الكمية الفيزيائية المقاسة

*- المجس أو المتحسس Transducer / Sensor الذي سيقيس تلك الإشارة من مصدرها بشكل مباشر عن طريق تحويل الطاقة الأصلية التي تمثلها الإشارة (حسب أنواعها المختلفة) إلى طاقة كهربائية يسهل التعامل معها.

*- معالج الإشارة Signal Conditioner ، الذي يقوم بتغيير بعض خواص الإشارة المقاسة لملاءمتها بشكل أكثر وأدق (دون أن يؤثر ذلك على الغرض المطلوب من قياسها) من خلال تكبير مدى تلك الإشارة Amplitude (حيث تمتاز معظم الإشارات الحيوية بقلة مداها)، أو ترشيح حزمة التردد الخاصة بها Bandwidth ومنع التشويش الحاصل من التداخل مع مؤثرات المحيط الخارجي، أو تحويل نمط تلك الإشارة من النمط المتشابه Analogue إلى النمط الرقمي Digital ليسهل التعامل معها وتحليلها بوساطة الحاسوب لزيادة دقة القياس. يتم ذلك كله باستخدام دوائر الكترونية مثل المكبرات والمرشحات وغيرها.

*- وحدة العرض Display System التي تمثل جزء الإخراج للمنظومة ومن خلالها سيتم عرض التمثيل النهائي للإشارة المقاسة (المطلوبة)، كما تتعدد وسائل العرض تلك فقد تكون على شكل شاشة عرض CRT أو بشكل إشارة مرسومة ومطبوعة على ورق حراري خاص أو ممثلة على شريط مغناطيسي.

يجب أن تكون منظومة القياس مزودة بالإضافة للمكونات الرئيسة المذكورة بوحدة إنذار Alarm من أجل التحذير بوجود خلل في قياس الإشارة، كما أن وجود وحدة لخزن المعلومات له أهمية كبيرة خصوصاً في مجال البحوث العلمية. تمثل أجهزة ومنظومات قياس الإشارة الطبية ركن أساسي من أركان الأجهزة الطبية بالمستشفيات.

تعددت أنواع ومسميات أجهزة قياس الإشارات الطبية باختلاف الإشارة الحيوية التي تقيس، نذكر منها الأجهزة المستخدمة في قياس تخطيط القلب الكهربائي EKG ، تخطيط الدماغ المغناطيسي MEG ، تخطيط العضلات EMG ، ممانعة الجلد GSR وظائف الرئتين Spirometry وتركيز الأوكسجين في الدم Oximetry. وتتلخص وظيفة تجهيزات التقاط الإشارة الحيوية بالنقاط التالية:

1- التقاط الإشارة الحيوية وتحصيلها من دون تشويه أو تغيير في مواصفاتها (من حيث الشكل أو المطال،..).

2- تضخيم الإشارة الحيوية إلى مستوى يمكن عنده معالجة الإشارة جيداً، بحيث يمكن عند هذا المستوى للمطال معالجة الإشارة وتحليلها سواء تمثيلياً أو رقمياً.

3- تخليص الإشارة من كل المركبات غير المرغوب فيها (إشارات التشويش).

4- إمكانية عرض هذه الإشارة أو طباعتها أو تسجيلها.

وتختلف أجهزة القياس عن بعضها باختلاف مبدأ وطريقة القياس المستخدمة للحصول على المعيار المطلوب، ويمكن التمييز بين ثلاثة أنواع من هذه التجهيزات هي:

1- الأجهزة التي تعتمد الطرق الجراحية Invasive: يتم فيها التقاط الإشارة من داخل الجسم، مثل أجهزة قياس إشارات الضغط والجريان بوساطة القثاطر Catheters.

2- الأجهزة التي تعتمد الطرق غير الجراحية Non-Invasive: يتم فيها التقاط الإشارات عن السطح الخارجي للجسم إما مباشرة بوساطة الإلكترودات السطحية، مثل إشارات القلب والعضلات والدماغ، أو عن بعد مثل قياس درجة حرارة الجسم بوساطة الأشعة تحت الحمراء.

3- الأجهزة التي تقوم بتحويل الإشارات غير الكهربائية إلى قيم كهربائية بالاستعانة بالمبدلات والحساسات، إذ يمكن بعدها دراسة الإشارة وتحليلها. ويبين الجدول 2-1، أهم الطرق المستخدمة في أجهزة قياس الإشارات الحيوية التي وردت في الجدول 1-1.

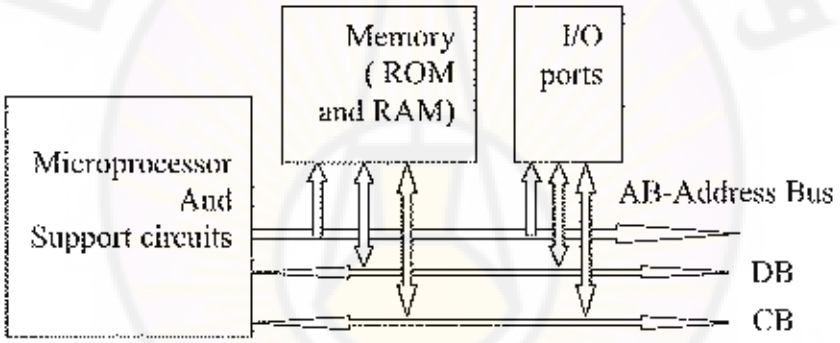
الجدول2-1: أهم الطرق المستخدمة في أجهزة قياس الإشارات الحيوية

نوع القياس	الطريقة المستخدمة
جريان الدم	الكهرومغناطيسية أو الأمواج فوق الصوتية Electromagnetic or Ultrasonic
ضغط الدم	كم الضغط أو مقياس الإجهاد Cuff or Strain gage
نتاج القلب	مبدأ فيك، تمدد الكاشف أو تغير الممانعة الكهربائية الصدر Fick, Dye dilution or Impedance Plethysomgraphy
إشارة القلب الكهربائية ECG	الإلكترودات على سطح الجلد Skin electrodes
إشارة الدماغ الكهربائية EEG	الإلكترودات على فروة الرأس Scalp electrodes
إشارة العضلات الكهربائية EMG	الإلكترودات الإبرية Needle electrodes
إشارة الشبكية الكهربائية ERG	إلكترودات كعدسات لاصقة Contact lens electrodes
درجة الحموضة PH	الكترود الحموضة pH electrode
الضغط الجزئي لغاز ثاني أوكسيد الكربون PCO2	الكترود PCO2 PCO2 electrode
الضغط الجزئي لغاز الأوكسجين PO2	الكترود PO2 PO2 electrode
جريان التنفس	مقياس التنفس Pneumotachometer
تردد التنفس	الممانعة Impedance
الحرارة	الثيرموستات Thermostat

الفصل الثاني: الحاسوب الرقمي

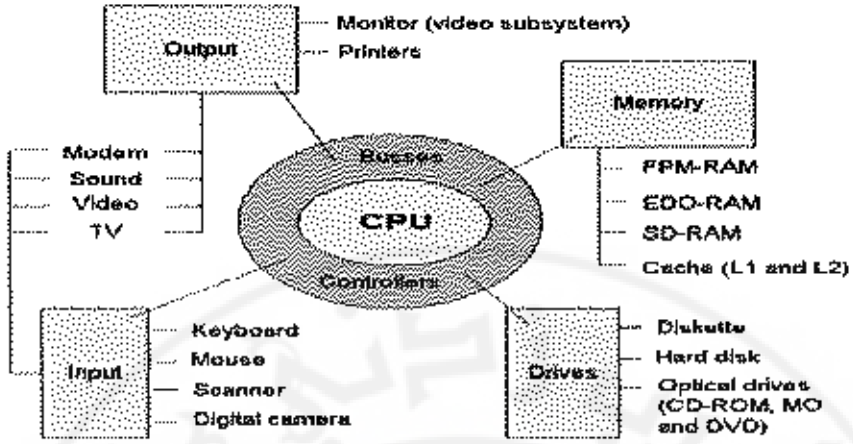
ظهر الحاسوب الرقمي Digital computer حصيلة لعدة سنوات من البحث والتطوير لتقنيات أنصاف النواقل Semiconductor Technology وقد تركز هذا التطوير على زيادة إمكانية التغليف للدوائر المتكاملة وسرعة تنفيذ العمليات، إضافة إلى قلة التكلفة وزيادة التطبيقات العملية.

يعرف الحاسوب الرقمي على أنه الجهاز الذي يقوم بسلسلة من العمليات الرياضية والمنطقية على مجموعة من البيانات Data باستخدام برامج مكتوبة بإحدى لغات البرمجة الخاصة بالحواسيب . يمكننا أن نتصور بأن الشكل العام لأي حاسوب رقمي وتحديدا الحاسوب الصغري أو المايكروي (مايكرو كومبيوتر Microcomputer) كما في الشكل 1-2.



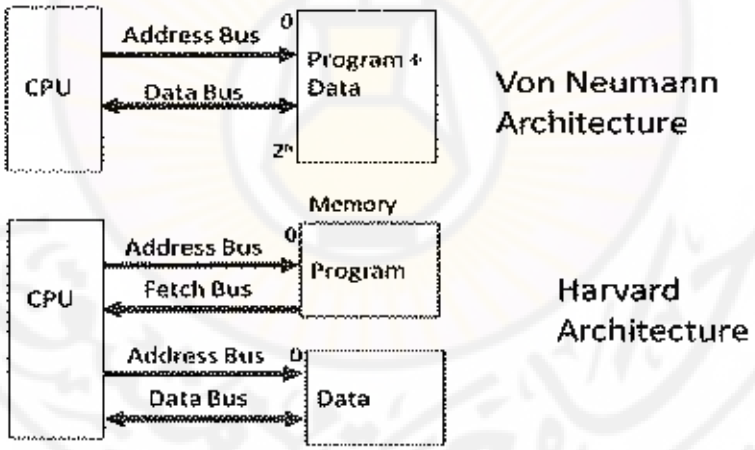
شكل 1-2: المخطط الصندوقي للمايكرو كومبيوتر

يعدّ المعالج الصغري العنصر الأساسي في تركيبة الحاسوب الصغري (الشخصي Personal Computer). تعود بدايات الحواسيب إلى عام 1940 عندما اقترح العالم Von Neumann نموذجه المشهور الذي يتمثل بأن كل حاسوب يتكون بالإضافة إلى وحدة المعالجة المركزية من ذاكرة وبوابات للدخل والخرج ومجموعة من الساعات وفق ما هو مبين بالشكل 2-2 .



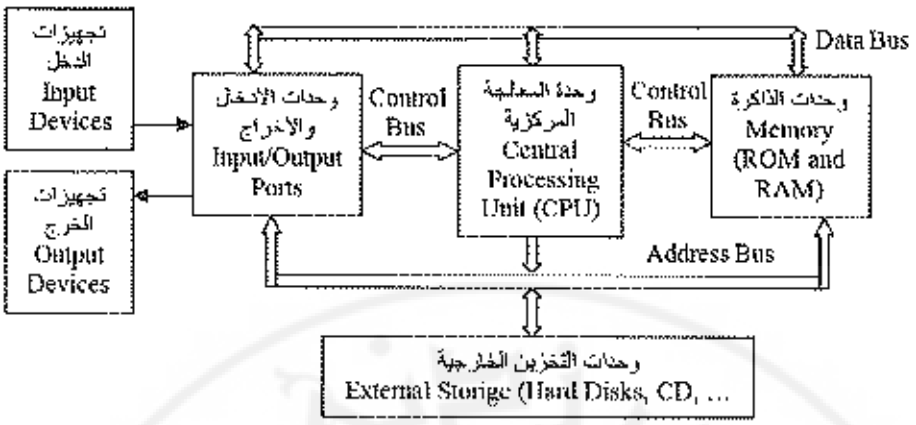
الشكل2-2: تركيبة الحاسوب الرقمي وفق معمارية فون نويمان

ونشير هنا إلى أنه بالإضافة لنموذج فون نويمان (الذي فيه تخصص الذاكرة للبرامج والمعطيات معاً) يوجد نموذج آخر، يسمى نموذج Havard (الذي فيه يخصص قسم من الذاكرة للبرامج فقط وقسم آخر منفصل للمعطيات) وفق ما هو مبين بالشكل3-2:



شكل3-2: البنى الأساسية المتعارف عليها للحاسوب الصغري

وبما أن معظم الحواسيب الشخصية التي نصادفها مبنية وفق نموذج فون نويمان، نعيد ترتيب المكونات الأساسية للحاسوب الصغري بشكل آخر وفق الشكل4-2.



شكل4-2: المخطط الصندوقي للحاسوب الصغري (المايكروي)

ونصادف بالحواسيب المصطلحات التالية:

- المكونات الصلبة أو العتاد للحاسوب Hardware
- المكونات اللينة أو البرمجيات Software
- البرامج البنوية الداخلية Firmware

فالمكونات الصلبة هي الاسم المعطى للأجهزة الفيزيائية ومجموعة الدارات الكهربائية للحاسوب، أو بمعنى آخر كل ما يلمس باليد من الحاسوب. وتشير المكونات اللينة إلى مختلف البرامج، أما البرامج البنوية الداخلية فهي المصطلح المعطى للبرامج المخزنة في ذاكرة القراءة ROM لتهيئة عمل الحاسوب ومختلف الأعمال ذات الطابع التحكمي. أما بخصوص أنواع الحواسيب الرقمية ، فقد كانت الحواسيب في البدايات تصنف وفق معيار سرعة التنفيذ وحجم البيانات المخزنة إلى ثلاث أنواع أساسية هي:

- حواسيب صغرية Microcomputers
- حواسيب متوسطة Minicomputers
- حواسيب كبيرة Main Frame computers

تأتي الحواسيب الكبيرة عادة بإمكانيات عالية جداً وتستخدم على سبيل المثال في المجالات العسكرية وفي معالجة معطيات الأعمال على مستوى بلد ما، أما الحواسيب الصغيرة فتستخدم بتطبيقات محددة على مستوى شخص، أو مكتب. والآن بفعل التطور السريع

لتقنية الحواسيب تداخلت الحدود بين هذه الأنواع الثلاثة للحواسيب، وأصبح من الصعب تحديد الحد الفاصل بين كل من هذه الأنواع ، إذ لم يعد هناك من شيء جوهري يميز بين هذه التصنيفات سوى أن وحدة المعالجة المركزية للحاسوب الصغري تكون عادة عبارة عن دارة متكاملة واحدة تدعى بالمعالج الصغري.

٢-١ طرق الولوج للحاسوب

بالنظر إلى أننا نعيش في مجتمع معلوماتي التوجه لابد أن نشير هنا أيضاً إلى بعض المصطلحات الأخرى المتعلقة بالحواسيب وخاصة ما يتعلق منها بالطرق المستخدمة لإعطاء المستخدمين إمكانية الولوج إلى الحاسوب ومنها:

1- المعالجة الدفعية Batch processing

بالنسبة للحواسيب القديمة، كان يتوجب على المبرمجين تجهيز البرامج على شكل دفعات Batches، بحيث لا يمكن للحاسوب أن يبدأ بتنفيذ برنامج جديد إلا بعد الانتهاء من البرنامج القديم. ولذلك كان من المفروض على المبرمج أن ينتظر بعض الوقت لكي يحصل على نتيجة عمله، أي أن نظام من هذا النوع يخدم مستخدم واحد في وقت واحد ولا يسمح بتفاعل سهل متبادل بين الحاسوب والمستخدم.

2- البرمجة المتعددة Multiprogramming

يسمح هذا النوع من البرمجة بوضع عدة برامج منفصلة في ذاكرة الحاسوب في نفس الوقت. وبالتالي يبدأ الحاسوب بتنفيذ البرنامج ويتابع ذلك حتى يصادف نقطة برمجية معينة تحتاج للتعامل مع جهاز محيطي كالطابعة. فإذا كانت الطابعة مشغولة فسيخزن الحاسوب المعطيات الواجب طباعتها على قرص مغنطيسي وبعدئذ يستطيع أن يبدأ عملا برمجيا آخر ريثما تصبح الطابعة غير مشغولة. وعندما تصبح الطابعة متاحة يتمكن الحاسوب من إرسال نتائج البرنامج الأول ومن ثم يعود إلى البرنامج الثاني. أي أن عملية البرمجة تزيد من فعالية وكفاءة الحاسوب وذلك بإبقائه مشغولا معظم الوقت.

3- الأنظمة المتشاركة بالزمن والمتعددة المهام

تتصل في هذه الأنظمة Time-Share and Multitasking وبنفس الوقت عدة طرفيات، وبالتالي يستطيع الحاسوب أن يخدم عدداً من المستخدمين بطريقة التقسيم الزمني بينهم وفق كميات صغيرة كأن يعطي لبرنامج المستخدم الأول 10 ms، والثاني والثالث نفس الفترة وهكذا حتى يتم تخديم كل المستخدمين. ينهي الحاسوب دورة كاملة بتخديم كافة المستخدمين، لكن يبدو للمستخدم بأنه الوحيد وذلك لان الحاسوب يعالج المعطيات بسرعات كبيرة تفوق بكثير سرعة إدخال المعطيات من قبل المستخدمين.

4- المعالجة الموزعة أو المعالجة المتعددة Distributed Processing

يملك في هذا النظام كل مستخدم (أو مجموعة مستخدمين) حاسوبا صغيرا عوضا عن طرفية إظهار فقط. وهذا يعني أن كل مستخدم يستطيع تنفيذ مهامه كليا باستخدام الحاسوب الصغري دون اللجوء إلى الحاسوب الضخم وبالتالي كل مستخدم يستطيع أيضاً الولوج إلى برامج الحساب أو أية موارد أخرى للحاسوب الضخم عند الحاجة لها.

٢-٢ التركيبة العامة للحواسيب الصغرية

يمكننا بشكل مختصر تعريف كل من المكونات الثلاثة الأساسية للحاسوب كالتالي:

١- وحدة المعالجة المركزية Central Processing Unit - CPU

تسيطر على عمل الحاسوب ككل، فهي التي تجلب التعليمات المشفرة من الذاكرة، وتفككها إلى سلسلة من الأفعال، ثم تنفذها. وتضم CPU المكونات التالية:

- وحدة الحساب والمنطق ALU (Arithmetic Logic Unit) التي تستطيع انجاز العمليات الحسابية والمنطقية على كلمات ثنائية عندما تعطى الأمر بتنفيذ ذلك.
- عدّاد عناوين Address Counter يستخدم للاحتفاظ بعنوان التعليمة التالية.
- مسجلات ذات أغراض عامة تستخدم من أجل التخزين المؤقت للمعطيات ونقلها من مكان لآخر ضمن المعالج.
- جملة من الدارات التي تولد إشارات سمر التحكم وتشمل كافة الإشارات القيادية للمعالج.

٢- وحدة الذاكرة Memory

يتألف هذا الجزء عادة من مزيج من نواكر ROM و RAM التي تمثل الوسيط لقراءة بعض الأوامر ومنطقة العمل والتخزين المؤقت لما بين النتائج، وتضم عادة البيانات البرامج التي هي قيد التنفيذ . يوجد هدفين لاستخدام الذاكرة هما:

الأول: تخزين الشفرات الثنائية لتسلسل من التعليمات التي يراد تنفيذها.

الثاني: تخزين المعطيات المشفرة ثنائيا التي سيعمل عليها الحاسوب.

وبالنظر لأهمية هذه الوحدة في التركيبة العامة لأي حاسوب، نفرد لها فقرة كاملة لاحقا.

٣- وحدات الدخل والخرج Input/output Ports

ويسمح هذا الجزء I/O Ports للحاسوب بأن يتلقى المعطيات من الوسط الخارجي أو يرسل المعطيات إلى الوسط الخارجي، حيث يوصل بوحدات الدخل تجهيزات الدخل Input Peripherals مثل لوحة المفاتيح Keyboard وغيرها ويوصل بوحدات الخرج تجهيزات الخرج مثل الطابعة Printer وغيرها من الملحقات.

تؤدي وحدات الدخل والخرج دوراً أساسياً في تهيئة معطيات الدخل للشكل المناسب للمعالجة ضمن الحاسوب وتهيئة النتائج للشكل المناسب لفهمها من قبل المستخدم. وهنا نشير إلى أنه عند التعامل مع البوابات تستخدم بعض المصطلحات مثل:

المسجل Register- يمثل وسط تخزين ضمن المعالج يستخدم عادة لحفظ البيانات التي يتطلب الولوج لها سرعة عالية.

الحاجز Buffer- يستخدم للاحتفاظ بالمعطيات ضمن المعالج للتعامل معها قبل إرسالها لأي جهاز آخر . والـ Buffer هو نوع من المسجلات يقوم بتقوية أي إشارة توضع على الدخل بدون تغير القيمة.

الماسك Latch- يمثل دارة الكترونية قادرة على تخزين بيت واحد من الذاكرة، مثله مثل القلاب. وغالبا ما تكون منافذ الدخل أو الخرج عبارة عن مجموعة من الـ Buffers والـ Latches التي تسمح للمعطيات بالمرور خلالها عندما تكون ممكّنة Enabled.

توصل الأجزاء الرئيسية للحاسوب عبر ثلاثة مجموعات من النواقل (الخطوط) التي تدعى بالممرات Buses. تقوم هذه الممرات بتأمين نقل البيانات ضمن الكتل الرئيسية للحاسوب وتواصل الحاسوب مع الوسط الخارجي، حيث يؤدي نوع الممر (الممر) من عرض وسرعة دور كبير في تميز حاسوب على آخر.

يرمز بالمخطط الصندوقي للحاسوب لكل من الممرات الثلاثة الرئيسية بخطين متوازين للإشارة إلى أن كل ممر مكون من أكثر من خط، وهذه الممرات الثلاثة هي:

AB – Address Bus ممرات لنقل العناوين: ترسل الوحدة CPU على هذه الخطوط عنوان موقع ذاكرة أو عنوان بوابة، وذلك للكتابة فيها أو القراءة منها. يحدد عدد خطوط هذا الممر عدد المواقع الذاكرية التي تستطيع الوحدة CPU من عنونتها. فإذا كان عدد خطوط ممر العناوين للوحدة CPU هو N فإن الوحدة تستطيع أن تعنون مباشرة 2^N موقعا ذاكريا. فمثلا من أجل عشرة خطوط عنونة من A_0 وحتى A_9 يمكن عنونة 2^{10} موضع ذاكرة، وهذا ما يعادل 1024 موضع ، أي 1 KB.

DB – Data Bus ممرات لنقل المعطيات: وهي ثنائية الاتجاه Bidirectional وهذا يعني أن الوحدة CPU تستطيع قراءة المعطيات على هذه الخطوط من الذاكرة أو من منفذ ما، وتستطيع أن ترسل المعطيات على هذه الخطوط إلى موقع ذاكرة أو إلى منفذ معين. يمكن أن تكون مخارج عدة أجهزة في النظام موصولة إلى ممر المعطيات، بحيث إن المخارج التابعة لجهاز واحد فقط ستكون ممكنة (مسموح لها) وبالتالي يجب أن تكون مخارج أي جهاز موصول على ممر المعطيات ثلاثية الحالة Three-state، أي أن مخارج الجهاز تكون معومة Float عندما لا يكون الجهاز قيد الاستخدام.

CB – Control Bus ممرات لنقل إشارات القيادة والتحكم: حيث ترسل الوحدة CPU إشارات التحكم لتمكين مخارج الأجهزة الذاكرية المعنونة أو أجهزة المنافذ، ويمكن أن تستقبل بعض الإشارات من المحيط الخارجي كطلبات المقاطعة. وبالتالي إن بعض خطوط هذا الممر قد تكون دخل والبعض الآخر قد يكون خرج. وهكذا لقراءة بايت معطيات من موقع ذاكري مثلا، ترسل الوحدة CPU عنوان البايت المطلوب على ممر العناوين، وبعدئذ

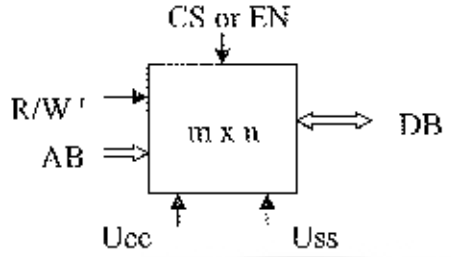
ترسل إشارة MR (Memory Read) القراءة من الذاكرة على ممر التحكم، ثم تمكن إشارة القراءة الذاكرة المعنونة لتخرج بايت المعطيات على ممر المعطيات وذلك لقراءة هذا البايت بوساطة الوحدة CPU. يوجد في الحواسيب الحديثة بالإضافة للممرات الثلاثة المذكورة أعلاه ممرات أخرى، كممر التغذية مثلا الذي يضم كافة جهود التغذية اللازمة للحاسوب ومكوناته الأساسية

٢-٣ مفهوم الذاكرة

لتخزين كمية كبيرة من المعلومات المثنائية، نلجأ عادة لاستخدام وحدة الذاكرة التي تضم دارة واحدة أو أكثر، وتمثل الذاكرة بشكل عام مجموعة كبيرة من الخلايا القادرة على تخزين حجم أكبر بكثير من المعلومات التي يمكن تخزينها بالمسجلات.

تشير هنا إلى الذواكر المصنعة من أنصاف النواقل فقط، بعدما حل هذا النوع من الذواكر محل الحلقات المغناطيسية (التي كانت توجد في الأجيال الأولى للحاسبات). ينظر للذاكرة Memory على إنها دارة منطقية تستعمل لتخزين المعلومات وتشكل مع المعالج العناصر الأساسية في تركيبة أي حاسوب، حيث لا يمكن لمعالج صغري مستخدم بتطبيق ما أن يكمل أعماله بمعزل عن وجود ذاكرة لتخزين المعلومات المستقبلة من تجهيزات الدخل لمعالجتها ونقل النتائج إلى تجهيزات الخرج لإظهارها.

يمكن تصور الذاكرة كمجموعة من السجلات المتماثلة في الطول. تسمى مجموعة الخلايا التي تشكل صفاً واحداً في مصفوفة الذاكرة بالموقع Location أو بطول كلمة الذاكرة Memory Word. تمثل الخلية Cell الواحدة في الذاكرة أصغر حيز لتخزين البيانات، حيث كل خلية قادرة على تخزين معلومة بطول 1 bit (0 أو 1). يرمز للذاكرة بـ mxn، بحيث يشير الرمز m لعدد صفوف المصفوفة (المواضع) و n لعدد الأعمدة (عدد الخلايا في الصف الواحد) و الجداء mxn إلى الحجم الكلي للذاكرة. يوجد لكل ذاكرة مصنعة على شكل دارة متكاملة مجموعة من الإشارات القياسية مثل الإشارات المبينة بالرمز الصندوقي للذاكرة في الشكل 5-2.



شكل 5-2: الرمز الصندوقي للذاكرة مع أهم الإشارات القياسية

يتم التعامل مع الذاكرة من خلال عمليتين أساسيتين هما:

الأولى – كتابة المعلومات ضمن الذاكرة Write Operation (ونرمز لها W).

الثانية – قراءة المعلومات من الذاكرة Read Operation (ونرمز لها R).

توجد إشارة تحكم منفصلة لكل من هاتين العمليتين، حيث كل إشارة منهما قد تكون فعّالة بالصفر أو بالواحد، ويمكن دمج الإشارتين بخط واحد بالاسم R/W'، بحيث تكون إحداها فعّالة بعكس الأخرى (لأنه لا يمكن بنفس اللحظة إجراء عمليتي القراءة والكتابة) تكون الذاكرة موجودة مع غيرها من الدارات المتكاملة ضمن منظومة ما. لتميز قطعة الذاكرة المراد التعامل معها بلحظة ما عن غيرها من الدارات الأخرى، تستخدم إشارة تحكم بالاسم CS (اختصاراً من Chip Select وفي بعض الدارات يستخدم الرمز EN من Enable أي التمكين) لانتقاء الدارة. قد يكون للقطعة الواحدة أكثر من إشارة CS (بعضها فعّال بالصفر والبعض الآخر فعّال بالواحد).

يشير الرمز AB في الشكل أعلاه إلى ممرات العناوين و DB إلى ممرات المعطيات. يكون مصدر ممرات العناوين AB دوما المعالج، أما ممرات المعطيات فقد تكون بالاتجاهين أي للكتابة وتكون داخلة أو للقراءة وتكون خارجة من الذاكرة. تحمل الممرات AB قيماً ثنائية تمثل العناوين و DB قيماً ثنائية تمثل المعطيات.

يعطي عدد خطوط AB المجال العنواني الممكن للذاكرة. بمعنى انه بخط عنونة واحد A_0 يمكن عنونة $2^1 = 2$ موضع في الذاكرة، هما 0 لموضع و 1 لموضع آخر. ويمكن بخطين A_1A_0 عنونة 4 مواضع تكون بالعناوين 00, 01, 10, 11 ويمكن بـ 16 خط عنونة 2^{16} موضع، أي من العنوان:

0000H = 0000 0000 0000 0000B

0001H = 0000 0000 0000 0001B

.....

وحتى FFFFH = 1111 1111 1111 1111B وهذا ما يعادل 64 KB . يعطي عدد خطوط DB طول كلمة الذاكرة، وقد يكون هذا الطول 1 bit، وبهذه الحالة تكون مصفوفة الذاكرة مكونة من عمود واحد، وقد يكون طول الكلمة 4 bit. الطول القياسي لكلمة الذاكرة هو 1 byte = 8 bits انطلاقاً من أن المعطيات تكون ممثلة في الذاكرة عادة وفق كود ASCII الذي يخصص بايت واحد لكل محرف. آلية التعامل مع الذاكرة بغض النظر عن نوعها تتم كالتالي:

يرسل المعالج عندما يقرر التعامل مع الذاكرة، عنوان الموضع الذي يرغب التعامل معه في الذاكرة عبر AB وبعد تفعيل الإشارة CS وانتقاء الذاكرة عن غيرها، يرسل المعالج إشارة التحكم بحسب العملية المطلوبة على المأخذ R/W' وفور جهوزية الذاكرة للاستجابة، يتم إرسال المعطيات أو استقبالها بحسب العملية المقررة عبر الممر DB.

٢-٣-١ أنواع الذواكر

تصنف الذواكر المصنعة من أصناف التواقل وفق معايير عدة، نذكر منها المعايير الأكثر انتشاراً مثل معيار: – العبور (الوصول) إلى المعطيات داخل الذاكرة

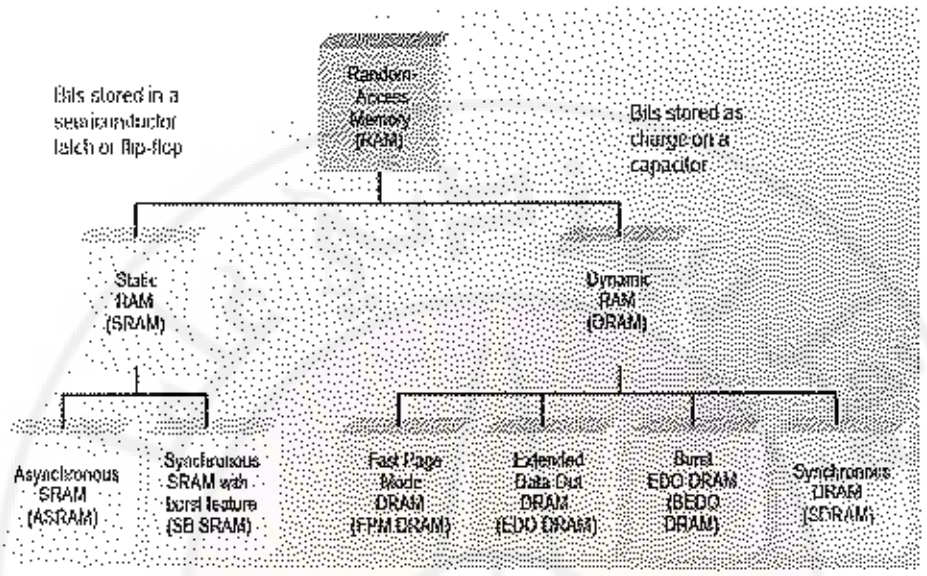
– مبدأ تحقيق خلايا الذاكرة الأولية

– إمكانية تغيير محتويات الذاكرة

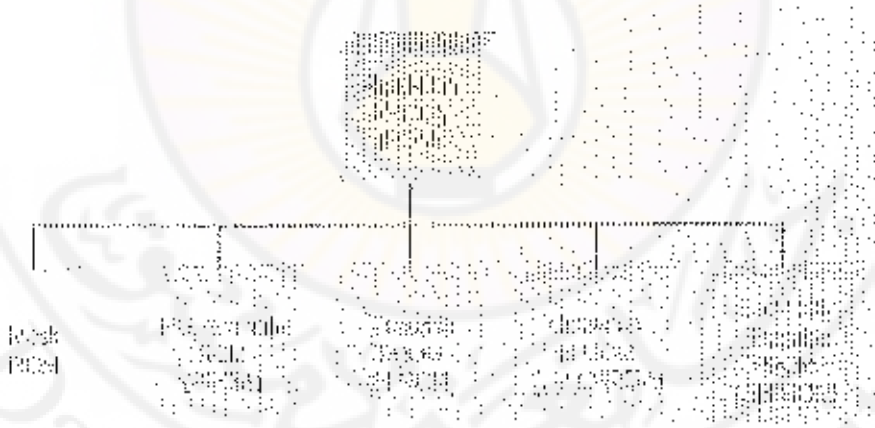
من طرف آخر إن مجمل أنواع الذواكر تصنف بعائلتين وفق ماهو مبين بالشكل 6-2. ويمكننا توصيف هاتين العائلتين وفق التالي:

*– ذواكر القراءة والكتابة RWM من Read Write Memories وتسمى اختصارا RAM التي يمكن استخدامها لعملية قراءة المعطيات وكتابتها. وتميز بحسب معيار المبدأ المستخدم في تنجيز خلايا الذاكرة الأولية لذواكر RAM بين نوعين أساسيين هما: ذواكر بخلايا ساكنة Static وفيها تكون كل خلية مكونة من ثنائي استقرار.

وذواكر بخلايا متحركة Dynamic وفيها تكون كل خلية في الذاكرة مبنية على استخدام السعات البارازيتية.



a) عائلة ذواكر RAM

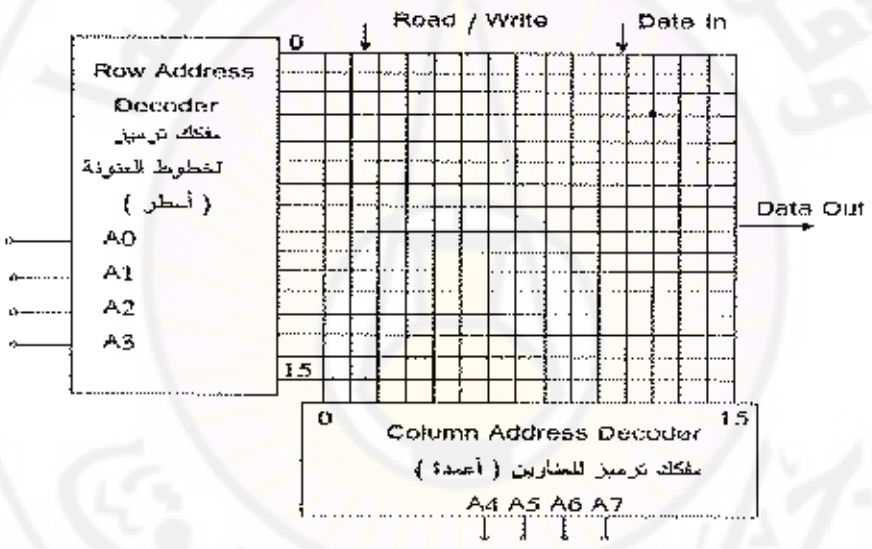


b) عائلة ذواكر ROM

الشكل6-2: الأنواع الأساسية للذواكر

ترتب خلايا الذاكرة RAM على هيئة مصفوفة بحيث يمكن اختيار البيانات المخزنة بموقع ذاكرة عن طريق تطابق خط اختيار صف وخط اختيار عمود.

ولتوضيح ذلك أكثر ننطلق من مثال محدد وليكن تركيبة ذاكرة RAM بسعة ربع كيلو. نحصل على السعة المذكورة بترتيب مصفوفة الذاكرة على شكل جداء مابين كل من عدد الصفوف وعدد الأعمدة وفق الشكل: 256 x1 bits أو بالشكل 2 x 128 أو 4 x 64 أو 8 x 32 وأخيرا 16 x16 bits. يعبّر الرقم الأول في الجداء عن عدد المواضع في الذاكرة والعدد الثاني يمثل طول الموضع الواحد. يتم التعبير عن ذاكرة RAM بشكل جداء باستثناء حالة واحدة إن كان طول كلمة الذاكرة قياسياً أي 8 bits لا داعي لكتابة الجداء. باختلاف ترتيب مصفوفة الذاكرة، تختلف التركيبة العامة للذاكرة. فمثلاً، إذا تم اعتماد الترتيب 16x 16 لخلايا التخزين بالمصفوفة، كما هو مبين في الشكل 7-2.

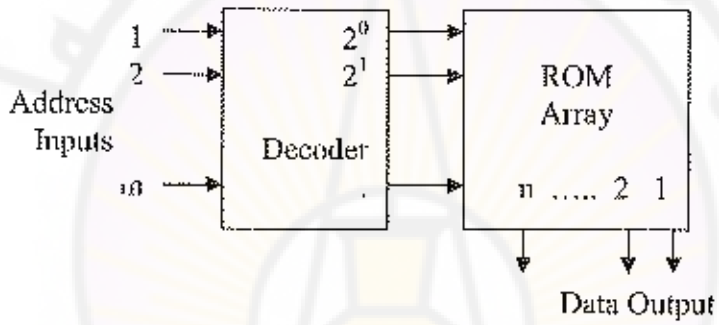


شكل 7-2: المخطط الصندوقي لذاكرة RAM 16 x 16

يعني ذلك وجوب توافر أربعة خطوط عنونة A_0, A_1, A_2, A_3 لتحديد 16 صف بالمصفوفة عن طريق مفكك ترميز 4x16 وأربعة خطوط عنونة A_4, A_5, A_6, A_7 لتحديد عدد أعمدة المصفوفة وهو 16 عن طريق مفكك ترميز 16 x 4 أيضاً، ليصبح إجمالي عدد خطوط العنونة المطلوب ثمانية والتي يمكن بوساطتها عنونة 2^8 أي ما يعادل 256 bits وهو نفس ناتج 16 x 16.

تستخدم ذواكر RAM الديناميكية عدداً أقل من الترانزستورات في بناء خلية الذاكرة، وبالتالي فهي تمتاز عن RAM الساكنة بان كثافة تغليفها عالية، وان تكلفتها أقل. أما ذواكر RAM الساكنة فتمتاز بسهولة الاستخدام، لأنها لا تحتاج إلى دارات إضافية مثل دارات الإنعاش التي تستخدم مع ذواكر RAM الديناميكية. كما وتمتاز بأن استهلاكها للطاقة أقل بكثير من ذواكر RAM الديناميكية.

*- ذواكر القراءة فقط ROM من Read Only Memories وهي ذواكر مخصصة للقراءة فقط ولا يمكن الكتابة فيها. يبين الشكل 8-2 التركيبة العامة لذاكرة من نوع ROM، حيث يتضح من الشكل أن الأجزاء الرئيسية لذاكرة ROM هي:



شكل 8-2: المخطط الصندوقي لذاكرة من نوع ROM

- دارة فك الترميز التي تستخدم لتحديد مواقع الذاكرة المختلفة.
- دارة المصفوفة التي تمثل ساحة التخزين.

نتعامل مع الذواكر على مختلف أنواعها بهذا المقرر باعتبارها دارات دعم للمعالجات وأن

1 - الذواكر غير الدائمة Volatile Memories تفقد محتوياتها بمجرد انقطاع التغذية الكهربائية وهي تقابل النوع RAM وتقسم بدورها إلى:

- ذواكر استاتيكية SRAM، تخزن فيها المعلومات داخل تشكيل من الترانزستورات التي تمر بحالتين مستقرتين (0 أو 1) وتبقى بالوضع الذي تستقر به طالما هناك وجود للتيار الكهربائي.

- ذواكر ديناميكية DRAM ، تخزن فيها المعلومات وتحفظ على هيئة شحنة كهربائية بمكثف. لكن بالنظر للتسريب الذي يمكن أن يحدث في المكثفات يلزم تعريض هذا النوع من الذواكر إلى دورات إنعاش Refresh لتعويض جزء الشحنة المتسرب من المكثف على فترات زمنية تتفاوت بحسب نوع الذاكرة، علماً أن هذه الفترات لا تتجاوز عند أي منها بضعة ميلي ثانية.

2- الذواكر الدائمة Nonvolatile Memories وهي ذواكر تحتفظ بمحتوياتها حتى ولو انقطعت التغذية الكهربائية عنها، وهي تقابل الذواكر ROM. يوجد عدة أنواع من ذواكر ROM ويأتي هذا التعدد في الأنواع من تعدد طرق برمجتها. ومن طرق البرمجة نذكر:

- برمجة القناع Mask Programming وتسمى ذواكر ROM ذي القناع. تتم برمجة هذا النوع من الذواكر أثناء عملية تصنيع الذاكرة.

- البرمجة لمرة واحدة فقط وتسمى بذواكر PROM من Programmable ROM يستخدم هذا النوع من الذواكر التي تبرمج من قبل المستخدم في التطبيقات التي يرغب فيها المستخدم برمجة بعض الأوامر الخاصة بالتطبيق الخاص به لمرة واحدة من دون أي تعديل على محتواها بعد برمجتها. تصنع ذواكر PROM خالية من المعلومات (أي أن محتوياتها تكون بالكامل واحدات)، وهي تحتوي على وصلات يمكن تحويلها من واحدات إلى أصفار عند البرمجة، بالتالي عملية البرمجة تعني تحويل بعض الواحدات فيها إلى أصفار وترك بعض الواحدات كما هي، بحيث بعد الانتهاء من عملية البرمجة تصبح المعلومات ثابتة لا تتأثر بوجود التغذية الكهربائية

- البرمجة لعدة مرات من قبل المستخدم وهي تقابل ذواكر قابلة للبرمجة والمحو. ونميز بين نوعين هما:

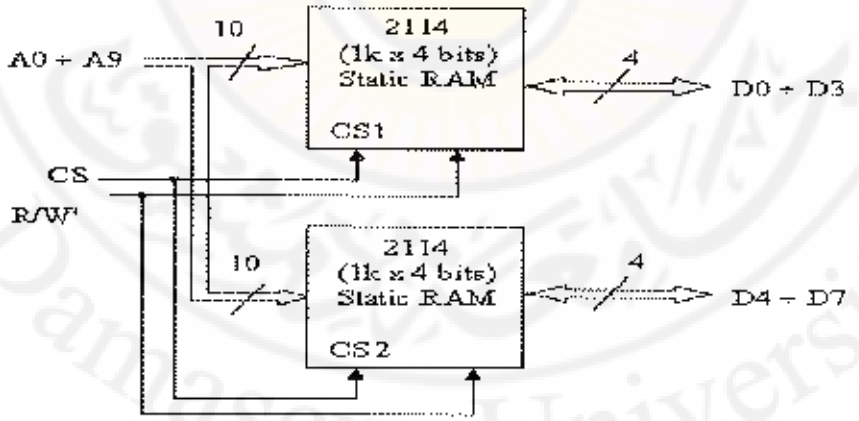
*- ذواكر EPROM من Erasable Programmatically ROM ويمكن إعادة هذا النوع من الذواكر إلى الحالة الابتدائية حتى ولو كانت مبرمجة أكثر من مرة، وذلك بتعريضها للأشعة فوق البنفسجية Ultraviolet Light عند كل مرة نريد فيها مسح محتوياتها وإعادة برمجتها من جديد.

*- ذواكر EEPROM من Electrically-Erasable PROM ويمكن بهذا النوع من الذواكر مسح الوصلات المبرمجة بوساطة إشارة كهربائية، حتى من دون الحاجة لإخراجها من القاعدة التي تكون متوضعة بها.

٢-٣-٢ زيادة طول كلمة الذاكرة

يوجد لبعض الاعتبارات ذواكر مصنعة على شكل دارات متكاملة بطول كلمة غير قياسي، مثل الدارة 2114 مثلاً المنظمة وفق الشكل 1 K x 4bit.

يلزم ببعض التطبيقات زيادة طول كلمة الذاكرة من هذا النوع ليصبح بطول 8 bit. لبناء ذاكرة بحجم 1 KB باستخدام هذه الدارة مثلاً (أي ذاكرة بسعة 1KB وبطول كلمة قياسي 8 bits)، يلزم ربط أكثر من دارة مع بعضها البعض (تحديداً دارتان) لتحقيق المطلوب. يمكن تحقيق ذلك، وفق ما هو مبين في الشكل 9-2، حيث توصل خطوط العنونة اللازمة لعنونة 1KB وهي من A_0 وحتى A_9 بشكل قياسي على الدارتين بأن واحد لعنونة الموضع المطلوب، كما توصل إشارة التحكم CS إلى الدارتين معاً، لأنه يفترض انتقاء الدارتين بنفس اللحظة، وهذا ينطبق أيضاً على إشارة التحكم R/W'. تترك خطوط المعطيات الأربع لكل من الدارتين بشكل مستقل بعد إعادة ترميزها في الشكل D_0 وحتى D_3 للدارة الأولى و D_4 حتى D_7 للدارة الثانية.

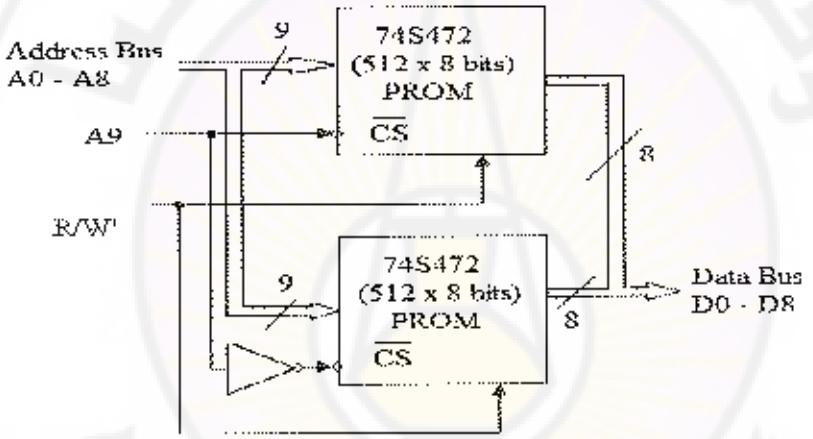


شكل 9-2: زيادة طول كلمة الذاكرة

عند تفعيل الإشارة CS (من أي منطق تحكمي خارجي) يتم انتقاء الدارتين بآن واحد ويشكل خرج الدارة الأولى بطول 4bits القسم السفلي من الكلمة وخرج الدارة الثانية بطول 4bits القسم العلوي، ليصبح الطول الإجمالي للكلمة من الدارتين معاً 8 bits.

٢-٣-٣ زيادة حجم الذاكرة

يلزم ببعض التطبيقات زيادة حجم الذاكرة من دون التغيير في طول الكلمة. فمثلاً، باعتماد الدارة ذات الرقم 74S472، التي تمثل ذاكرة PROM بحجم 1/2 KB وبطول قياسي للكلمة، يطلب تشكيل ذاكرة بحجم 1KB. يكون الجواب بتوصيل دارتان مع بعضهما وذلك بوضعهما فوق بعض بشكل عمودي، وفق ما هو مبين في الشكل 10-2



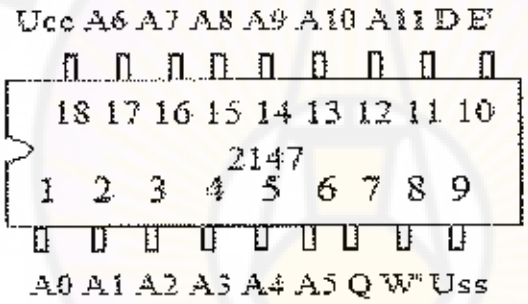
شكل 10-2: زيادة حجم الذاكرة

يتضح من الشكل أعلاه، انه في لحظة واحدة سيتم التعامل مع دارة واحدة من الدارتين الموجودتين في الشكل، لان قيمة إشارة CS الموصولة لإحداهما هي بعكس قيمة نفس الإشارة الموصلة للثانية. ثم وصل ممرات DB لكلا الدارتين معاً، لان طول كلمة كل منهما قياسية.

وبما أن حجم القطعة الواحدة من الذاكرة 74S472 هو نصف كيلوبايت، يلزم لعنونتها تسعة خطوط من A_0 وحتى A_8 توصل بشكل قياسي على كل من الدارتين. لتشكيل 1KB يلزم نصفي كيلوبايت (أي دارتان)، لذا تم إضافة خط عنونة عاشر هو A_9 وتم وصله

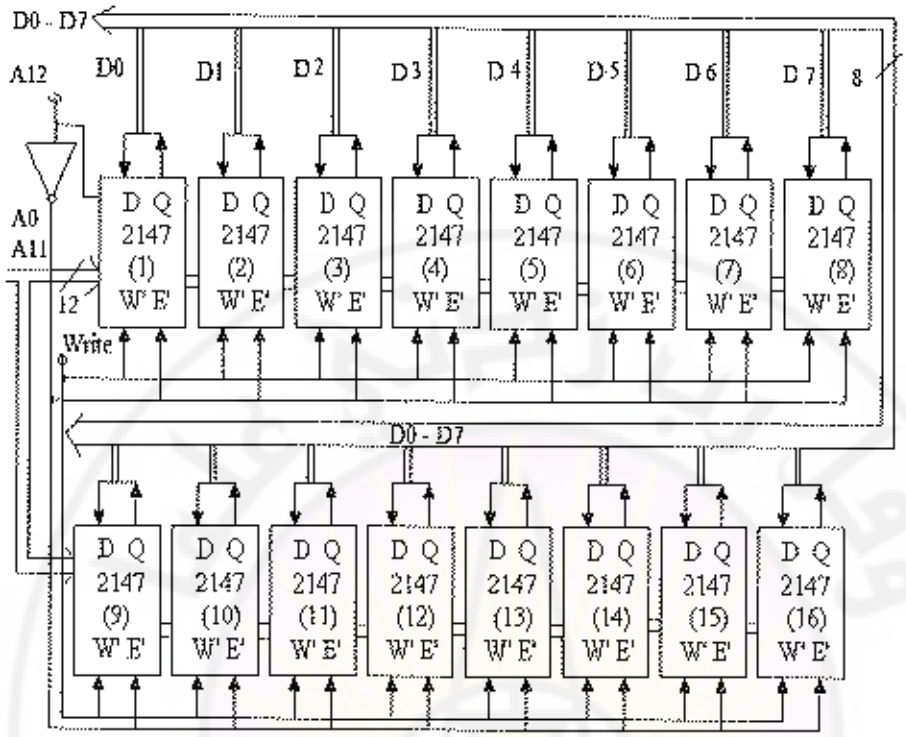
وفق ما هو مبين في الشكل أعلاه، بحيث عندما $A_9 = 0$ سيكون $CS' = 0$ (فعّال) وهذا يعني أنه تم انتقاء الدارة الأولى، والثانية ستكون بسبب وجود الـ NOT الذي يحول الـ CS' العائد لها إلى الواحد خارج الخدمة، وهذا يمثل النصف الأول من الكيلوبايت المطلوب. بمجرد أن يصبح $A_9 = 1$ تفعّل الدارة الثانية أي النصف الثاني من الكيلو وتكون الدارة الأولى خارج الخدمة، وبهذا الشكل يمكن الحصول على 1 KB من دارتين حجم كل منهما نصف كيلوبايت.

يمكن زيادة كل من طول كلمة الذاكرة وحجمها بآن واحد. وفيما يلي نعرض مثالا محدّداً عن ذلك باستخدام دارة مثل 2147، التي تمثل ذاكرة RAM بالتنظيم 4k x 1bit، حيث إن توزع إشارات الدارة على مغار زها وفق ورودها في كتالوجك الشركة الصانعة مبين في الشكل 11-2.



شكل 11-2: توزع مأخذ الدارة المتكاملة 2147

وليكن المثال: تصميم كتلة ذاكرة (بطاقة) بسعة 8 KB باستخدام الدارة 2147 ؟
الجواب: بما أن طول كلمة كل قطعة ذاكرة هو 1 bit يلزم ثماني دارات لجعل طول الكلمة قياسياً أي 8 bits . وبالتالي يلزم لتشكيل أول 4 KB وصل ثماني دارات مع بعضها. وبنفس الأسلوب يلزم لزيادة الحجم إلى القيمة المطلوبة نفس القدر من الدارات، أي يلزم 16 دارة لتحقيق المطلوب، توصل مع بعضها البعض وفق ما هو مبين في الشكل 12-2



شكل12--2: كرت ذاكرة بسعة 8 KB باستخدام الدارة 2147

وبهذا الشكل، عندما $A_{12} = 0$ يتم انتقاء أول ثماني قطع التي تشكل أول 4 KB، وعندما يصبح $A_{12} = 1$ يتم انتقاء ثاني ثماني قطع التي تشكل الـ 4 KB الثانية. بالطبع وصل خطوط المعطيات والعنونة لكل قطعة من قطع الذاكرة في الشكل أعلاه يتم بشكل قياسي. وكذلك الحال بالنسبة لإشارات التحكم الأخرى.

٢-٣-٤ تصميم كتلة ذاكرة

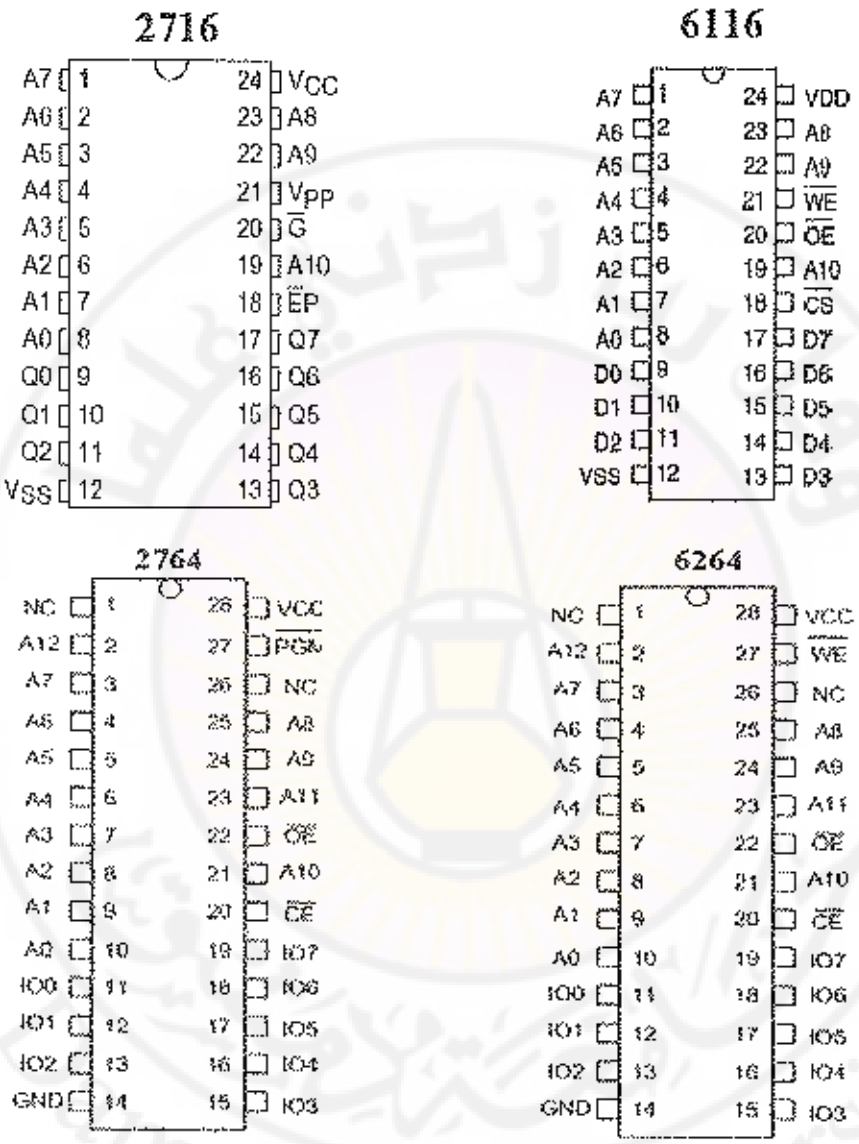
بداية نعرض في الجدول 1-2 أرقام بعض الذواكر المصنعة على شكل دارات متكاملة (PROM, EPROM, SRAM, DRAM) بتنظيمات مختلفة وأنواع مختلفة لاستخدامها لاحقا بأمثلة متفرقة.

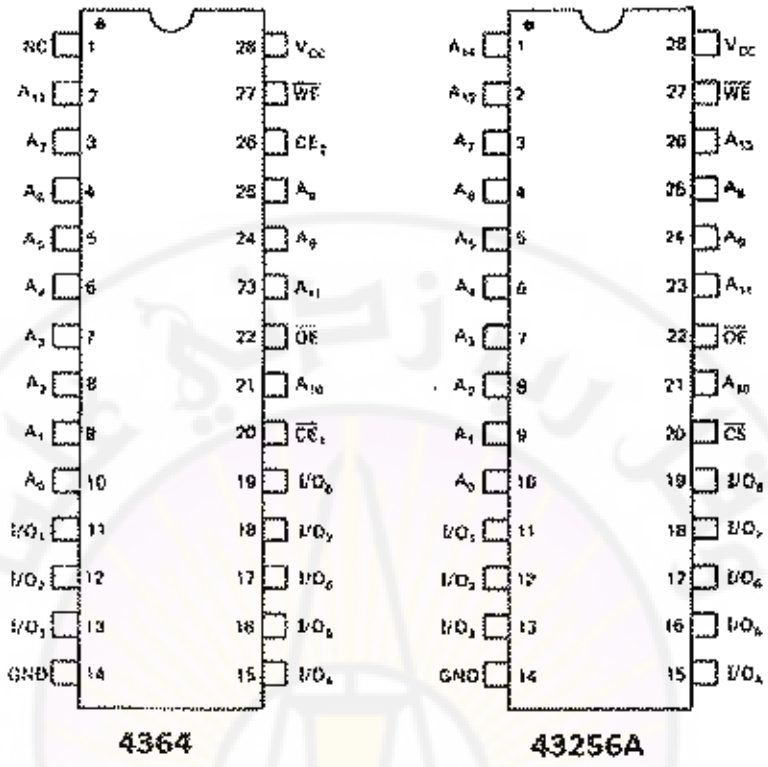
جدول 1-2: أرقام دارات متكاملة تمثل ذواكر مختلفة (RAM & EPROM)

رقم الدارة	تنظيم الذاكرة	نوع الذاكرة
74S287	256 x 4 bit	PROM
82S113	(256 x 8 bit)	
8608	(1k x 8 bit)	
8708	(1k x 8 bit)	EPROM
2716	(2k x 8 bit)	
2732	(4k x 8 bit)	
2764	(8k x 8 bit)	
27128	(16k x 8 bit)	
2102	(1k x 1 bit)	Static RAM
2114	(1k x 4 bit)	
2147	(4k x 1 bit)	
4361	(64k x 1 bit)	
4363	(64k x 4 bit)	
6116	(2k x 8 bit)	
6264	(8k x 8 bit)	
4364	(64k x 8 bit)	
43256	(32k x 8 bit)	
4116	(16k x 1 bit)	Dynamic RAM

يطلب في بعض التطبيقات التقيد بنوع محدد من الدارات، نظرا لعدم توفر ماقد يلزم من أنواع. بالتالي يجب التركيز على المبادئ العامة التي يفترض إتباعها خلال عملية التصميم متجاوزين بذلك خلال تصميم كتلة ذاكرة بعض الاعتبارات المتعلقة بها.

ويظهر الشكل 13–2 تغليف وتوزع إشارات بعض هذه الذواكر كثيرة الاستخدام.





شكل13-2: توزع إشارات بعض ذواكر Static RAM و EPROM

مثال: يطلب تصميم بطاقة ذاكرة بحجم 16 KB ربعها قابل للقراءة فقط؟

الجواب: بحسب نوع الذواكر التي ستستخدم في التصميم، قد تكون الخيارات واسعة. فإذا تقيدنا باستخدام دارات من الجدول أعلاه، يمكن بقصد التبسيط اختيار الدارة 2716 باعتبارها ذاكرة للقراءة فقط (EPROM بسعة 2KB) والدارة 6116 للقراءة والكتابة (أي RAM بسعة 2KB أيضاً).

بما أن كلا من الذاكرتين اللتين تم اختيارهما بالتنظيم (2K x 8bits)، يلزم من EPROM قطعتان لتشكيل حجم 4KB، بينما يلزم من RAM ستة قطع بحجم إجمالي قدره 12 KB وفق ما هو مطلوب.

وبهذه الحالة التي يطلب فيها عنونة ذواكر بأنواع متعددة وبأحجام مختلفة، يستخدم مفكك الترميز لتشكيل العنوان المناسب من ممرات العناوين لكل قطعة على حدة. يفضل اختيار

القطعة المطلوب التعامل معها في لحظة ما وفق عنوان محدّد ووضعها بالخدمة فقط عندما سيتم التعامل معها بقراءة معلومة ما من إحدى المواضع أو الكتابة بها وليس تفعيل كافة القطع دفعة واحدة. لفعل ذلك نبيّن فيما يلي، آلية العنونة للذاكرة باستخدام مفكك الترميز وتحديد عنوان كل قطعة من الذواكر على حدة. باستخدام مفكك ترميز بالتركيبة 3x8، يمكن بوصل خطوط العنونة A_0, A_1, A_2 كمدخل لمفكك الترميز الحصول على ثمانية مخارج هي: CS7 , ... ,CS0، يمكن لكل خرج من مفكك الترميز أن يقابل قطعة واحدة، بحيث القطعة الأولى التي يتم انتقاؤها بالخط CS0 تقابل التركيبة 000 على مداخل مفكك الترميز $A_2A_1A_0$، والقطعة الثامنة التي يتم انتقاؤها بـ CS7 تقابل العنوان $A_2A_1A_0$ = 111. أما لو كانت مداخل مفكك الترميز موصولة إلى خطوط العنونة A_1, A_2, A_3 بدلا من A_0, A_1, A_2 سيقابل أول خرج لمفكك الترميز العنوان $A_3A_2A_1$ = 000 ولكن قيمة A_0 الأدنى يمكن أن تكون القيمة 0 أو 1، أي d. بالتالي مجال كل خرج هو قطعتين، واحدة بالعنوان 0000 والثانية بالعنوان 0001. وهكذا إذا كانت مداخل مفكك الترميز خطوط العنونة $A_{13}A_{12}A_{11}$، يحسب مجال الخرج كالتالي: بما أن خط العنوان الأعلى A_{13} يسمح بعنونة ذاكرة بحجم 16K، وأن عدد مخارج مفكك الترميز المستخدم هو 8، سيكون وزن كل خرج 16K/8 = 2K. ويبين الجدول2-2 مجال العنونة لكل دارة. تم الانطلاق برسم مخطط الذاكرة من أن الذاكرة EPROM تبدأ عند العنوان 0H من المعالج والذاكرة RAM ستقع بعدها مباشرة من دون أي فراغ بالمجال العنونى للمعالج الذي يتعامل معها.

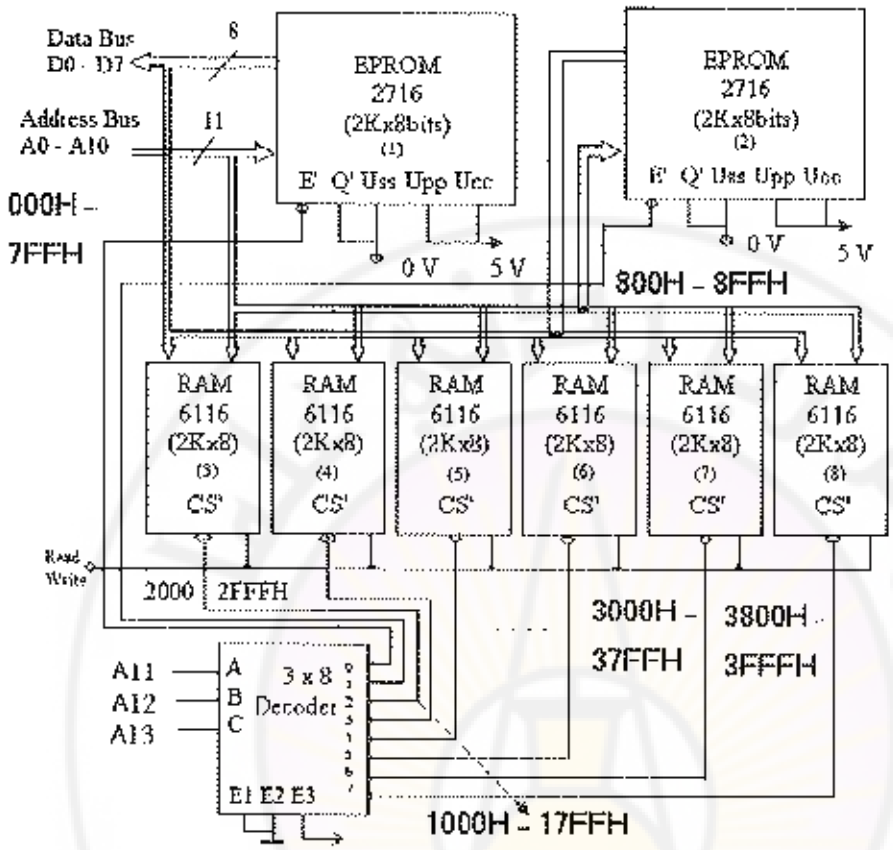
والشكل العام للبطاقة المقابلة وفق هذا التصميم مبين في الشكل 14-2. بالطبع كان بالإمكان اختيار دارات أخرى من الجدول أو خارجه وتعديل شكل التصميم، حيث باستخدام ذواكر بتنظيم غير قياسي يتطلب في المرحلة الأولى تأمين الشكل القياسي لكل نوع من الذواكر، وفي المرحلة الثانية إتمام عملية التصميم.

جدول 2-2: مخطط عنونة ذاكرة بسعة 16 KB ربعها قابل للقراءة

رقم الدارة	عنوان القطعة	العنوان ضمن قطعة الذاكرة			نوع الذاكرة
	$A_{13}A_{12}A_{11}$	$A_{10}A_9A_8$	A_7--A_4	A_3--A_0	
CS0' ~ 1	000	000	0000	0000	EPROM
	000	111	1111	1111	
CS1' ~ 2	001	000	0000	0000	EPROM
	001	111	1111	1111	
CS2' ~ 3	010	....	....	....	RAM
...	...	...	...	...	...
CS5' ~ 6	101	....	....	....	RAM
CS6' ~ 7	110	....	....	....	RAM
CS7' ~ 8	111	000	0000	0000	RAM
	111	111	1111	1111	

تتركز عملية التصميم على إيجاد إشارة الـ CS من خطوط العنونة لكل دارة على حدة. يلزم في المثال أعلاه لكل من الذاكرتين 12 خط عنوان من A_0 وحتى A_{11} ، وكذلك يوصل لكل قطعة ثمانية خطوط للمعطيات $D_0 - D_7$ بشكل قياسي إلى ممرات المعطيات مع الانتباه إلى أن هذه الممرات ستكون خارجة من EPROM، كما توصل الإشارات الأخرى مثل Read أو Write إلى ممرات التحكم.

تخصص خطوط العنونة من $A_0 - A_{11}$ للعنونة ضمن القطعة، أما لتحديد القطعة التي سيتم التعامل معها في لحظة ما، تستخدم خطوط عنونة جديدة بعد الخط A_{11}.



الشكل14--2: كرت ذاكرة EPROM & RAM بسعة إجمالية 16 KB

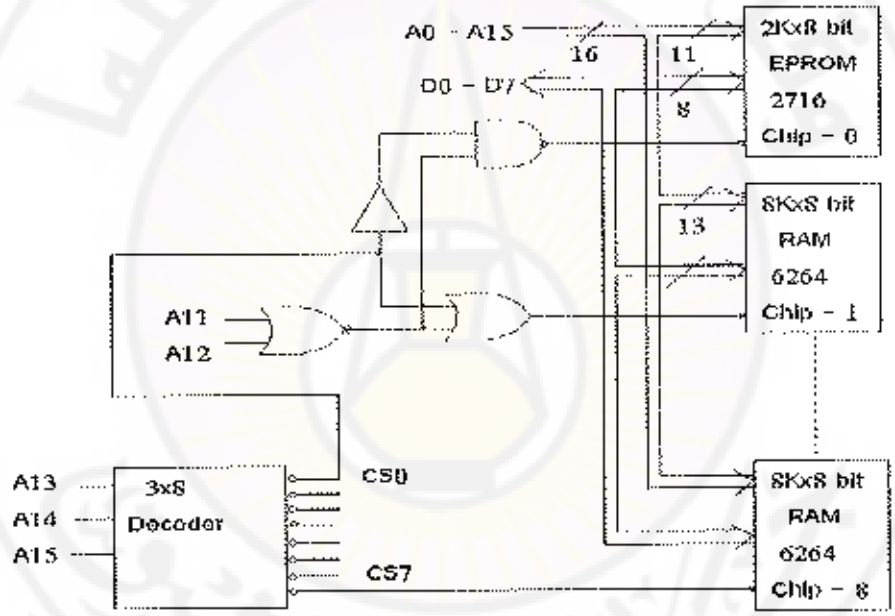
٢-٣-٥ الفراغ في المجال العنوني

تصادف مشكلة الفراغات في المجال العنوني عند الالتزام بتصميم كتلة ذاكرة باستخدام ذواكر متعددة وبأحجام متباينة. ولتوضيح هذه المسألة، نعرض المثال التالي.

مثال: لنفترض بأنه طلب تصميم كتلة ذاكرة بسعة 2 KB EPROM و 62 KB RAM لربطها بمعالج مجاله العنوني 64 KB، وأن الذواكر المتوافرة لهذه الغاية هي الدارة 2716 التي تمثل EPROM، والدارة 6264 التي تمثل RAM .

الجواب: لعنونة 64 KB يلزم كامل المجال العنوني للمعالج، بالتالي عند استخدم مفكك ترميز من الشكل 3x8 يفترض استخدام خطوط العنونة من A_{13} وحتى A_{15} لعنونة كامل

المجال وبقية خطوط العنونة A_{12} وحتى A_0 للعنونة ضمن الدارات. من طرف آخر يلزم 8 قطع من الدارة 6264 لتأمين 62K ويكفي دارة واحدة من 2716 لتأمين 2K من الذاكرة EPROM، أي يلزم 9 قطع ذاكريه، علما أن خرج الـ Decoder يغطي ثماني دارات فقط، وبالتالي المشكلة تكمن في وزن خرج مفكك الترميز الذي سيشكل فراغ في المجال العنوني باعتماد وزن 8 K لدارة بحجم 2K، أي سيبقى 6K من إحدى الخطوط معنونة بدون وجود ذاكرة. لتجاوز مثل هذه المشكلة لابد من اعتماد منطق تحكمي إضافي يسمح بربط نوعين مختلفين من الذواكر على نفس خط الانتقاء CS وبالتالي يمكن أن يكون التصميم وفق ما هو مبيّن في الشكل 15-2.



شكل 15-2: مخطط ذاكرة يظهر آلية إلغاء الفراغ في المجال العنوني

وتكون العناوين المقابلة لكل قطعة ذاكرة وفق ما هو مبين بمخطط الذاكرة بالجدول3-2. حيث تم في التصميم الانطلاق من أن أول 2 KB ذاكرة ستكون مخصصة لـ EPROM وهي تأخذ، وفق ما هو مبيّن في الجدول، المجال من 0000H وحتى 07FFH.

بعد ذلك الذاكرة من العنوان 0800H وحتى 1FFFH التي تعادل حجم 6 KB تخصص للذاكرة RAM من القطعة الأولى، حيث يبقى أول 2 KB من القطعة الأولى 6264 فارغاً من الناحية الفيزيائية مع وجود اتصال بين الذاكرتين EPROM و RAM منطقياً.

جدول3-2: مخطط ذاكرة لـ 64 KB بمعدل 2 KB EPROM و 62 KB RAM

$A_{15}A_{14}$	A_{13}	A_{12}	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	
0 0	0	0	0 0 0 0	0 0 0 0	0 0 0 0	~ 0000H
.....						
0 0 0 0	0	1	1 1	1 1 1 1	1 1 1 1	~ 2 KB
0 0 0 0	1	0	0 0	0 0 0 0	0 0 0 0	
.....						
0 0 0 1	1	1	1 1	1 1 1 1	1 1 1 1	~ 6 KB
0 0 1 0	0	0	0 0	0 0 0 0	0 0 0 0	
.....						
0 0 1 1	1	1	1 1	1 1 1 1	1 1 1 1	~ 8 KB
.....						
1 1 1 0	0	0	0 0	0 0 0 0	0 0 0 0	
1 1 1 1	1	1	1 1	1 1 1 1	1 1 1 1	~ FFFF

ويتضح من مخطط العنونة أنه طالما $A_{11} = 1$ أو $A_{12} = 1$ سيكون العنوان خارج مجال EPROM، أي في RAM. يتم تحقيق ذلك باستخدام منطق يسمح في حال $CS0' = 0$ وأن $A_{11} = A_{12} = 0$ الولوج إلى الذاكرة EPROM فقط. بمجرد أن يصبح أي من A_{11} أو A_{12} بالقيمة 1 يصبح الولوج إلى الذاكرة RAM.

٢ – ٤ المعالجات الصغرية

يمتاز المعالج الصغري بصغر الحجم والوثوقية العالية والقدرة على إجراء الحسابات والتحكم. تتطور المعالجات بشكل عام مع مرور الزمن بشكل كبير وسريع لدرجة أن المعالج ربما يكون هو أكثر أجزاء الحاسوب سرعة في التطور. نميز بين شكلين لوجود المعالج الصغري هما:

١ – معالجات الشريحة الواحدة

يتشكل كل معالج من معالجات هذه المجموعة من شريحة الكترونية واحدة يركب عليها العديد من الدوائر الرقمية القادرة على تنفيذ التعليمات الموجودة في برنامج، ويصنع كامل المعالج على شكل دارة متكاملة واحدة تضم كافة مكوناته الأساسية، وتسمى بمعالجات وحيدة البلورة Monolithic Microprocessor.

يمكننا النظر إلى أن أي من المعالجات الحديثة وحيدة البلورة يمكن أن يتشكل من سبعة أقسام أو وحدات رئيسية هي: وحدة التحكم، ووحدة المنطق الرياضي، ووحدة متابعة تنفيذ الأوامر، ووحدة فك التشفير، ووحدة كود الناقل، ووحدة ما قبل المعلومات، والذاكرة الفورية Cache Memory وأن التفاوت بين أي معالج وآخر يتوقف على مواصفات كل من هذه الأقسام السبعة.

يتعامل المعالج الصغري مع البيانات على شكل بتات وهو لا يفكر من تلقاء نفسه بل يحتوي على وحدة تحكم تقوم بفك شفرة الأمر وإصدار إشارات التحكم المقابلة، كما ويحتوي على دوائر رقمية تستخدم في تخزين معلومات، يطلق عليها اسم المسجلات.

٢ – المعالجات متعددة الشرائح

يصادف نوع آخر من المعالجات المصنعة من أكثر من شريحة تسمى بالمعالجات الشرائحية Bit Slice Processor، حيث إن البعض يصنفها مع المعالجات العادية والبعض الآخر يصنفها كا Mocro logic. نكتفي هنا بالإشارة إلى معالجات هذه الفئة من خلال مثال محدد دون الدخول بمزيد من التفاصيل، لأنها ليست موضوع دراستنا في هذا الكتاب.

فمثلا شركة إنتل بالإضافة إلى المعالجات وحيدة البلورة (مثل 8080) المصنع بتراكيب من التقانة MOS أنتجت أيضاً معالجات شرائحية مصنعة بناء على تراكيب من التقانة TTL وهذا يوحي بأن عناصر هذه السلسلة تمتاز بسرعة عالية مقابل الانخفاض في درجة التكامل، والوثوقية والزيادة في استهلاك الطاقة الكهربائية.

ظهرت عائلة المعالجات الشرائحية 3000 من Intel لأول مرة عام 1973 وتضم هذه العائلة عناصر يمكن استخدامها لبناء معالجات بعرض بيانات قابل للزيادة بـ 2، بمعنى أنه يمكن بناء معالجات بطول كلمة 2, 4, 8 bit وهكذا.

يركب المعالج بالسلسلة 3000 من عدة شرائح وبعض الدارات المتممة ، حيث إن عدد الدارات يرتبط بطول الكلمة والتركيبة المطلوبة. العنصر الرئيسي بهذه العائلة من شركة إنتل هو العنصر (الشريحة) Intel 3002 الذي يمثل عنصر معالجة مركزي CPE (Central Processing Element) وهو مكون من ALU بعرض 2 bit ويحتوي على 11 مسجلا ومراكماً. تستطيع الشريحة CPE تنفيذ عمليات حسابية ومنطقية، إضافة لعمليات الإزاحة والاختبار وغيرها.

يمكن ترتيب أربع وحدات ALU (كل منها بطول 2 bit) بجانب بعضها مع إشارات تحكم لتشكيل وحدة CPU بعرض 8 bit (أي أن المصمم يمكن أن يضيف عدداً من الشرائح وفق طول الكلمة الذي يتطلب التعامل معها).

العنصر 3002 لا يستطيع جلب تعليمة من الذاكرة، إنها مهمة وحدة التحكم بالبرنامج المايكروي MCU 3001 (Microprogram Controller Unit) الذي يستطيع عنونة حتى 512 كلمة من ذاكرة البرنامج، تظهر كمصفوفة من 32 سطر و 16 عمود. تستخدم العائلة Intel 3000 ترانزستورات شوتكي Bipolar Schottky ثنائية القطبية، ويصنع عناصر عدة منها كدارات دعم مثل مفكك الترميز 3205.

وبالعودة إلى المعالجات وحيدة البلورة، نشير إلى أنه أصبح يوجد في أيامنا هذه العديد من المعالجات ومن شركات عدة، لكن نقتصر في عرضنا ودراستنا للمعالجات في هذا الكتاب فقط على المعالجات المصنعة من قبل شركة إنتل Intel نظرا لتوافقها مع حواسيب IBM ووجودها بالسوق المحلية.

يلزم للتعريف بأي معالج صغري أياً كان نوعه بالحد الأدنى كتاب كامل، وبالتالي لا يمكننا التعريف بهذا الكم الهائل من أنواع المعالجات المتوفرة بالوقت الحاضر ومن شركات متعددة بكتاب واحد. يوجد بعض المفاهيم الأساسية التي نصادفها عند تعرّف أي نوع من المعالجات الصغرية، نتطرق لها فيما يلي باختصار قبل المباشرة بتعرّف نوع محدد من المعالجات ومن هذه المفاهيم:

٢-٤-١ تطور المعالجات الصغرية

ظهر أول معالج صغري من شركة Intel في العام 1971 تحت الاسم I4004 وكان لهذا المعالج تركيبة معالج متوازٍ بطول 4 bits وهو مصنع بالتقانة PMOS. بعد عام من ظهور أول معالج من شركة إنتل ظهر المعالج I8008 وكان بالمقارنة مع المعالج I4004 أفضل بكثير من حيث البنية الداخلية ودارات الدعم وجملة التعليمات. ظهر بالعام 1974 من شركة Intel معالجات الجيل الثاني متمثلة بالمعالج I8080 المصنع بالتقانة NMOS حيث كان أسرع بعشر مرات على الأقل من المعالجات المصنعة بالتقانة PMOS . بعد شركة Intel ظهرت شركات أخرى لإنتاج المعالجات مثل شركة Motorola و Zilog وغيرها. ومن أشهر المعالجات التي ظهرت من الشركات الأخرى بفترة ظهور المعالج I8080 نذكر المعالجين MC 6800 & Z-80.

قامت شركة إنتل على أثر ظهور بعض معالجات الجيل الثاني من بعض الشركات الأخرى بعرض المعالج 8085 بالتركيبة MCS-85 بقصد المنافسة وتغطية العيوب التي كانت موجودة بالمعالج I8080 وبخاصة على صعيد النقل التسلسلي للمعطيات ومنظومة المقاطعة.

امتازت معالجات الجيل الثاني بالمقارنة مع معالجات الجيل الأول باستخدام ممرات منفصلة، بمعنى ممر واحدة للعناوين وآخر للمعطيات وثالثة للتحكم. ومع تطور المعالجات تطورت أيضاً دارات الدعم المتممة لكل منها.

ظهرت معالجات الجيل الثالث بطول كلمة 16bit بعد منتصف العام 1976 وقد ظهر من شركة إنتل، المعالج I8086 وكذلك المعالج I8088 وقد سمي هذين المعالجين مع

المعالجات التي ظهرت بعدهما بالاسم I8086/8088 Family نظراً للتوافق بينها. يعد المعالج 8086 أول أفراد العائلة 8086 Family وهو مصنع بالتقانة HMOS. يستطيع كل من المعالجين 8086/8088 عنونة ذاكرة بسعة 1 MB ويمتاز كل منهما بالعمل في النمط الحقيقي فقط Real--mode أي تنفيذ برنامج واحد بلحظة واحدة. ظهر بعد المعالجين 8086/8088 مباشرة معالجين ببعض المواصفات الفنية الإضافية هما المعالجين 80186/80188. ومنذ ذلك الحين يمكن تقسيم المعالجات التي تنتجها شركة Intelوالتي استخدمت في نظم الحواسيب الشخصية المتوافقة مع IBM إلى عدة أجيال، حيث كل جيل كان يتوج على عرش معالجات الحواسيب الشخصية مدة تقارب الثلاث سنوات قبل الجيل التالي، ومن هذه المعالجات:

المعالج 80286 وهو بطول كلمة 16bits قادر على عنونة ذاكرة بسعة 16MB ويعمل بالنمطين الحقيقي والمحمي الذي يعني نظام تعدد المهام Multi tasking. ظهر المعالج 80386 بعد المعالج 80286 وهو أول معالج بطول كلمة 32bits ويستطيع أن يعنون ذاكرة بسعة 4GB ويعمل بالنمطين الحقيقي والمحمي. المعالج 80486 هو أيضاً معالج بطول كلمة 32bits وقد صمم ليعمل بفعالية أكبر. وأخيرا ظهر المعالج Pentium الذي يقابل معالج بالرقم 80586 وظهر له عدة اصدارات، يتميز كل منها عن الآخر بالإضافة لاسم المعالج بالتواتر الذي يعمل عليه المعالج وبعض الخصائص الإضافية. اعتمدت شركة Intel اسم Pentium بدلا من الرقم 80586 لامتلاك حقوق الإصدار Copyright التي تنسب لاسم ولا تجوز على الرقم. يتم استخدام المعالج لبناء نظام حاسوب بخصائص محددة كما في حالات استخدام المعالج 8088 لبناء الحاسوب من النوع IBM PC واستخدام المعالج 80286 لبناء الحاسوب المسمى (eXtended Technology) PC-XT كما تم بناء النظام AT (Advanced Technology) مع ظهور المعالج 80386.

وبالنظر لأهمية وضع نظم ثابتة ومعروفة للجميع ظهر بعد ذلك أنظمة ISA و EISA حيث ISA (Industry Standard Arch.) و (Extended ISA) EISA. مع ظهور المعالج Pentium ظهرت الحاجة لأنظمة جديدة ذات سرعة عالية فظهرت

أنظمة الناقل المحلي Local Bus Systems مثل نظام PCI ونظام VESA وذلك للاستفادة من الإمكانات الجديدة للمعالجات.

حافظت معالجات Intel على التوافقية بحيث يتم استيعاب وتنفيذ البرامج التي تمت كتابتها للمعالجات القديمة لتعمل في المعالجات الجديدة بدون مشاكل وهو ما يسمى بتوافقية البرامج ، حيث تم الاحتفاظ بالبرامج القديمة مع إمكانية تشغيل البرامج الجديدة.

٢-٤-٢ أنماط عمل المعالجات

أنماط العمل بالنسبة للمعالجات هي وصف للبيئة التي يعمل فيها المعالج من حيث قدرته على الوصول للذاكرة العشوائية وتشغيل أكثر من برنامج في نفس الوقت. إن نمط العمل لمعالج ما يتحدد بنظام التشغيل ونوع المعالج الذي يستخدمه.

يوجد كما سنجد لاحقا في القسم الثاني من هذا الكتاب أنماط مختلفة نتجت من تطور الحواسيب الشخصية من بدايتها. فالنمط الأول الذي يعني إمكانية التعامل مع برنامج واحد فقط قيد التشغيل في وقت واحد، يسمى بالنمط الحقيقي وقد ظهر مع ظهور المعالج 8086. أما النمط الثاني الذي يستخدم في كافة أنظمة التشغيل متعددة المهام الحديثة يسمى بالنمط المحمي وقد ظهر مع ظهور المعالج 80286 والنمط الثالث يسمى بالنمط الظاهري وهو في الواقع قدرة إضافية وتعزيز للنمط المحمي. يسمى النمط المحمي " نمط 386 المحسّن " لان معالج 80386 هو أول معالج سمح بالانتقال بين النمط المحمي والنمط الحقيقي بدون إعادة تشغيل الحاسوب.

٢-٤-٣ فولطية المعالجات

يحتاج المعالج كجهاز إلكتروني للكهرباء، تحديدا لتيار مباشر مثل التيار الذي ينتج من البطاريات، تعمل المعالجات المختلفة بفولطية مختلفة عن بعضها . يوجد على اللوحة الأم محول فولطية يوفر للمعالج الفولطية التي يحتاجها، لذا فإن أحد الأسباب التي تجعل اللوحة الأم تتحكم بنوع المعالج هو مقدار الفولطية التي يعمل عليها المعالج، وللفولطية دور مهم، حيث إن الفولطية الأعلى تعني زيادة درجة الحرارة مما يؤثر في عمل المعالج من حيث عمره الافتراضي واستقرار عمله ووثوقيته، ويولد مشاكل في التبريد.

تعني الفولطية في الحاسبات المحمولة استهلاك طاقة أعلى مما يعجل بنفاد البطارية. كانت أغلب المعالجات تعمل بفولطية 5 V ثم تم إنقاصها إلى 3,3V، وما لبثت الشركات أن قررت خفض الفولطية إلى أقل من ذلك بطريقة فصل الفولطية، أي أن تعمل الأجزاء المختلفة من الحاسوب بفولطيات مختلفة فأصبح المعالج يقسم لقسمين: الأول: وحدات الإدخال والإخراج وتعمل بـ 3,3 V

والثاني: قلب المعالج ويعمل بأقل من ذلك (حسب المعالج)

إن كمية الحرارة الصادرة من معالج مثل PentiumII تكفي لمنعك من وضع يدك على المعالج أثناء عمله لذا تحتاج المعالجات للتبريد ...

٢-٤-٤ معمارية المعالجات

يضم كل معالج صغري من المعالجات الحديثة ملايين الترانزستورات الصغيرة الحجم التي تتصل فيما بينها بأسلاك دقيقة للغاية. تتوضع الترانزستورات داخل المعالج الواحد بمساحة صغيرة جداً لدرجة أنها لا ترى بالعين المجردة وتكون موصلة مع بعضها البعض بأسلاك صغيرة جداً تضمن تدفق البيانات بين الترانزستورات، وتقاس سماكة هذه الأسلاك بالمايكرون، وسماكة هذه الأسلاك هي التي تحدد معمارية المعالج ، حيث إن معظم وحدات المعالجة المركزية الحالية تصنع باستخدام تقنية الـ 0,018 ميكرو. ويشير هذا الرقم إلى المسافة بين المكونات القريبة على سطح الشريحة (ضمن الدارة المتكاملة)، وهذه المسافة أصغر من سمك شعرة برأس الإنسان بمقدار 500 مرة تقريبا. تنظم الترانزستورات داخل المعالج لتكوين مجموعات (وحدات) ذات وظيفة معينة تختلف باختلاف ترتيب وتنسيق هذه الترانزستورات، وبذلك يمكننا تكوين أنواع عديدة من المجموعات. وفي كل مجموعة يختلف عدد الترانزستورات وطريقة تجمعها ما يؤثر في وظيفتها. وتصنيع المعالج ما هو إلا وضع هذه المجموعات وربطها ببعضها بالشكل المطلوب. فمعالج بنتيوم مثلا هو من رتبة 0,05 مايكرون بينما المعالج MMX بنتيوم معماريته 0,035 مايكرون، والمعالج PentiumII يستعمل معمارية 0,025 مايكرون. لقد نجحت شركة IBM بفضل نوع من التقنيات الجديدة بتطوير طريقة لصنع معالجات بمعمارية 0,013 مايكرون وهذا

من بين المكونات (الكتل) الأساسية التي سنصادفها بكل معالج صغري والمبينة بالشكل أعلاه نذكر :

- المراكم Accumulator اختصاراً ACC : وهو مسجل عرضه يساوي عادة طول كلمة المعالج، يتميز هذا المسجل عن غيره من المسجلات الأخرى بالاستخدام لبعض الأعمال المحددة، وهو من المسجلات التي هي بمتناول المستخدم عند القيام ببعض الأعمال البرمجية. كما أن لبعض المعالجات أكثر من مراكم.

- عداد البرامج PC – Program Counter : يتحدد عادة طول هذا المسجل بأكبر حجم ممكن للذاكرة التي يمكن عنونتها من قبل المعالج، وهو يحتوي عادة على عنوان خطوة التنفيذ التالية ، حيث يزداد محتوى هذا المسجل تلقائياً بعد تنفيذ كل تعليمة.

- مؤشر المكدس أو الخزان SP – Stack Pointer : ويحدد طول هذا المسجل بطول الذاكرة التي يمكن للمعالج أن يقوم بعنونتها وهو دوماً يشير إلى قمة المكدس. يستعمل المكدس عندما يصادف المعالج أثناء تنفيذ البرنامج أوامر قفز إلى برنامج فرعي ليحفظ محتوى عداد البرامج للرجوع إلى البرنامج الرئيسي عند الانتهاء من تنفيذ البرنامج الفرعي.

- Register Instruction & Decoding Circuits مسجل التعليمة ودوائر فك الترميز: وهي تمثل المتحكم بالمعالج (بقسميه) المخصص لحفظ الكود الخاص بالتعليمة وتفكيكها، أي تحديد نوع التعليمة.

- مسجلات أخرى: إضافة للمسجلات المبينة أعلاه يضم كل معالج مجموعة أخرى من المسجلات يختلف عددها من معالج لآخر وهي تستخدم لتسهيل وتسرع عمل المعالج لان التعامل معها أسرع بكثير من التعامل مع الذواكر.

- وحدة الحساب والمنطق ALU (Arithmetic Logic Unit): وهي من أهم المكونات الأساسية لأي معالج لأنها تقوم بمختلف العمليات الحسابية والمنطقية وهي تمثل جامعة بعرض كلمة يساوي عرض كلمة المعالج ويرتبط معها:

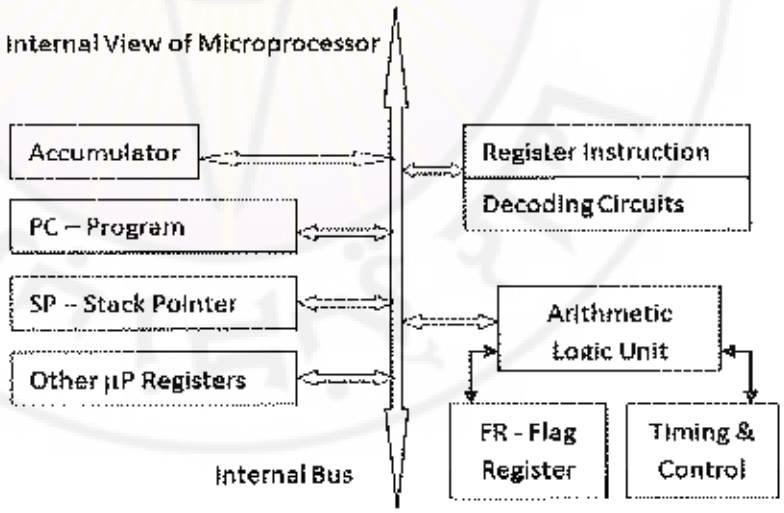
- دارات التوقيت التي تقوم بتأمين استمرار تدفق التسلسل السليم للوقائع.

- مسجل المؤشرات FR – Flag Register الذي يعكس ما يتم في وحدة ALU ويحتوي على عدد من المؤشرات الذي يختلف عددها من معالج لآخر.

عندما نود الإشارة إلى نوع حاسوب ما، فإننا نلجأ إلى نوع المعالج الصغري الذي بني وفقه الحاسوب والتواتر الذي يعمل عليه، كأن نقول " الجهاز هو PentiumIII 600 MHz ". وبالتالي عند تعرّفنا على أي معالج صغري لاستخدامه في تطبيق معين، ننظر إليه على أنه دارة متكاملة لابد قبل استخدامه من تكوين فكرة حول التركيبة الداخلية له ومفهوم كل من الإشارات الداخلة والخارجة منه لنتعرف آلية التعامل معه. نعرض فيما يلي بعض المفاهيم الأساسية والمشتركة التي يمكن أن نصادفها بمختلف المعالجات.

التركيبة الداخلية للمعالج الصغري

التركيبة الداخلية لمعظم المعالجات Internal Structure of Microprocessor هي قياسية، تضم المكونات الأساسية للمعالج من وحدة المعالجة والمسجلات ودارات التحكم وسنبدأ بتعرّف مفهوم بعض هذه المكونات التي نصادفها بكل معالج، ابتداء من المعالجات بطول كلمة 8 bits. علما أن المعالجات الأحدث تحتوي بالإضافة لهذه المكونات على المزيد من الكتل الأخرى التي يختلف عددها ومواصفاتها بحسب نوع المعالج. يظهر الشكل I-1 الكتل الأساسية التي يمكن أن نصادفها بمعالجات من رتبة 8 bit.



الشكل I-1: التركيبة الأساسية لمعالجات صغرية بكلمة 8 bits

Damascus University

الجزء الثاني: المعالجات الصغرية

ويضم إحدى عشر فصلا

الرابع: المعالج الصغري Intel 8080A

الخامس: دارات الدعم غير المبرمجة للمعالج 8080A

السادس: دارات الدعم المبرمجة للمعالج 8080A

السابع: المعالج الصغري 8085A

الثامن: معالجات Intel 16 bits

التاسع: وحدات متممة للمعالجات الصغرية

العاشر: دارات الدعم للمعالج 8086

الحادي عشر: تصميم مايكرو كمبيوتر باستخدام معالجات 8086 Family

الثاني عشر: معالجات صغرية بمحيطيات متكاملة

الثالث عشر: معالجات صغرية 32 Bits

الرابع عشر: معالجات Pentium

يمكن ببساطة ربط المخارج المتوازية للمحول مع العدد المطلوب من أجل منفذ الدخل وقراءة خرج المحول تحت سيطرة برنامج التحكم ، كما يوجد بالإضافة إلى خطوط المعطيات ، خطا إشارة آخران لمحول التقريب المتتالي A/D نحتاجهما للربط مع الحاسوب الصغري. الإشارة الأولى هي " إشارة بدء التحويل" Start Convert ، التي يرسلها الحاسوب الصغري إلى المحول A/D ليخبره بلحظة إجراء التحويل. والإشارة الثانية هي" إشارة الحالة " Status التي يخرجها المحول A/D ليشير بها إلى أن التحويل قد انتهى وبأن الكلمة على المخارج موجودة الآن بشكلها الصحيح.

٣-٣-٤ ربط المحولات A/D مع الحواسب الصغرية

Interfacing Different Types of A/D converter to Microcomputers

وهنا نميز بين ثلاثة أنواع من الربط البيني هي:

أولا- الربط البيني مع محولات A/D ذات مقارنات متوازية:

يبدو أنه في أي تطبيق يستخدم فيه المحول A/D بمقارن على التوازي ، فإن المحول يشكل قيم الخرج الرقمية بصورة أسرع بكثير من إمكانية قراءة الحاسوب الصغري لها . ولذلك تستخدم جملة دارات مستقلة لتجاوز Bypass الحاسوب وكذلك لتحميل مجموعة العينات من المحول مباشرة في سلسلة من المواقع الذكرية، ويمكن للحاسوب بعد ذلك إجراء العملية المقررة على تلك العينات. تسمى عملية التجاوز المصممة بهذه الطريقة " بالولوج المباشر إلى الذاكرة " DMA . أي أن هناك دارة متكاملة تخبر المعالج بأن يضع ممراته في الحالة العائمة، وعندما يفعل المعالج ذلك، فإن متحكم الولوج المباشر إلى الذاكرة يتولى قيادة الممرات، وتسمح بنقل المعطيات مباشرة من A/D إلى المواقع الذكرية المتعاقبة.

ثانياً- الربط البيني مع المحولات A/D ذات الميل:

Interfacing to Slope – Type A/D Converters

كانت معظم المحولات الشائعة الاستخدام من النوع الذي يعمل حسب طريقة الميل المضاعف، مصممة لقيادة وحدات الإظهار ذات القطع السبع مثل مقياس الجهد الرقمي. ولذلك فإن هذه المحولات تخرج المعطيات عادة بصيغة العشري المشفر ثنائيا BCD والمضاعف Multiplexed أو بصيغة القطع السبع 7- segments . تمتلك هذه المحولات عدادات التردد Frequency Counters ومقاييس الجهد الرقمية digital voltmeters وأجهزة لاختبار أخرى.

ثالثاً- الربط البيني مع المحولات A/D ذي تقريب متتال:

Interfacing a Successive- Approximation A/D Converter تمتلك المحولات A/D ذات التقريب المتتالي عادة مخارج لكل بت ، وتكون مخارج الشفرة على هذه الخطوط عادة ثنائية مباشرة Straight Binary .

فعند جهد دخل قدره 2.6 V مثلا ستكون مخارج المقارنات A1 و A2 في المستوى المنطقي المرتفع. إن الميزة الرئيسة للمحول المتوازي (التفرعي) A/D أو " ألوميضي " Flash هي سرعة التحويل، وهي ببساطة زمن تأخير المقارنات. ولا تكون شفرة الخرج الناتجة من المقارنات شفرة ثنائية قياسية، ولكن يمكن تحويلها إلى هذه الصيغة بوساطة دارات منطقية بسيطة. أما السيئة الأساسية للمحول التفرعي A/D فهي عدد المقارنات المطلوبة للحصول على نتيجة بالقدر المعقول من دقة التحويل . ولتشكيل محول ذي N بت فان الأمر يتطلب $(2^N - 1)$ مقارن.

2- المحولات A/D ذات الميل المضاعف Dual – Slope A/D Converters

يستخدم هذا النوع من المحولات غالبا في مقياس الجهد الرقمي Digital Voltmeter لأنه يستطيع إعطاء عدد كبير من البتات التي تمثل دقة التحويل بكلفة منخفضة. تتوفر المحولات A/D ذات الميل المضاعف على شكل دارات متكاملة، ونكتفي بالنسبة لهذا النوع من المحولات بذكر مثال هو الدارة ICL7136 من شركة Intersil .

3- المحولات A/D ذات التقريب المتتالي

Successive- Approximation A/D Converters

نكتفي هنا بعرض دارة كمثال على المحولات ذات التقريب المتتالي هي المحول National DAC 1280 الذي يجري فيه التحويل بزمن قدره 22 µs . يتألف هذا المحول من شريحة واحدة ويعطي إمكانية تحويل بعرض 12 bit.

يوجد كثير من محولات التقريب المتتالي التي تحوي مجمعات Multiplexers على مداخلها. (فمثلا المحول ADC0816 من شركة National يحوي مجمعا له 16 مدخلا، مما يسمح للمحول بتحويل رقمي لأي واحدة من إشارات الدخل الـ 16، وتتحدد القناة المطلوب تحويل إشارتها بوساطة عنوان مؤلف من 4 bit).

الذي يمثل ببساطة الزمن الذي يأخذه المحول لتشكيل شفرة خرج صحيحة من أجل جهد دخل معين .

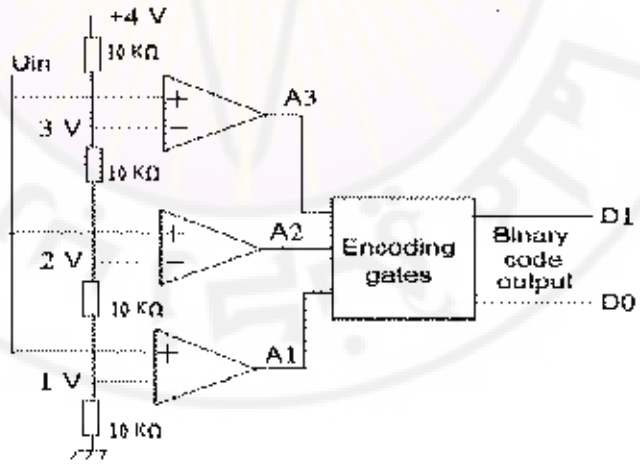
٣-٣-٣-١ طرق تحويل الإشارة التمثيلية إلى قيمة رقمية

توجد طرق عديدة لإجراء التحويل من تشابهي إلى رقمي، تمثل تنوعا واسعا لأزمنة التحويل، نشير هنا إلى ثلاثة منها وهي:

1- المقارن المتوازي Parallel Comparator and A/D converter

لمناقشة طريقة استخدام المقارنات المتوازية، ننطلق من دارة محول بسيط مؤلف من2 bit فقط للتوضيح، وفق ما هو مبين بالشكل 8-3. يستخدم بالشكل أدناه مقسم الجهد لوضع الجهود المرجعية على كل المداخل العاكسة للمقارنات، ويمثل الجهد عند قمة سلسلة المقسّم قيمة كامل المقياس للمحول.

يطبق الجهد المطلوب تحويله على المداخل غير العاكسة لكل المقارنات على التوازي. فإذا كان جهد الدخل على المقارن أكبر من الجهد المرجعي على المدخل العاكس، سيكون خرج المقارن في المستوى المنطقي المرتفع High وستعطي مخارج المقارنات، التمثيل الرقمي لمستوى إشارة الدخل.



الشكل8-3: محول A/D مكون من مقارنات متوازية

مداخل المحول مع الممرات مباشرة. عملية ربط المحول مع الحاسوب الصغري تكون قياسية إذا كانت طول كلمة الحاسوب مماثلة لدقة تحويل المحول. أما إذا احتجنا لتطبيق معين دقة تحويله 12 bit مثلا مع حاسوب له ممر معطيات بعرض 8 bit، فإننا نحتاج لربط محول ذي 12 bit وبهذه الحالة نربط المداخل الثمانية السفلى للمحول إلى أحد المخارج، ويربط المداخل الأربعة العليا إلى منفذ آخر، بحيث ترسل البتات الثمانية الأخفض بعملية كتابة واحدة، بينما ترسل البتات الأربعة العليا بعملية كتابة أخرى. تسبب هذه الحالة بعض المشاكل التي تنتج عن الزمن بين عمليتي الكتابة (أي أن الخرج سوف يذهب إلى قيمة غير مطلوبة في الزمن الفاصل بين عمليتي الكتابة). يتم التخلص من هذه المشكلة باستخدام ماسكات Latches للمعطيات على خطوط النقل. إذ يمكن لهذه الماسكات أن تحمّل بشكل منفصل وتتمكن معا لتمرير الـ 12 bit إلى المحول بنفس الوقت. ولذلك أصبحت معظم المحولات المتوفرة حاليا تحتوي مسجلات مسك متكاملة معها على نفس الشريحة وذلك لجعل الأمر أسهل. تتطلب المحولات D/A جهدا مرجعيا، وبالتالي يجب تأمين هذا الجهد وبقيمة دقيقة مع كل محول تجنبا لأية مشاكل إضافية بعملية التحويل. كما أن للمحولات D/A خرج تياري Current Output وبالتالي حسب نوع التطبيق قد يلزم في حالات كثيرة استخدام مضخم عملياتي لتحويل تيار الخرج إلى الجهد الموافق. ويلاحظ أثناء ربط المحولات مع الحاسوب، أن معظم المحولات من نوع D/A تملك تأريضاً رقمياً Digital Ground وآخر تشابهياً Analog Ground وذلك لتفادي الضجيج الرقمي في الأجزاء التشابهية للدارة، بالتالي فان هذين التأريضين يجب أن يربطا معا فقط عند مصدر التغذية.

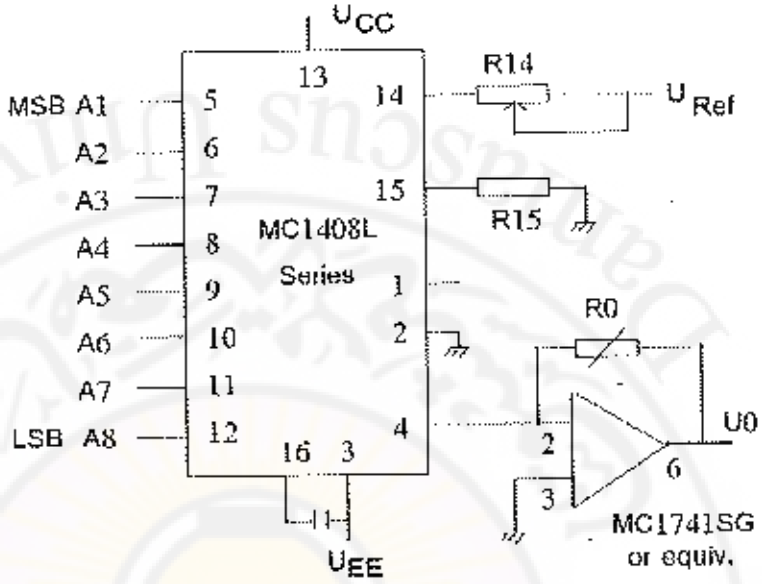
٣-٣-٢ محولات الإشارة التشابهية الرقمية A/D

إن وظيفة المحول من تشابهي إلى رقمي A/D هي تشكيل كلمة رقمية تقابل مقدار جهد أو تياراً تشابهياً معيناً. تتشابه خصائص المحول A/D كثيرا مع تلك الخصائص الموجودة في المحول D/A، حيث إن دقة تحويل المحول تشير إلى عدد البتات في كلمة الخرج الثنائية . يوجد أيضاً خاصية مهمة للمحول A/D وهي " زمن التحويل " Conversion time

إن خاصية الدقة أو الضبط (Accuracy) للمحول D/A هي المقارنة بين الخرج الفعلي والخرج المتوقع . ويتحدد ذلك كنسبة مئوية من جهد خرج المقياس الكامل. إذا كان للمحول خرج ذو مقياس كامل 10V وله دقة مئوية قدرها % 0.2 ± ، فإن الخطأ الأعظمي لأي خرج سيكون بالقيمة 0.002 x 10.00V او 20 mV، ونموذجيا يجب أن لا يكون الخطأ الأعظمي في المحول D/A اكبر من 1/2 ± من قيمة البت الأقل أهمية LSB.
أما الخاصية الأخرى المهمة للمحول D/A فهي " الخطية " (Linearity) والخطية هي قياس مقدار انحراف التابع الخطي المتزايد Ramp عن الخط المستقيم، وذلك عندما ينتقل المحول من حالة وجود المفاتيح في حالة فصلOFF إلى الحالة التي تكون فيها كل المفاتيح موصولة ON. وفي الحالة المثالية يجب أن لا يزيد انحراف الخرج عن الخط المستقيم بأكثر من 1/2 ± من قيمة البت الأقل أهمية، وذلك للحفاظ على الدقة.
إضافة لما سبق يوجد خاصية أخرى للمحول D/A وهي " زمن الوصول إلى حالة الاستقرار " Setting Time ، فعندما نغير الكلمة الثنائية المطبقة على دخل المحول ، سيتغير خرجه إلى قيمة مناسبة جديدة ، وسيتأرجح حول القيمة الحقيقية لفترة من الزمن قبل إن يستقر عليها أخيرا . يسمى الزمن الذي يأخذه الخرج لكي يستقر ضمن القيمة 1/2 LSB ± من القيمة النهاية بزمن الوصول إلى حالة السكون أو الاستقرار. وتعد هذه الخاصية مهمة لان المحول سيعمل عند تردد عالٍ وبالتالي فقد لا يملك وقتا كافيا للاستقرار على قيمة واحدة قبل أن ترد الكلمة الثنائية التالية على مداخله.

٣-٣-٢ ربط المحولات D/A مع الحواسيب الصغرية

توجد تطبيقات عديدة لمحولات D/A كتحويل المعطيات الثنائية المخزنة إلى إشارات صوتية تشابهية وغيرها من التطبيقات والتي لا يسمح المجال هنا للدخول بتفاصيل أي منها، لكن بكل التطبيقات الممكنة يمكن بغض النظر عن نوع المحول ربط مداخل المحول مباشرة إلى منفذ خرج الحاسوب الصغري، فبالنسبة للمحول MC 1408 (بكلمة 8 bit) يمكن ربط المداخل A8 – A1 مباشرة إلى منفذ خرج الحاسوب ويمكن (كجزء من برنامج) تشكيل أي جهد مطلوب على خرج المحول. يتم من أجل بعض محولات D/A ببساطة ربط



الشكل7-3: المحول MC 1408 8 bit D/A مع محول من تيار إلى جهد

الخاصة الثانية للمحول D/A والمطلوب تحديدها هي " جهد خرج كامل المقياس أو التدريج Full-Scale Output Voltage ". ففي المحول المبين بالشكل أعلاه (من أجل U_{Ref} = 2.0V DC و R0 = 5.0 KΩ و R14 = R15 = 1.0 KΩ)، يتم تقديم كل التيارات بوساطة الجهد المرجعي U_{Ref} عبر المقاومة R14. إذ يتم سحب تيار الخرج من المأخذ 4 عبر المقاومة R0 لتشكيل جهد الخرج، حيث إن صيغة جهد الخرج هي من الشكل التالي: $U0 = U_{Ref}/R14(R0)\{A1/2 + A2/4 + \ldots + A8/256\}$

$$U0 = 2/1\ K\Omega(15\ K\Omega)\{1/2 + 2/4 + \ldots + 8/256\} =$$

$$10V\{255/256\} = 9.96V$$

يمثل الحد A1 في المعادلة شرط المفتاح من أجل ذلك البت. إذا كان أحد المفاتيح مغلقا فهذا يسمح للتيار بالتدفق، ويتم وضع 1 في ذلك البت. وإذا كان المفتاح مفتوحا سوف يتم وضع 0 في ذلك البت. أما إذا كانت جميع المفاتيح مغلقة فإن الخرج سيكون 9.96 V. تستخدم عادة شفرات ثنائية مختلفة عديدة كمداخل للمحولات D/A مثل العشري المشفر ثنائيا BCD أو " الثنائي المباشر" (Straight Binary) .

يتم سحب مجموع التيارين إلى الخرج عبر مقاومة التغذية العكسية، فعلى المضخم أن يضع خرجه بقيمة جهد تساوي 0.15 mA x 10 KΩ أو 1.5 V. النقطة المهمة هنا أن المقاومات الموزونة ثنائياً Binary–Weighted–Resistors تشكل تيارات موزونة ثنائياً أيضاً، وهذه التيارات يتم تجميعها من قبل المضخم العملياتي إلى جهد الخرج المناسب، حيث تشكل الكلمة الثنائية المطبقة على المفاتيح جهد خرج متناسب معها، وتقنياً يمكن اعتبار الخرج رقمياً لأنه يمكن أن يأخذ قيماً ثابتة ومحددة تماما كما في الإظهار الموجود في مقياس الجهد الرقمي .

يمثل المفتاح D3 في الشكل أعلاه البت الأكثر أهمية Most – Significant bit وذلك لان إغلاقه يشكل التيار الأكبر. لاحظ في هذا الشكل أيضاً أن الجهد المرجعي سالب، وبالتالي سيذهب الخرج بالاتجاه الموجب عندما تكون المفاتيح مغلقة Closed نلاحظ هنا أن قلب المحول D/A هو مجموعة من مصادر التيار الموزونة ثنائياً، والتي يمكن فتحها وإغلاقها تبعاً للكلمة الثنائية المطبقة على مداخلها. وبما أن هذه المصادر من التيار تكون موجودة عادة داخل دارة متكاملة ، فلذلك ليس هناك ضرورة لمناقشة الطرق المختلفة التي يمكن بها توليد تيارات موزونة ثنائياً وهكذا يمكن استخدام دارة مضخم عملياتي بسيط كما هو مبين في الشكل أعلاه لتحويل مجموع التيارات إلى جهد متناسب مع هذه التيارات.

ميزات وخصائص المحول D/A

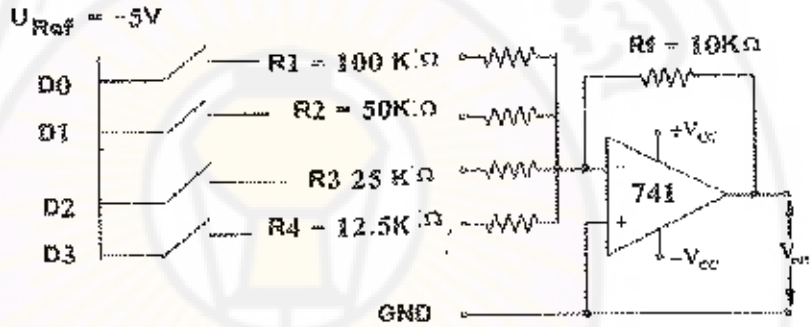
لمناقشة مميزات وخصائص المحولات D/A ننطلق من دارة محول محدد مثل دارة المحول MC 1408 8 bit D/A المبينة مع توصيلها القياسي في الشكل 7–3. إن الميزة الأولى للمحول D/A والتي تؤخذ بالاعتبار هي "دقة التحويل" Resolution التي تتحدد بعدد البتات في كلمة الدخل الثنائية . وهكذا فإن محولا بثمانية مداخل ثنائية يمتلك 2^8 أو 256 مستوى ممكن في الخرج ، وبذلك تكون دقة تحويله جزء واحد من 256 . وقد يعبر عن دقة التحويل في بعض الأحيان بالنسبة المئوية.

الأول: لتحويل الإشارات التمثيلية إلى قيم رقمية Analog to Digital Converter اختصارا A/D Converter

الثاني: لتحويل القيم الرقمية إلى إشارة تمثيلية Digital to Analog Converter اختصارا D/A Converter

٣-٣-١ محولات الإشارة الرقمية التشابهية D/A

إن الهدف من المحول الرقمي إلى تشابهي هو تحويل الكلمة الثنائية إلى تيار أو جهد موافق، ولكي نرى كيفية عمل ذلك، ننظر إلى دارة المضخم المبينة في الشكل 6-3.



شكل6-3: محول بسيط من رقمي إلى تشابهي بعرض 4 bit

حيث تؤدي هذه الدارة وظائف جامع Adder ، بما أن المدخل غير العاكس للمضخم مؤرض، فسوف يعمل المضخم باستمرار للمحافظة على مدخله العاكس عند الجهد صفر أيضاً. وهذا ما يشار إليه بالأرضي الظاهري Virtual Ground أو نقطة الجمع Summing Point. إذا أغلقنا أحد المفاتيح، سيمر تيار من الجهد المرجعي V_{REF} = −5V من خلال تلك المقاومة إلى نقطة التجميع ، وفي هذه الحالة يسحب المضخم العملياتي التيار عبر مقاومة التغذية العكسية لتشكيل جهد خرج متناسب مع التيار. إذا أغلقنا المفتاح D0 على سبيل المثال، فإن تياراً قدره 0.05 mA سيتدفق إلى نقطة التجميع، ولكي نسحب هذا التيار عبر مقاومة التغذية العكسية، فإن المضخم العملياتي سوف يضع جهداً قدره 0.05 mA x10 k Ω على خرجه أي [illegible] وإذا أغلقنا المفتاح D1 ، أيضاً فإنه سيرسل تياراً آخر قدره 0.1 mA إلى نقطة التجميع، لكي

معدل دوران هذا الدولاب يكون متناسبا مع معدل التدفق للسائل أو الغاز. ويمكن وضع مشفر ضوئي Optical Encoder على محور الدولاب بغرض تشكيل معلومات رقمية تدل على سرعة دوران الدولاب.

ثاني الطرق الغالبة الاستخدام لقياس التدفق هي استخدام " مبدل ضغط تفاضلي" Differential Pressure Transducer التي يتم وفقها وضع شبكة أسلاك Mesh Wire أو شاشة Screen في طريق الأنبوب من أجل خلق مقاومة معينة. حيث يشكل الجريان عبر هذه المقاومة اختلافا في الضغط بين طرفي المقاومة. ويعطي مبدل الضغط خرجا متناسبا مع فرق الضغط بين طرفي المقاومة. وبنفس الطريقة التي يتناسب فيها الجهد عبر مقاومة كهربائية مع تدفق التيار خلال المقاومة، فإن خرج مبدل الضغط هنا يتناسب أيضاً مع تدفق السائل أو الغاز خلال الأنبوب.

4- محولات (أو مبدلات) القوة والضغط

لكي يتم تحويل القوة أو الضغط (قوة / سطح) إلى إشارة كهربائية موافقة، فان أغلب الطرق الشائعة تستخدم مقاييس الاستطالة Strain gages ومحولات تفاضلية متغيرة خطياً أو LVDT (Linear variable differential transformers) حيث يؤدي كل من هاتين الطريقتين إلى تحريك شيء معين، وهذا هو السبب في تسمية هذه العناصر "مبدلات" وليس حساسات.

٣-٣ محولات الإشارة

بعد أن أشرنا إلى استخدام الحساسات للحصول على إشارات كهربائية متناسبة مع عوامل متعددة مثل الضغط والحرارة وغيرها وكيفية استخدام المضخمات العملياتية لتضخيم وترشيح هذه الإشارات، بقي أن نتعرف على كيفية تحويل هذه الإشارات من شكلها التمثيلي (أو التشابهي) إلى قيم رقمية بصيغة يستطيع فيها الحاسوب الصغري من التعامل معها. وكيفية تحويل القيم الرقمية الناتجة من الحاسوب إلى إشارة تمثيلية. يوجد لهذه الغاية نوعين من المحولات Converters، هما:

*- مصادر التيار الحساسة للحرارة Temperature Sensitive Current Sources وكمثال على هذا النوع نذكر العنصر AD590 حيث يولد هذا الجهاز تياراً قيمته واحد ميكروأمبير لكل كلفن (1 μA/Kelvin).

المزدوجات الحرارية Thermocouplers

يمكن استخدام المزدوجات الحرارية لقياس درجات حرارة منخفضة جداً، وأيضاً درجات حرارة عالية جداً. معلوم أنه كلما تم تحقيق تماس بين معدنين مختلفين يتشكل جهد صغير بينهما يعتمد على نوع المعدنين المستخدمين وعلى الحرارة أيضاً. واعتمادا على المعادن فإن الجهد الناتج يتزايد بين 7μV و 75μV من أجل كل زيادة في الحرارة مقدارها 1 C^o. تملك وصلة المزدوجة الحرارية المكونة من الحديد وسبيكة نحاس ونيكل مجالا حراريا جيدا حوالي 184 C^o - إلى 760 C^o + وتملك الوصلة المصنعة من البلاتين وخليطه من البلاتين مع 13% روديوم مجالا حراريا مفيدا من 0 C^o إلى حوالي 1600 C^o. يمكن صنع الحساسات الحرارية بحيث تكون بحجم صغير ومتينة، إلا أن فيها ثلاث مشاكل رئيسية يجب التغلب عليها وهي:

*- أولى هذه المشاكل أن الخرج الناتج منها صغير جدا ويجب أن يضخم بمعالجة كبيرة لكي يصل إلى المجال الذي يستطيع فيه مثلا قيادة محول من النوع A/D.

*- المشكلة الثانية أنه لكي نجري قياسات دقيقة يجب أن تتضمن الدارة وصلة ثانية من نفس المعدن كمرجع للعمل، ويشار إلى إضافة هذه الوصلة الثانية بما يسمى " تعويض الوصلة الباردة".

*- المشكلة الثالثة للمزدوجات الحرارية هي أن جهود خرجها لا تتغير خطيا مع الحرارة ويمكن تصحيح ذلك بجملة دارات تشابهية تستطيع تغير ربح المضخم تبعا لقيمة الإشارة.

3- حساسات التدفق Flow Sensors

إذا أردنا التحكم بمعدل التدفق أو الجريان لمادة سائلة أو غازية، فنحن بحاجة إلى وسيلة لقياس هذا التدفق. واعتمادا على المادة، ومعدل الجريان والحرارة فإن: أولى الطرق المستخدمة هي وضع دولاب ذو أرياش Paddle wheel في طريق التدفق، حيث إن

1- الحساسات الضوئية Light Sensors

أحد أبسط أنواع الحساسات الضوئية هو المقاومة Light-dependent Resistor التي تعتمد على الضوء. تتغير قيمة بعض المقاومات (مثل المقاومة CL905 التي تعتمد مقاومتها على كمية الضوء الساقط عليها، حيث إنها مجهزة بنافذة زجاجية تسمح للضوء بالسقوط على نموذج متعرج ضمنها) من حوالي 15 MΩ في الظلمة إلى حوالي 15 KΩ في الضوء الساطع. حيث إن مقاومات كهذه لا تملك زمن استجابة سريع جداً، كما إنها ليست مستقرة مع الحرارة ولكنها حساسة ومتينة ورخيصة. العنصر الآخر الذي يستخدم لتحسس كمية الضوء الموجودة هو Photodiode أي الديود الضوئي. فإذا سقط الضوء على ديود من السليكون (ذو تصميم خاص)، فإن تيار التسريب العكسي للديود سيزداد خطيا عندما تزداد كمية الضوء الساقطة عليه. أخيرا نشير إلى أن هناك جهازاً آخر حساس للضوء، وله تطبيقات مفيدة هو الخلية الشمسية Solar-Cell التي تمثل ببساطة وصلات PN كبيرة من السيلكون. يسبب الضوء المشع على الخلية الشمسية مرور تيار التسريب العكسي تماما كما في حالة الديود الضوئي. وبسبب المساحة الكبيرة للخلية الشمسية فإن التيار يكون بالميلي أمبير وليس بالميكرو.

2- الحساسات الحرارية Temperature Sensors

هناك أنواع عديدة من الحساسات الحرارية، نشير هنا إلى نوعين منها فقط، هما الحساسات الحرارية نصف الناقلة والمزدوجات الحرارية، حيث إن:

الحساسات الحرارية نصف الناقلة

تعدّ حساسات أنصاف النواقل (Semiconductor Temperature Sensors) غير مكلفة ويمكن استخدامها لقياس درجات حرارة في المجال من 55 – وحتى $^{\circ}C$ 100 + يوجد نوعان رئيسيان من الحساسات الحرارية نصف الناقلة هما:

*- مصادر الجهد الحساسة للحرارة Temperature Sensitive Voltage Sources وكمثال على هذا النوع نذكر العنصر National LM35 حيث يتزايد جهد الخرج من هذه الدارة بمقدار 10 mV لكل درجة مئوية تزداد فيها الحرارة.

10 مرات في التردد. ومن أجل مرشح من المرتبة الثانية يتناقص الربح بالعامل 100 لكل زيادة مقدارها 10 مرات في التردد . سنكتفي هنا بالإشارة إلى نوعين من المرشحات هما:

أولا: عمل المضخم العملياتي كمرشح تمرير منخفض من المرتبة الثانية:

يبين الشكل3b-3 دارة عامة لمرشح تمرير منخفض من المرتبة الثانية (حيث أن C1 = 2C2 و R1 = R2 و R3 = 1 KΩ). إن الطريقة التي نستطيع فيها تمييز المرشح بأنه مرشح تمرير منخفض هي النظر إلى مسار المركبة (DC –Path) المستمرة من الدخل إلى الخرج . فإذا كان المسار المستمر موجودا، نعرف أن المضخم يستطيع تضخيم القيم المستمرة DC والترددات المنخفضة وبالتالي يكون المرشح عبارة عن مرشح منخفض التمرير .

ثانياً: عمل المضخم العملياتي كمرشح تمرير مرتفع من المرتبة الثانية:

يبين الشكل 3c-3 دارة مرشح التمرير المرتفع من المرتبة الثانية (حيث إن C1 = C2 و R2 = 2R1 و R3 = 1 KΩ). يمكن ملاحظة أن المركبة المستمرة لإشارة الدخل في هذه الدارة لا تستطيع الوصول إلى الدخل غير العاكس ، وذلك بسبب وجود المكثفين C1 و C2 على التسلسل مع ذلك الدخل . وبناء على ذلك سوف لن تضخم المركبة المستمرة والإشارات ذات التردد المنخفض. يمكن وضع مرشح التمرير المنخفض والمرتفع على التسلسل لتشكيل مرشح تمرير الحزمة band pass-filter الذي يسمح بتمرير الترددات ضمن مجال محدد.

٣-٢ الحساسات والمبدلات

يوجد أنواع مختلفة وكثيرة من الحساسات Sensors and Transducers. فمنها ما يستخدم لقياس درجة PH وآخر لقياس تركيز الغازات المتنوعة، وبعضها لقياس ثخانة المواد المختلفة. ولا يمكننا هنا وصف طرق عمل وتطبيقات كل الأنواع المختلفة من الحساسات والمبدلات، وبالتالي سنكتفي بالتعريف ببعض هذه العناصر لإظهار كيفية استخدامها للحصول على المعطيات وإدخالها إلى الحاسوب الصغري.

٧- المضخم العملياتي كدارة جامع Adder (Mixer)

يبين الشكل 3-3a دارة مضخم عاكس تقوم بجمع Add أو مزج Mix إشارتي دخل أو أكثر . ولفهم ذلك نعلم أن المضخم العملياتي في الدارة العاكسة يحافظ على جهد الدخل العاكس عند المستوى الأرضي الظاهري، وهنا في الدارة يمرر جهد الدخل U1 تيارا عبر R1 إلى هذه النقطة . كما يمرر جهد الدخل U2 تيارا عبر R2 إلى هذه النقطة أيضاً . حيث يتم جمع التيارين عند الأرضي الظاهري، والذي يدعى عموما بنقطة " الجمع (Summing Point)" لهذه الدارة . يسحب مجموع التيارين عبر المقاومة Rf بوساطة المضخم ليتم تثبيت الدخل العاكس عند جهد صفري . وعند ذلك فإن جهد الخرج $U_{out} = (U1/R1 + U2/R2) \times Rf$ (والقيمة سالبة لان $\varnothing = 180$).

٨- عمل المضخم العملياتي كمرشح فعال OP-AMP Active Filter

نحتاج في تطبيقات عديدة لترشيح التردد المنخفض غير المرغوب فيه أو لحذف الضجيج ذي التردد العالي من الإشارات المقروءة من الحساسات . ويمكن عمل ذلك بمرشحات RC بسيطة ، إلا أن " المرشحات الفعالة " Active Filters باستخدام المضخمات العملياتية تعطي تحكما وسيطرة أفضل بكثير على خصائص المرشح ، وهناك تشكيلات مختلفة عديدة من هذه المرشحات باستخدام المضخم العملياتي. إن النقاط الأساسية التي نريد تذكرها هنا هي معاني المصطلحات التالية:

- مرشح التمرير المخفض (Low-pass filter)
- ومرشح التمرير المرتفع (High- pass filter)
- ومرشح تمرير الحزمة (Band- pass filter)

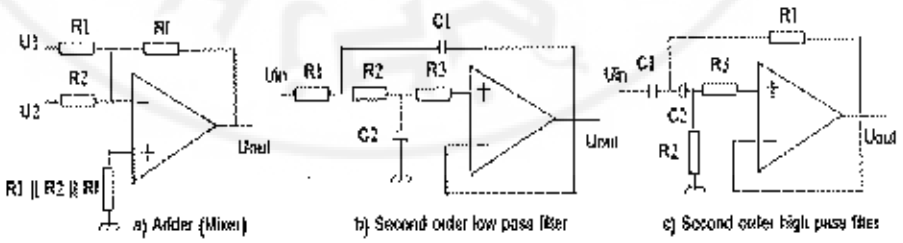
يضخم مرشح التمرير المنخفض أو يمرر ترددات منخفضة، ولكن خرجه يبدأ بالتناقص عند تردد يتحدد بقيم الدارة. إذ يسمى التردد الذي يهبط الخرج عنده إلى القيمة 0.707 من قيمة التردد المنخفض بالتردد الحدي Critical frequency أو Break point نقطة القطع، حيث إن الربح يتناقص ما بعد التردد الحدي fc بشكل سريع. ففي مرشح من المرتبة الأولى كالمرشحات RC يتناقص الربح بالعامل 10 من أجل كل زيادة قدرها

مساويا الصفر. وإذا طبقنا جهدا جديدا على المدخل ، فإن الجهد على المكثف لا يستطيع التغيير لحظيا ، وبذلك يزاح الدخل العاكس عن الصفر. يسبب ذلك قيادة خرج المضخم العملياتي باتجاه شحن المكثف وإرجاع المدخل العاكس إلى الصفر. حيث يتحدد الزمن المطلوب من أجل عودة الخرج إلى الصفر بالثابت الزمني للمقاومة R1 والمكثف C .

٦- عمل المضخم العملياتي كمكامل OA Integrator Circuit

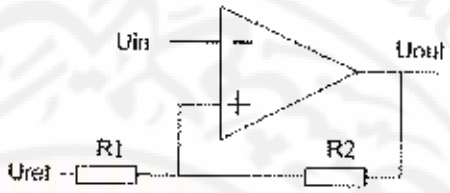
يبين الشكل 3-3c دارة مضخم عملياتي يمكن استخدامها لتوليد جهود متزايدة خطيا مع الزمن (Ramp-Generator) . يطبق جهد مستمر DC متقطع على دخل هذه الدارة، وهذا الجهد يسبب مرور تيار ثابت قيمته Uin/R1 بنقطة الأرضي الظاهرة . يمر هذا التيار في إحدى صفائح المكثف، ولكي نثبّت جهد المدخل العاكس على قيمة تساوي جهد الأرضي، فإنه ينبغي على خرج المضخم العملياتي أن يسحب نفس التيار من الصفيحة الأخرى للمكثف، وبذلك يكون قد تم شحن المكثف بتيار ثابت Uin/ R1. وهكذا نلاحظ هنا أنه بسبب توصيل المضخم العاكس فإن الخرج سيكون تابعا خطيا سالبا من أجل جهد دخل موجب . ونلاحظ أيضاً بأنه يجب إجراء بعض الاحتياطات وذلك لمنع الخرج من التزايد إلى "حد الإشباع Saturation". تسمى هذه الدارة بدارة المكامل Integrator، لأنها تشكل جهد خرج متناسب مع التكامل أو " مجموع " Sum التيار المتشكل بوساطة جهد الدخل على كامل الدور الزمني .

إضافة للوظائف المشار إليها أعلاه لمضخم العمليات، يظهر الشكل 3-5 ثلاث توصيلات أخرى لثلاث وظائف لمضخم العمليات هي:



الشكل3-3: توصيلات مضخم العمليات كجامع ومرشح

الصغري (معظم المقارنات المتوفرة حاليا مثل الدارة LM319 لها مخارج متوافقة مع TTL بحيث يمكن ربطها مباشرة إلى منافذ الحاسوب الصغري). ومن الدارات الخاصة للمقارن وشائعة الاستخدام الدارة المبينة بالشكل 4-3، والتي تسمى مقارن بتخلف (Comparator with Hysteresis)



الشكل4-3: دارة مقارن بتخلف

وفيها تكون الإشارة المرجعية مطبقة على المدخل غير العاكس وجهد الدخل مطبق على المدخل العاكس، ويلاحظ وجود مقاومات تغذية عكسية موجبة من الخرج إلى المدخل غير العاكس (R1 = 10 Ω, R2 = 10 K)، وهذه التغذية العكسية تعطي المقارن ما يسمى بخاصية " التخلف Hysteresis أو التأخير"، ومفادها أن جهد الخرج يتغير عند جهد دخل مختلف، عندما يذهب الدخل في الاتجاه الموجب. كما تحصل خاصية التأخير أيضاً عندما يذهب جهد الدخل في الاتجاه السالب. ويمكن تلخيص عمل المقارن بالتالي:

If $U_{in} < U_{Reference} \rightarrow$ Output = U – 1V

If $U_{in} > U_{Reference} \rightarrow$ Output = – U + 1V

حيث إن $U_{Hysteresis} = Uout\ (R1/(R1 + R2)$

ومن الحالات التي نشاهد فيها فعلا تأخيريا، هي حالة وجود إشارة متغيرة ببطء مع ضجيج فيها. حيث يمنع فعل التأخير هنا الضجيج من التسبب في اهتزاز خرج المقارن عندما تكون إشارة الدخل قريبة جدا من الجهد المرجعي.

٥- عمل المضخم العملياتي كمفاضل OP AMP Differentiator Circuit

يبين الشكل 3-3b دارة مضخم عملياتي تعطي إشارة خرج متناسبة مع معدل التغير في إشارة الدخل . إذا كان جهد الدخل لهذه الدارة صفرا أو ثابتا (مستمر)، فإن الخرج سيكون

٤- المضخم العملياتي كمقارن OP AMP as Comparator

بفرض إننا طبقنا على توصيلات مصدر التغذية لمضخم عملياتي جهود تغذية بمقدار 15V+ و 15V- ووصلنا المدخل العاكس إلى الأرضي وطبقنا إشارة مستمرة قدرها 0.01V DC على المدخل غير العاكس كما هو مبين في الشكل 3-3a . يلاحظ أن OP AMP سيحاول تكبير هذه الإشارة 100000 مرة وإعطاء النتيجة على خرجه. إن إشارة قدرها 0.01V مضروبة بربح قدره 100000 سيعطي جهد خرج قدره 100V إلا إن النتيجة غير ذلك، إذ يمكن لخرج OP AMP أن يستمر في الصعود الموجب حتى يصل إلى جهد اقل من جهد التغذية الموجب (ربما 13 V) وهذا أقصى شيء يستطيعه المضخم. لنفرض الآن أننا طبقنا إشارة قدرها (0.01V –) على المدخل غير العاكس، نلاحظ أن الخرج سيحاول الآن الذهاب إلى القيمة (100V –) بأسرع ما يمكن، أي الخرج يستطيع في هذه الحالة الذهاب إلى القيمة (13 V –) وبذلك يقف هنا.

يقارن المضخم العملياتي في هذه الدارة بفاعلية جهد الدخل مع الجهد على المدخل العاكس، حيث يعطي خرجا مرتفعا أو منخفضاً اعتمادا على نتيجة المقارنة. فإذا كان جهد الدخل على المدخل العاكس فوق الجهد المرجعي ببضعة ميكرو فولت، فإن الخرج سوف يكون مرتفعا (13 V +). أما إذا كان جهد الدخل أكثر سلبية ببضعة ميكرو فولت من الجهد المرجعي، فإن الخرج سوف يكون منخفضا (13 V –).

يدعى المضخم العملياتي المستخدم بهذه الطريقة " بالمقارن Comparator". إن الجهد المرجعي Reference Voltage المطبق على الدخل العاكس لا يكون عادة الارضي 0V ، إذ يمكن مقارنة جهد الدخل مع أي جهد ضمن مجال الدخل المحدد لكل مضخم عملياتي بشكل مستقل. ويمكننا بشكل عام تلخيص عمل المقارن بالشكل:

If $U_{in} > U_{Referenne} \rightarrow$ Output = +U – 1V

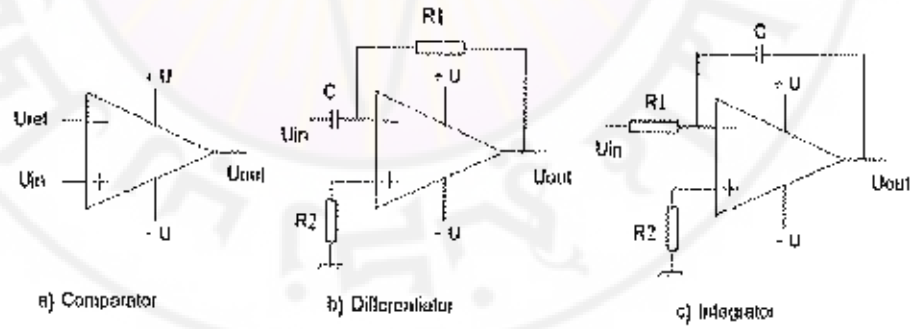
If $U_{In} < U_{Reference} \rightarrow$ Output = –U + 1V

بالطبع للمقارنات تطبيقات عديدة، إذ يمكن على سبيل المثال ربط مقارن إلى حساس حراري في تطبيق ما، وعندما يصبح الجهد الناتج من الحساس أعلى من الجهد المرجعي للمقارن، فإن خرج المقارن سيغير الحالة ويرسل إشارة مقاطعة Interrupt إلى الحاسوب

عليها عند مستوى الأرضي . حيث يحدد ربح الجهد بوساطة نسبة المقاومتين . إن ربح الحلقة المغلقة Avcl لهذه الدارة عند الترددات المنخفضة يساوي نسبة المقاومتين .Avcl = – Rf/R1

يمكن اشتقاق هذه العلاقة إذا فكرنا بالمقاومتين كمقسم جهد موصول إليه الجهد Uin عند إحدى النهايات، والجهود الصفرية في الوسط والجهد Uout عند النهاية الأخرى. تشير إشارة الناقص في علاقة الربح ببساطة إلى أن الخرج يعاكس الدخل بالصفحة وتكون ممانعة الدخل Zin لهذه الدارة مساوية تقريبا للمقاومة R1، وذلك لان المضخم العملياتي يؤمن على أحد أطراف هذه المقاومة جهدا يساوي الصفر تقريبا. وهنا نذكر بخاصية إضافية للمضخم هي " ناتج ضرب الربح بعرض الحزمة Gain–Bandwidth Product" حيث كما نعلم بأن المضخم يمتلك ربح حلقة مفتوحة (Open–loop DC gain) بمقدار 100000 أو أكثر. وعند الترددات العالية يتناقص الربح حتى يصل عند بعض الترددات إلى قيمة الواحد. يرمز عادة للتردد الذي يكون عنده الربح مساويا الواحد "بجداء عرض حزمة الربح حيث إن القيمة المعتادة لهذا التردد هي 1 MHz.

وبعد التعرف بشكل مبسط على ثلاثة من الدارات الأساسية لمضخم العمليات، نكمل تعرف بعض داراته الأخرى الواسعة الاستخدام وفق ما هو مبين بالشكل 3–3.



الشكل3–3: توصيلات مضخم العمليات كمقارن ومفاضل ومكامل

وعندما يرتفع الخرج بالاتجاه الموجب ، فإن جزءاً من جهد الخرج هذا يعطى كتغذية عكسية إلى المدخل العاكس من خلال مقسم الجهد، وستؤدي هذه التغذية العكسية على المدخل العاكس إلى تناقص الفرق في الجهد بين المدخلين . وفي هذه الحالة تصل الدارة بسرعة إلى نقطة التوازن ، حيث يكون الجهد على المدخل العاكس $U_{\mp}$ قريباً جداً من الجهد على المدخل غير العاكس Uin. ويمكن أن يكون فرق جهد التوازن هذا حوالي 10 µV من أجل خرج 1V DC. وإذا افترضنا أن الجهدين على المدخلين متساويان، فإن حساب جهد الخرج من أجل جهد دخل مُعطى يكون Uout = Uin . A_{vcl}، حيث إن ربح الحلقة المغلقة Avcl (closed-loop-gain) لهذه الدارة يساوي Avcl = (R1+R2)/R1 .

تسمى التغذية العكسية التي تجعل المضخم العملياتي يقاوم التغير على خرجه "بالتغذية العكسية السالبة" (Negative-Feedback) وبسبب هذه التغذية العكسية السالبة ، فإن المضخم العملياتي سوف يعمل دوما للحفاظ على خرجه مستقراً بحيث يكون الجهد على مدخليه متساوياً تقريباً . ومن المحتمل أن تكون هذه الخاصية أهم نقطة نحتاجها لمعرفة تحليل دارة المضخم العملياتي ذي التغذية العكسية السالبة. تستخدم الدارة غير العاكسة غالباً كمرحلة عزل بينية (Buffer) وذلك لأنها تملك ممانعة دخل Zin عالية جداً.

٣- دارة المضخم العملياتي كمضخم عاكس Inverting Amplifier

يبين الشكل 3-2c دارة مضخم متعدد الجوانب، وتستخدم فيه التغذية العكسية السالبة. يلاحظ في هذه الدارة بأن المدخل غير العاكس موصول بالأرضي عن طريق مقاومة، بينما تطبق الإشارة المطلوب تكبيرها على المدخل العاكس عبر مقاومة أيضاً. ولهذا ستكون إشارة الخرج مقلوبة بالصفحة بمقدار 180 درجة عن إشارة الدخل (180= ∅). تؤمّن التغذية العكسية السالبة عن طريق المقاومة Rf، التي تعمل على أن يكون كلا المدخلين عند نفس الجهد تقريبا .

وبما أن المدخل غير العاكس مربوط إلى الأرضي، فان المضخم العملياتي سوف يسحب (Sink) أو يعطي تيارا لجعل الجهد على المدخل العاكس صفريا . ويشار في هذه الدارة إلى نقطة الدخل العاكس على أنها الأرضي الظاهري، وذلك لان المضخم العملياتي يحافظ

يدعى المدخل المشار إليه بالرمز – بالمدخل العاكس أو القالب Inverting Input. ويدعى المدخل المشار إليه بالرمز + بالمدخل غير العاكس Noninverting Input. بالطبع ليس للإشارة + أو – على هذه المداخل أي علاقة بجهود مصدر التغذية، وإنما تشير إلى علاقة الطور Phase بين الإشارة المطبقة على هذا المدخل والنتيجة التي تشكلها تلك الإشارة على الخرج.

فإذا تم مثلا جعل المدخل غير العاكس أكثر ايجابية من المدخل العاكس، فإن الخرج سوف يتحرك بالاتجاه الموجب وسيكون بنفس الطور مع إشارة الدخل المطبقة. بعد هذا التعريف المختصر بالمضخم العملياتي نذكّر فيما يلي ببعض الاستخدامات (الوظائف) المهمة له والشكل العام للتوصيل المقابل لكل منها

٢ – دارة المضخم العملياتي كمضخم غير عاكس Noninverting Amplifier

عند العمل في نموذج الحلقة المفتوحة Open–Loop أي بعدم وجود تغذية عكسية مطبقة على الدخل العاكس، فإن المضخم العملياتي يعطي ربحاً عالياً جداً ، ولكنه غير قابل للتحكم، ويكون هذا الوضع مقبولا عند استخدام مضخم العمليات كمقارن. ولتشكيل مضخم بربح ثابت يمكن حسابه، يلزم وصل تغذية عكسية من الخرج إلى الدخل. يبيّن الشكل 3–2k طريقة التغذية العكسية السالبة التي تضاف إلى المضخم العملياتي بهذه الحالة لتشكيل المضخم المطلوب. وبالتالي يمكننا ملاحظة:

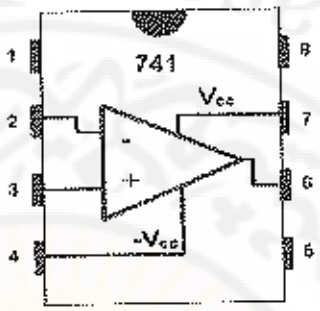
*– أن إشارة الدخل في هذه الدارة مطبقة على الدخل غير العاكس، بحيث يكون الخرج بنفس الطور مع الدخل (in phase ~ ∅).

*– أن جزءاً من إشارة الخرج يعطى كتغذية عكسية إلى الدخل العاكس. ولبيان عمل ذلك: نفترض في البداية بأن جهد الدخل Uin = 0V ، وجهد الخرج Uout = 0V ، وأن الجهد على الدخل العاكس يساوي الصفر 0.

وسنفترض الآن بأننا طبقنا إشارة مستمرة قيمتها 0.01 V DC + على المدخل غير العاكس، بما أن الفرق بين المدخلين يضخّم بمقدار 100000 مرة، فإن الخرج سوف يرتفع باتجاه القيمة 100V بأسرع ما يمكن .

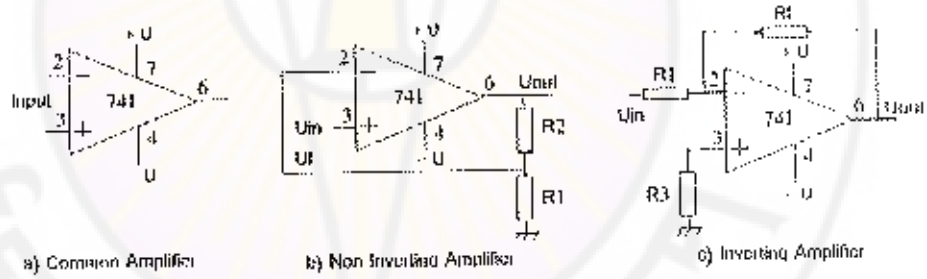
٣-١ الخصائص العامة للمضخم العملياتي ودارته الأساسية

انطلاقا من الشكل العام لمضخم العمليات الذي يصنع عادة على شكل دارة متكاملة، كما هو مبين بالشكل 1-3 بالنسبة لتوزع مآخذ الدارة 741.



الشكل1-3: توزع مآخذ الدارة المتكاملة 741 (مضخم عمليات)

يمكننا رسم دارات متعددة لمضخم العمليات، بحيث إن كل شكل يقابل وظيفة محددة.
يبين الشكل2-3 مجموعة من توصيلات مضخم العمليات.



الشكل2-3: توصيلات مختلفة لمضخم العمليات

وفيما يلي نعرض باختصار مفهوم كل من هذه التوصيلات على حدة وفق التالي:

١ - مضخم العمليات العادي Common OP AMP

يمثل الشكل 3-2a رمزاً لمضخم عملياتي OP AMP عام وفيه نجد المآخذ U+ و U- تمثل توصيلات مصدر التغذية وهي عادة 15V+ و 15V- (أو 12V+ و 12V-).
ويمتلك OP AMP أيضاً مدخلين للإشارة، وهو يقوم بتكبير الفرق في الجهد بين هذين المدخلين بقيمة 100000 مرة أو أكثر.

الفصل الثالث: محولات الإشارات والربط البيني مع الحاسوب

للتحكم بالأجهزة الطبية أو بالمحركات الآلية وغيرها من التجهيزات بوساطة الحاسوب، نحتاج لتحديد قيم جملة من المتغيرات كالضغط والحرارة والتدفق وغيرها من المتحولات. توجد عادة عدة خطوات للحصول على الإشارات التي تمثل قيم هذه المتغيرات، وبما أن معظم هذه الإشارات، مثل إشارات القلب أو الدماغ أو العضلات وغيرها ذات طابع تمثيلي (أو تشابهي Analog) يلزم قبل معالجتها باستخدام الحاسوب تحويلها إلى صيغة رقمية يفهمها المعالج الذي بني عليه الحاسوب. وبالتالي تتطلب الخطوة الأولى من عملية ربط مثل هذه التجهيزات الخارجية بالحاسوب، مجساً (أو حساسا Sensor) يحول مطال إشارة ما أو قيمة متحول كالضغط الفيزيائي مثلا أو التدفق أو الحرارة وغيرها من المتحولات الأخرى إلى جهد أو تيار يتناسب مع المتحول.

تكون الإشارات الناتجة من معظم الحساسات وبخاصة الإشارات الحيوية صغيرة جداً، بالتالي قد يلزم في الخطوة الثانية تكبيرها وترشيحها ومن ثم تحويلها إلى إشارات رقمية. يتم تضخيم الإشارات وترشيحها عادة باستخدام بعض دارات مضخم العمليات Operational Amplifiers (اختصارا OP AMP).

يتم تحويل الإشارات التمثيلية إلى صيغة رقمية بوساطة محولات إشارة A/D (اختصارا لـ Analog to Digital Converter). وقد يلزم في كثير من التطبيقات تحويل القيم الرقمية بعد معالجتها بالحاسوب إلى إشارة تمثيلية وهذا يتم عادة باستخدام محولات D/A (اختصارا لـ Digital to Analog Converter).

يخصص هذا الفصل للتذكير وبشكل مختصر ببعض الحساسات وآلية عملها وببعض دارات المضخمات الشائعة الاستخدام، وصولا إلى طرق تحويل الإشارة من شكل إلى آخر (من رقمي إلى تمثيلي وبالعكس) وتعرف بعض محولات الإشارة وطريقة ربطها بالحاسوب Converters and Analog Interfacing with Computer.

8- تعتمد سرعة المعالج الصغري على

١- a bus width ٢- Address bus width

٣- Clock ٤- Instruction Set

9- ما عدد خطوط العنونة التي تحتاجها ذاكرة بسعة (16K x 8 Bits) 16 KB؟

١- 14 ٢- 15 ٣- 12 ٤- 13

10- لتشكيل ذاكرة 2k X 8bit من ذواكر بالتنظيم 1k X 4bit يلزم :

١- 16 Chips ٢- 8 Chips ٣- 4 Chips ٤- 2 Chips

11- وزن خرج 3x8 decoder بالمداخل A_{13} A_{14} A_{15} أثناء عنونة الذاكرة هو :

١- 4 KB ٢- 8 KB ٣- 32 KB ٤- 16 KB

12- أي من الذواكر التالية يعتبر من النوع غير المتطاير Non Volatile

١- Flash Memory ٢- SRAM

٣- Cache Memory ٤- All of these

13- أي من الذواكر التالية غير متطاير ويمكن الكتابة فيها لمرة واحدة؟

١- RAM ٢- ROM

٣- PROM ٤- None of the above

14- الذاكرة من نوع EPROM تصفر بشكل عام باستخدام

١- Ultraviolet rays ٢- infrared rays

٣- Electrical pulse ٤- All the above

15- أي نوع من الذواكر التالية هو الأسرع؟

١- SRAM ٢- DRAM

٣- Cache Memory ٤- ١ & ٣

16- اشارات القراءة والكتابة التي توصل بالذاكرة الرئيسية تكون من

١- DB ٢- AB ٣- CB ٤- IO devices

٢-٥ أسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

1- أي ليس من الكتل الوظيفية للحاسوب؟

١- Memory ٢- A/D Converter

٣- ALU ٤- Input Port

2- أي ليس من العناصر الأساسية للمعالج الصغري؟

١- Flag register ٢- Control unit

٣- ALU ٤- Microcontroller

3- يرمز بالحاسوب الصغري لـ CPU بـ:

١- وحدة المعالجة الحسابية ٢- وحدة البرمجة المركزية

٣- وحدة معالجة التحكم ٤- وحدة المعالجة المركزية

4- ممر المعطيات بالحاسوب يستخدم من أجل

١- التحكم بعمليات الدخل والخرج والنقل للذاكرة فقط

٢- نقل المعطيات بين الذاكرة والمعالج ووحدات الدخل والخرج

٣- عنونة الذواكر فقط

٤- ١ و ٣ فقط

5- أي من المسجلات التالية يستخدم ليحفظ عنوان موضع في الذاكرة عندما تتوضع التعليمة التالية للتنفيذ؟

١- instruction register ٢- memory address register

٣- program counter ٤- memory data register

6- التجهيزة عالية السرعة لدى CPU للتخزين المؤقت خلال عملية المعالجة تدعى:

١- a register ٢- a bus

٣- a data bus ٤- None of the above

7- أي من الذواكر التالية من النوع التركيبي Combinational؟

١- RAM ٢- ROM ٣- ١ & ٢ ٤- لاشيء مما سبق

٢-٤-٨ برمجة المعالجات

تسمى اللغة التي يبرمج بها المعالج الصغري بلغة الأسمبلي (أو لغة التجميع) Assembly Language وتختلف البرمجة بهذه اللغة عن البرمجة باللغات عالية المستوى بأنها مرتبطة بالتركيبة العامة للحاسوب.

سنخصص لهذا الموضوع جزء كامل من هذا الكتاب، وبالتالي سنكتفي هنا بالاشارة إلى أن لغة التجميع تستخدم Mnemonics أي مختصرات مكونة من حرفين أو ثلاثة أو أربعة لتدل على التعليمة المراد تنفيذها. ويمكن التعبير عن بعض التعليمات بهذه اللغة بالشكل: ADD AX, 5H حيث إن المختصر ADD مثلا يدل على الأحرف الثلاثة الأولى من الكلمة Addition وهكذا الحال بالنسبة لبقية التعليمات. تمثل لغة الأسمبلي (التأويل) خطوة وسطى ما بين اللغات العليا ولغة الآلة. ويمكن تمثيل عملية انتقال برنامج من لغة عليا مثل لغة الـ Pascal إلى لغة التنفيذ (الآلة) بالشكل التالي:

Pascal Program A

↓ Compiler

Assembly version of A

↓ Assembler

Machine Code version of A

وبما أن لغة التجميع تمثل مرحلة واحدة قبل لغة الآلة، فإن هذه اللغة تحافظ عند ترجمة أي برنامج على درجة ما من السرعة، وبالتالي يمكن استخدامها في الأماكن التي لا يفضل بها استخدام اللغات العليا، أي في التطبيقات التي تتطلب سرعة في التنفيذ.

تختم هذا الفصل (وبعض الفصول اللاحقة) بفقرة حول بعض الاسئلة متعددة الاجابات، تعرض فيها بعض التساؤلات من الفصل وعدة اجابات لكل سؤال، يكون إحداها فقط هو الجواب الصحيح. وبالطبع هذه التساؤلات لاتشمل الفصل بكامله وانما هي عينة عن كيفية طرح بعض الأسئلة.

- معالجات ثنائية الأنوية Core Duo هو الجيل الأول من معالجات إنتل ذات الأنوية المزدوجة والموجهة للحاسوب المحمول وهو يحمل نواتين Pentium M.

- معالجات Pentium D وهي الجيل الأول من معالجات إنتل ذات الأنوية المزدوجة الموجهة للأجهزة المكتبية وهي تحمل نواتين ... Pentium 4 ويوجد منها إصدارة احترافية أكثر قوة وتسمى Pentium Xtreme Edition

- Core 2 Duo وهو جيل مختلف تماماً، يحمل نواتين بمعمارية جديدة ويوجد منه ما هو موجه للحواسيب المكتبية، وما هو موجه للمحمول، ويوجد منه إصدارة احترافية أكثر قوة تسمى Core 2 Extreme .

أما بالنسبة لمعالجات AMD فكانAMD أول من بدأ بعمل معالجات ثنائية الأنوية، مثل:

- النوع AMD Athlon 64 X2 وهو معالج يحمل نواتي Athlon 64 ويوجد منه إصدارة احترافية تسمى Athlon 64 FX وهو من أنجح المعالجات وأكثرها ثباتاً وحافظ على الصدارة فترة طويلة، إلى أن صدر معالج Core2 Duo.

- النوع الثاني AMD Turion 64 X2 وهو أول معالج بتقنية 64 bit للمحمول ويمثل معالج بنواتين Turion 64 ويتميز بسعره المنافس نوعاً ما وقوة أدائه وهو يقع بين Core Duo و Core2 Duo من ناحية الأداء...

ونشير هنا إلى أنه يوجد التباس وهو أن Intel Centrino Duo هذا المسمى في الحقيقة ليس اسم معالج، بل هو اسم نظام متكامل مخصص للمحمول، وهو عبارة عن لوحة رئيسية من Intel للمحمول+ بطاقة شبكة لاسلكية من Intel للمحمول + معالج محمول ثنائي النواة للمحمول = منظومة Centrino 2 Duo .

٢-٤-٦ معالجات RISC و CISC

ينفذ المعالج عادة مجموعة التعليمات التي تصدر من البرنامج ، والتعليمات يمكن أن تكون بسيطة (مثلا القيام بعملية جمع)، أو معقدة (كالقيام بسلسلة من العمليات المترابطة). فالبرنامج هو عبارة عن مجموعة من التعليمات المترابطة التي تؤدي في مجملها عملا مفيداً وهو القائد والمحرك للمعالج. يخرج المعالج من المصنع " متعلماً " هذه التعليمات، بستطيع تنفيذ أي برنامج يحوي أي تركيب من هذه التعليمات وهذا هو السبب في أن الحاسوب يستطيع القيام بأي عمل (بحسب البرنامج). انقسم مصنعو المعالجات الصغرية في فلسفة بناء المعالج إلى فريقين:

الأول: زودوا معالجاتهم بالكثير من التعليمات المعقدة وتسمى هذه المعالجات معالجات CISC (Complex Instruction Set Computer).

والثاني: زودوا معالجاتهم بعدد قليل من التعليمات البسيطة وتسمى هذه المعالجات معالجات RISC (Reduced Instruction Computer)

إن الحكم على من منهما أسرع ليس شيئاً سهلا وإن ذلك يعتمد على تصميم المعالج نفسه ككل وعلى برامج التجميع المستخدمة في إنتاج البرامج وعلى عوامل أخرى، واليوم أصبح مصنعوا المعالجات يتجهون إلى استعمال كلا الفلسفتين معاً، وأصبح الفارق بينهما يندثر شيئاً فشيئاً. ما زالت المعالجات الحديثة تفهم نفس التعليمات التي تفهمها المعالجات القديمة. ففي كل مرة ينتج المصنعون (مثل شركة إنتل) جيلا جديداً من المعالجات يتم إضافة كمية من التعليمات لتحسين الأداء.

يمكن لمصنعي المعالجات أن يجعلوا معالجاتهم تعمل كمعالجات CISC ظاهرياً بينما تعمل في الحقيقة كمعالجات RISC ، ويتم عمل ذلك بإضافة وحدة خاصة في المعالج تقوم بتحويل تعليمات CISC إلى RISC ومن ثم يقوم المعالج بتنفيذها.

٢-٤-٧ المعالجات متعددة النوى

يقصد بمعالج متعدد النوى أي معالج يحمل بداخله أكثر من نواة. ويعود قسم من هذه المعالجات لـ Intel والقسم الآخر لشركة AMD . فبالنسبة لمعالجات Intel نميز بين

فتح الباب لمعماريات أصغر تمكن من وضع عدد أكبر من الترانزستورات في مساحة أقل وتصنيع معالجات أقوى بتكلفة منخفضة. بالطبع كلما كانت معمارية المعالج أصغر كان استهلاك الطاقة أقل (فولطية أقل للتيار)، وكانت الحرارة الناتجة من المعالج أقل مما يخفف من مشاكل التبريد.

٢ – ٤ – ٥ تبادل البيانات بين المعالج والأجزاء الأخرى

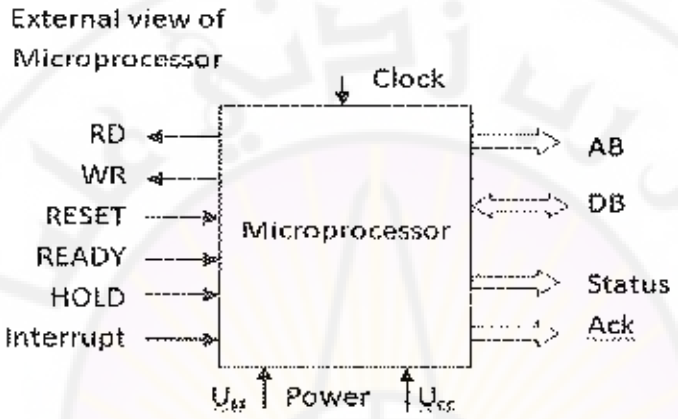
عند جمع عددين بالحاسوب مثلا ليس المهم بالنسبة للحاسوب حجم الأرقام وإنما المهم هو إيجاد الأرقام المراد جمعهما وإحضارها من الذاكرة العشوائية بأسرع وقت ممكن. فإذا كان تردد ناقل معين 100 MHz فهذا معناه أنه يرسل 100 مليون نبضة كهربائية في الثانية الواحدة مما يمكنه من إرسال معلومات أكثر من ناقل آخر يعمل بتردد 66 ميجاهرتز مثلا (إذا افترضنا أن عرض الناقل متساوي في الحالتين) يقوم المعالج بتبادل البيانات مع الأجزاء الأخرى عبر الناقل، وفيما يعمل المعالج بسرعة 700 ميجاهرتز مثلا لا تعمل باقي أجزاء الحاسوب بهذه السرعة الكبيرة لان ذلك من شأنه أن يجعل الحاسوب ككل غالي الثمن. وحتى يتم تبادل البيانات بين المعالج وناقل النظام الأقل سرعة بدون أي أخطاء لابد من التنسيق بينهما انطلاقاً من:

– أن الناقل يعمل في أغلب الأحيان بسرعة 66 أو 100 ميجاهرتز، فيما تبلغ سرعة المعالجات عدة أضعاف ذلك (مثلا 500 ميجاهرتز)

– خلال تعيين نسبة لعدد دورات ساعة (تردد) المعالج إلى عدد دورات ساعة الناقل وهو ما يسمى بعامل المضاعفة والذي يمكن تعريفه كنسبة بين تردد المعالج وتردد ناقل النظام ويكون عادة عدد صحيح أو عدد يقبل القسمة على 0.5 . ومن الأمثلة على هذا المعامل: 2 – 2.5 – 3 – 3.5 – 4 – 4.5 – 5 – 5.5 ولا يكون مثلا 2.3. فمثلا في حالة المعالج بتردد 500 ميجاهرتز فإن تردد الناقل هو 100 ميجاهرتز ومعامل المضاعفة في هذه الحالة هو5 لان 100 × 5 = 500 وهكذا فإذا لم يستطع الناقل توصيل البيانات بسرعة كافية فإن ذلك يعني عدم الاستفادة بصورة تامة من قدرات المعالج حيث إن المعالج يكون أسرع من الناقل في تلقي البيانات.

التركيبة الخارجية للمعالج الصغري

ويقصد بالتركيبة الخارجية للمعالج External Configuration of µP جملة الإشارات الداخلة من الوسط الخارجي إلى المعالج والخارجة منه لربط المعالج مع غيره من دارات الدعم بجملة واحدة . يبين الشكل 2-I البعض من هذه الإشارات التي سنصادفها بكل معالج صغري، حيث يخصص لكل منها مأخذ pin مستقل من المعالج.



الشكل2-I: الإشارات الأساسية لمعالجات صغرية بكلمة 8 bits

بالطبع عدد المآخذ pins للدارة المتكاملة التي تمثل معالج ما، يختلف بحسب نوع المعالج، وهنا نشير فقط إلى بعض الإشارات المشتركة التي يمكن أن نصادفها بأي معالج بغض النظر عن نوعه. ومن هذه الإشارات المشتركة:

- Power أي التغذية: حيث كل معالج يلزمه بالحد الأدنى خطين للتغذية المستمرة وهما Ucc and Uss ~ GND (بعض المعالجات يكون فيها أكثر من مأخذين).
- Clock الساعة: يلزم لعمل كل معالج نبضات ساعية بتواتر معين لتأمين التزامن اللازم من مذبذب خارجي (مثل الكريستال).
- Reset التصفير: كل معالج يلزمه إشارة لتصفير محتوى المسجلات الداخلية وجعله يبدأ في عمله من عنوان محدد .

- إشارات تحكم لعمليتي القراءة والكتابة RD و WR لإدخال المعطيات من الذاكرة أو البوابات إلى المعالج لمعالجتها وإخراجها بعد المعالجة إلى الوسط الخارجي.
- Ready للجاهزية والتوافق مع الدارات البطيئة وأن هذه الإشارة ضرورية في كل معالج لتأمين التزامن لعمل المعالج مع التجهيزات البطيئة.
- Hold إشارة دخل يطلب عبرها من المعالج ترك الـ Bus للعمل في نظام الولوج المباشر للذاكرة DMA- Direct Memory Access.
- Interrupt (أو INTR) إشارة أو أكثر لمقاطعة المعالج من تجهيزات خارجية.
- AB من Address Bus ممرات للعنونة وتمثل مجموعة الخطوط التي يرسل عليها المعالج العنوان المناسب لأي دارة سيتعامل معها وبالتالي فان عدد هذه الخطوط هو الذي يحدد المجال العنوني لأي معالج مع البوابات أو ضمن الذاكرة.
- DB من Data Bus ممرات لنقل المعطيات وعددها يحدد طول كلمة المعالج وتكون عادة هذه الخطوط مخصصة لنقل البيانات بالاتجاهين.
- Ack من Acknowledge ويقصد بها مجموعة من إشارات الخرج كجواب على بعض الإشارات الداخلة إلى المعالج، حيث تخرج إجابة من المعالج لبعض الإشارات الأساسية لقيادة عمل الدارات الأخرى المرتبطة مع المعالج مثل إشارة:

INTA جواب لـ INTR و WAIT جواب لـ READY و

HLDA جواب لـ HOLD و CLKOUT جواب لـ CLKIN و ...

ونشير هنا إلى أن الإلمام بالتركيبة الداخلية والخارجية للمعالج الصغري ونوعية المسألة المطلوب حلها أو التحكم بها، تأتي بالمرتبة الثانية من حيث الأهمية عند انتقاء معالج لتطبيق معين، بعد تحديد طول كلمة المعالج والتواتر الذي يعمل عليه.

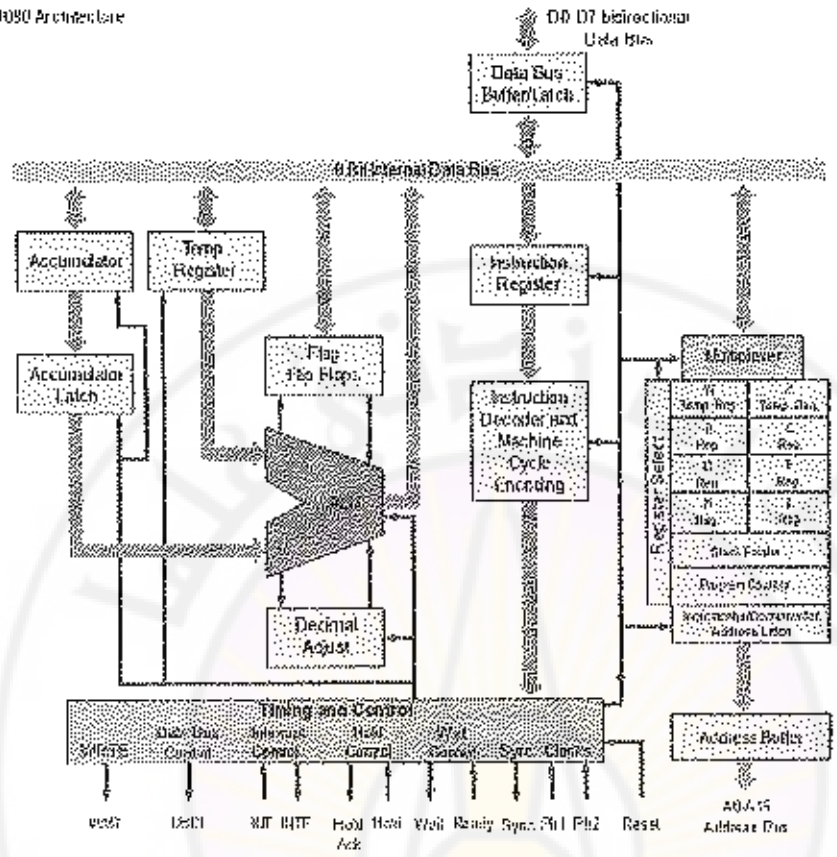
تمتاز التركيبة الداخلية للمعالجات الصغرية بطول كلمة 8 bits مثل المعالج 8080 بالبساطة وتختلف عن التركيبة الداخلية للمعالجات الأحدث بأمور عدة يأتي في مقدمتها عدم توفر وحدة الذاكرة المخبئية Cash Memory ضمن تركيبة المعالج.

٤-١ التركيبة الداخلية للمعالج 8080

نبدأ مسيرة تعرّف المعالجات الصغرية بتعرّف المعالج 8080I رغم قدمه، للمباشرة بتعرّف أولى المصطلحات والمفاهيم التي تستخدم بالمعالجات بشكل عام، والتي ستمر معنا لاحقا في المعالجات الأحدث، لان كل معالج من معالجات إنتل يستخدم نفس المصطلحات، وبالتالي التعرف بشكل جيد على إحدى معالجات هذه الشركة يساعد كثيرا في تعرّف أي معالج آخر من إنتاجها.

بعد ظهور المعالج 8080I الذي كان يمتاز بقدرة ضعيفة للـ BUS تم بعد عدة أشهر استبداله بالمعالج 8080AI الذي امتاز بمخارج ذات فاعلية أكبر. يمثل المعالج 8080AI وحدة معالجة مركزية CPU بتركيبة متوازية بعرض 8bits وتضم التركيبة الداخلية لهذا المعالج ما يقارب 5000 ترانزستور متكاملة على شريحة Chip واحدة. يصنع المعالج 8080AI بالتقانة NMOS وبالتغليف DIL-40 أي دارة متكاملة بصفين متوازين من المآخذ (بمعدل 20 pins في كل صف). يعدّ المعالج الصغري 8080A نموذجاً للمعالجات بذاكرة مسجلات مع جملة تعليمات موجهة للعمل مع هذه المسجلات. يظهر الشكل 1-4 المخطط الصندوقي للتركيبة الداخلية لهذا المعالج الذي كان في إحدى الفترات معياراً عالمياً للمعالجات الصغرية.

يتضمن المخطط الصندوقي للمعالج ممر نقل داخلي بعرض 8 bit تربط عليه مختلف المكونات الداخلية للمعالج. يتضح من المخطط الصندوقي للمعالج 8080A بأن التركيبة الداخلية له مكونة من عدة كتل نستعرض باختصار مفهوم كل منها كالتالي:



شكل 4-1: المخطط الصندوقي للبنية الداخلية للمعالج الصغري 8080

1- وحدة الحساب والمنطقALU: ويرتبط مع هذه الوحدة بعض المسجلات الخاصة بعملها والكتلة Decimal Adjust الخاصة بإصلاح النتائج خلال العمليات الحسابية، وكذلك مسجل خاص بالمؤشرات Flag Register الذي يحتوي على خمسة مؤشرات أساسية ويستخدم لمعرفة حالة المعالج بأي لحظة، حيث إن دلول كل مؤشر Flag من مؤشراته الخمسة هو bit 1 وترتبط قيمته بنتيجة العمليات ضمن ALU. تستخدم قيم مؤشرات هذا المسجل كما سنجد لاحقا، كشروط لتفريع (انتقال) البرنامج. يظهر الشكل 4-2 توزع المؤشرات على مستوى البايت، وهذه المؤشرات هي:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		C

الشكل2-4: الشكل العام لمسجل المؤشرات لدى المعالج 8080

- C من Carry الحمل: ويصبح هذا المؤشر بحالة Set أي بالقيمة واحد إذا حصل نقل أثناء العمليات الحسابية في ALU من الخانة المقابلة لـ MSB.

- P من Parity الازدواجية (الندية): ويقصد بها عدد الواحدات ضمن كلمة المعالج وهنا نميز بين نوعين من الندية هما:

الندية الزوجية Even Parity، ويرمز لها بـ PE وتكون بالقيمة واحد (PE = 1) إذا كان عدد الواحدات في البايت زوجي.

والندية الفردية Odd Parity، ويرمز لها بـ PO وهي تقابل وجود عدد فردي من الواحدات في البايت.

- AC من Auxiliary Carry الحمل المساعد: ويصبح واحد إذا حصل نقل من النصف السفلي للبايت (الـ Nibble السفلي) إلى النصف العلوي ضمن البايت الواحد، كما ويأخذ AC القيمة صفر في حال عدم حدوث نقل من البيت الرابع. لا يستخدم هذا المؤشر في تعليمات تفريع البرنامج.

- Z من Zero صفر: ويصبح بالقيمة واحد (ZF = 1) إذا كانت نتيجة العملية الحسابية ضمن ALU صفراً، وفي حال عكس ذلك يكون المؤشر ZF = 0.

- S من Sign الإشارة: ويصبح هذا المؤشر بالقيمة واحد إذا كانت نتيجة العملية الحسابية في الـ ALU سالبة (أي قيمة البت MSB = 1)، أما إذا كانت موجبة فتكون قيمته SF = 0.

2- ذاكرة المسجلات العامة General Purpose Registers: وهي مسجلات متكاملة على الـ Chip تمثل ذواكر سريعة، تستخدم للتخزين المؤقت، وتساعد في انجاز مختلف عمليات المعالج. يبلغ عدد المسجلات العامة للمعالج 8080A ثمانية وطول كل منها 8 bit وهي:

– المراكم Accumulator ويرمز له بالرمز A . وقد سبق وتمت الإشارة إلى مفهوم المراكم كمسجل للاستخدامات العامة وبعض الاستخدامات الخاصة.

– المسجلات الستة: B, C, D, E, H, L وتستخدم لمختلف العمليات على المعطيات التي طولها 8 bit. وللتعامل مع معطيات بطول 16 bit يمكن دمج كل مسجلين من هذه المسجلات الستة مع بعضهما البعض لتشكل Pair Registers (زوج مسجلات) وفق ما هو مبين بالجدول I–4.

الجدول I–4: أزواج المسجلات الممكنة لدى المعالج 8080A

Individual	B C	D E	H L
Combination	B&C	D&E	H&L

بحيث يمكن على سبيل المثال دمج المسجلين L وH مع بعضهما البعض ليشكلا زوج المسجلات HL الذي يمثل مراكم خلال جمع بايتين معاً أو لعنونة المسجل المساعد الوهمي M وهو المسجل الثامن بالنسبة للمعالج 8080A. أزواج المسجلات المتبقية BC وDE يمكن النظر لكل منهما أيضاً على أنه مسجل واحد بطول 16 bits.

– بقية المسجلات مثل PC و SP سبق وتمت الإشارة لهما وهي غير قابلة للتقسيم.

– مسجل التعليمات RI وهو بطول 8bit ويمثل أحد الجزأين الأساسيين للمنظم Controller وهو يحافظ على الرمز العملياتي للتعليمة طيلة مدة تنفيذها.

– مفكك ترميز التعليمة Instruction Decoder وهو الجزء الثاني للمنظم المسؤول عن قيادة فعالية المعالج خلال تنفيذ التعليمة.

– دارات التحكم والتزامن Timing and Control وهي الجزء الذي يقوم بتوليد الإشارات الأساسية للمعالج، يصدر من هذه الكتلة 12 إشارة قيادية بعضها دخل للمعالج اشارات المقاطعة أو التصفير والجاهزية والبعض الآخر خرج منها مثل اشارات التحكم بالقراءة والكتابة وغيرها.

٤-٢ التركيبة الخارجية للمعالج 8080A

ندرس تحت عنوان التركيبة الخارجية للمعالج، مجمل الإشارات المطبقة على كل مأخذ من مآخذه، حيث بالنسبة للمعالج 8080A كافة الإشارات المطبقة على مآخذه الأربعين متوافقة مع TTL باستثناء إشارة الساعة، إذ أن السوية الفولطية للصفر (أي log0) هي: 0.8V÷ 1- ، وللواحد (أي Log1) هي 13V ÷ 9. يظهر الشكل3-4 توزع إشارات المعالج 8080A على مآخذه (Pins) مع مستوى تفعيل كل إشارة، ونوعها ان كانت دخل أم خرج.



الشكل3-4: توزع الإشارات على مآخذ المعالج 8080A

نستعرض فيما يلي مجمل الاشارات الداخلة والخارجة من المعالج 8080A وفق التالي:

- يملك المعالج أربعة مآخذ لجهود التغذية 5V- ,5V+ ,12V+ ، إضافة للأرضي GND (0 V).

- إشارتان لمداخل الساعة Ø1, Ø2 ، حيث للمعالج 8080A سلسلتان من النبضات الساعية غير المتراكبة فوق بعضها وإن دور الساعة لهذا المعالج يتراوح (بحسب قيمة الكريستال الذي يوصل بالمعالج) بالقيمة مابين 2µs ÷ 0.48 (إذ ان كل معالج يعمل ضمن مجال محدد من الترددات). ينصح بالنسبة للمعالج 8080A استخدام الدور 0.5µs أو ما يعادل 500ns وهو ما يقابل وصل كريستال مع المعالج بقيمة 18MHz كما سنجد لاحقا (على اعتبار إن الدارة 8224 المسؤولة عن تأمين الساعة اللازمة للمعالج تقوم بتقسيم قيمة تواتر الكريستال الخارجي على 9).

- إشارة خرج Sync للتزامن Synchronization وهي تشير إلى بداية كل نبضة ساعية (takt) من كل Machine cycle. وبالتالي تستخدم هذه الاشارة لقيادة عمل دارات الدعم 8224 المسؤلة عن تأمين الاشارتين Ø2, Ø1 وبعض الاشارات الأخرى مثل RESET و READY.

- ثمانية مآخذ لخطوط نقل المعطيات $D_0 \div D_7$، للنقل بالاتجاهين.

- ستة عشر مأخذاً لممر العناوين $A_0 \div A_{15}$ وهي إشارات خارجة من المعالج باتجاه الدارات المتممة له، تسمح له بعنونة $2^{16} = 65536$ مواضع أو بايت ذاكرة وهذا ما يعادل 64KB أو عنونة $2^8 = 256$ بوابة للدخل والخرج.

وهنا نشير إلى نقطة هامة أثناء تصميم منظومة باستخدام المعالج 8080A تتمثل بالتالي: عند تعامل المعالج مع الذاكرة تستخدم الخطوط الستة عشر لعنونة أي موضع من ذاكرة يصل حجمها لـ 64KB، أم عند تعامل المعالج مع البوابات فيكفي من ممر العناوين استخدام أول ثمانية خطوط عنونة $A_0 - A_7$، لان المعالج لا يستطيع أن يعنون أكثر من 256 بوابة. بالتالي عند التعامل مع البوابات تكون القيم الموجودة على خطوط العنونة المتبقية $A_8 - A_{15}$ هي نفس القيم الموجودة على $A_0 - A_7$ (مرآة لها أي مكررة)، بحيث إن القيمة التي تكون موجودة على A_0 هي نفس القيمة التي تكون موجودة على A_8 وهكذا.

- إشارة واحدة لـ Reset للتصفير وهي إشارة دخل تقوم بتصفير المعالج وتستدعي قفز المعالج إلى العنوان 0000H في الذاكرة الرئيسية، كما وتقوم بتصفير بعض الإشارات الأخرى مثل INTE و HLDA كما سنجد لاحقا.

يتم عادة أثناء تصميم منظومة حاسوب صغري باستخدام المعالج 8080A تشكيل منطق خارجي لإشارة RESET لتحديد مدة تفعيلها ولتشغيلها من خلال كبسة خارجية.

- إشارة لـ DBIN من Data Bus In وهي إشارة خرج، تشير إلى أن ممر المعطيات هو في نظام الإدخال، بالتالي فهي تمثل Read حتى ولو لم تسمَ كذلك.

- إشارة لـ WR أي الكتابة Write وهي إشارة خرج من المعالج تقوم بإعلام الدارات الأخرى بأنه على ممر المعطيات توجد معطيات خرج جاهزة للكتابة.

- إشارة لـ Ready وهي إشارة دخل للمعالج تستخدم لتأمين التزامن مع الذواكر والتجهيزات المحيطية البطيئة. بمعنى أن هذه الإشارة عندما تكون غير فعالة، فهذا يعني أن المعطيات التي يطلبها المعالج من الذاكرة أو البوابات تكون غير جاهزة بسبب بطء هذه التجهيزات، ولا بد للمعالج أن ينتظر بعض الوقت حتى يتم تأمينها.

- إشارة لـ Wait وهي إشارة خرج من المعالج إلى الذواكر أو التجهيزات المحيطية البطيئة. وتُمثل جواباً لإشارة عدم الجاهزية أي الانتظار. وهي تُعلم الدارات الأخرى بأن المعالج في حالة انتظار إلى حين تأمين المعطيات المطلوبة.

- إشارة لـ HOLD وهي إشارة دخل للمعالج تنقله إلى حالة HOLD وفي هذه الحالة تتوقف فعالية المعالج مع الدارات الخارجية نظراً لتركه الممرات الخارجية.

- إشارة واحدة لـ HLDA وهي إشارة خرج من المعالج كجواب لإشارة HOLD ومن خلالها يتم التأكيد على أن المعالج يُحقق المطلب HOLD بترك الممرات.

- إشارة لـ INT وهي إشارة دخل للمعالج من التجهيزات الخارجية تقوم بإجراء مقاطعة Interrupt لعمل المعالج في حال طلب المقاطعة.

- إشارة لـ INTE وهي إشارة خرج كجواب لإشارة INT تؤشر إلى حالة القلاب الداخلي الذي يسمح بقبول مطلب المقاطعة. بمعنى إذا كانت INTE = 1 سيكون المطلب مخدماً . ويمكن التحكم بهذا القلاب في البرنامج بالتعليمة EI و DI وإذا كان مطلب المقاطعة مقبولا يكون ذلك دوماً بعد الانتهاء من التعليمة التي هي قيد التنفيذ.

٤-٣ التزامن لدى المعالج 8080A

يقصد بالتزامن لدى معالج ما تعرّف حالة كل إشارة من إشاراته مع مرور الساعة التي يعمل عليها المعالج، حيث إن لكل إشارة من إشارات المعالج الأساسية موجة زمنية يحدد من خلالها زمن تفعيل الإشارة وارتباطها بغيرها من الإشارات الأخرى. وهذه الموجات الساعية تكون محددة من قبل الشركة الصانعة للمعالج، ويشار إليها عند دراسة المعالج للتعرف عن قرب بآلية عمل المعالج عند وضعه بالخدمة الفعلية.

سوف لن ندخل بالدراسة التفصيلية للمخططات الزمنية للمعالجات، وقد نعرض البعض من أشكال الموجات لبعض الإشارات بقصد إظهار استجابتها مع ساعة المعالج لأنها قياسية (وتدرس عادة موجات التزامن لإشارات المعالج أثناء عملية التصميم الفعلية). يظهر الشكل 4-4 على سبيل المثال قسم من المخطط الزمني لعمل المعالج 8080A، تحديدا الموجات Waveforms الخاصة بعمليتي القراءة والكتابة، أي بالإشارتين DBIN و WR' وأزمنة تفعيل كل منهما.

ولفهم المخطط، نشير إلى أن ساعة المعالج، أي الإشارتين Ø1, Ø2 (t1, t2) تقسم كل منهما إلى عدة حلقات ساعية M1 ÷ M5 من (Machine Cycle اختصارا MC) وكل حلقة تقسم بدورها إلى عدة أدوار (Takt) من T1 ÷ T5.

وبحسب نوع التعليمة يلزم لتنفيذ كل من تعليمات المعالج من ثلاث وحتى خمس حلقات تعليمة بثلاث وحتى خمس أدوار، فمثلا التعليمة NOP تحتاج إلى ثلاثة MC بينما تعليمة CALL تحتاج لخمسة MC.

يلزم في بعض الحالات الخاصة (أثناء تعامل المعالج مع تجهيزات بطيئة) إضافة بعض الأزمنة (حالات انتظار TW) والتي كل منها يستغرق عدداً صحيحاً من الأزمنة تحدد بالإشارات الخارجية.

شكل4-4: قسم من المخطط الزمني لفعالية المعالج الصغري 8080

تشكل Ø1, Ø2 المعالج 8080A سلسلتان من النبضات الساعية غير المتراكبة التي تنظم عمل المعالج. الشكل العام للإشارتين Ø1, Ø2 مع كل من ممر العناوين والمعطيات لحلقة القراءة لدى المعالج 8080A وفق الشركة الصانعة مبينة بالشكل أعلاه. بالطبع لكل من الإشارات READY, WAIT, HOLD, HLDA, INT, ... موجتها الخاصة، وأن بعض الإشارات تقاد بالفاز الساعي Ø1 والبعض الآخر يقاد بالفاز Ø2 . فمن أجل دور الساعة العادي للمعالج 8080A والذي يعادل 500ns = 0.5µs يكون طول حلقة التعليمة 2µs (بشكل وسطي) حيث إن أقصر M هو 1.5µs، لان معظم التعليمات تحتاج بالحد الأدنى من M3 ÷ M1 وإن بعض التعليمات تحتاج إلى M5 ÷ M1 بـ 18 دور أي بزمن إجمالي يعادل حتى 9µs . بالطبع في كل T من كل MC يحدث بعض الأمور وبخاصة المتعلق منها بتشكيل كلمة الحالة للمعالج.

٤-٤ صيغ المعطيات والتعليمات لدى المعالج 8080A

تشمل عملية التعرف على أي معالج صغري بالإضافة لتعرف التركيبة الداخلية للمعالج وإشاراته الأساسية (الجانب الصلب Hardware)، تعرف آلية برمجة المعالج باستخدام التعليمات الخاصة به (الجانب اللين Software).

لكل معالج صغري جملة خاصة به من التعليمات Instruction Set وتعمل كل تعليمة من جملة تعليمات المعالج بنمط محدد للعنونة. يرتبط عدد أنماط العنونة لأي معالج ونوعها بالمسجلات العائدة للمعالج من طرف وبطرق ولوج المعالج إلى الذاكرة من طرف آخر. يحدد من خلال أنماط العنونة آلية ولوج المعالج إلى المعطيات سواء أكانت من الذاكرة أو البوابات.

من أنماط العنونة المعروفة لكل معالج نذكر، نمط العنونة المباشر Direct وغير المباشر Indirect ونمط العنونة الفوري Immediate Addressing Mode وغيرها، حيث إن عدد أنماط العنونة ونوعها يختلف من معالج لآخر.

وبالرغم من أننا سوف لن نتطرق في هذا الكتاب إلى البرمجة بلغة الأسمبلي للمعالج 8080A ، نشير باختصار إلى أن عدد التعليمات المنشورة للمعالج 8080A يبلغ وفق

الشركة الصانعة 78 تعليمة، مقسمة إلى عدة مجموعات بحسب الوظيفة التي تقوم بها تعليمات كل مجموعة وهذه المجموعات هي:

- مجموعة تعليمات المراكم (وهي تعمل بنمط عنونة مباشر، وغير مباشر وفوري).
- مجموعة تعليمات Move, Load and Store تسمح بالتعامل مع معطيات بطول 8, 16 bit بين الذاكرة ومسجلات المعالج للاستخدامات العامة باستخدام أساليب العنونة: مباشر، غير مباشر وفوري.
- توفر تعليمات المعالج 8080A القدرة على التفريع إلى أقسام مختلفة من البرنامج بتعليمات Jump المشروطة وغير المشروطة، وكذلك استدعاء البرامج الفرعية Call والعودة Return من البرامج الفرعية بشكل مشروط وغير مشروط.
- توفر أيضاً تعليمات المعالج 8080A إمكانية زيادة أو تنقيص الذاكرة والمسجلات، وتوسيع العملية لتشمل أزواج المسجلات Pair Registers(BC, DE, HL) و SP.
- تتوفر أيضاً إمكانية تدوير Rotate المراكم إلى اليمين أو اليسار حول بيت النقل.
- يوفر المعالج تعليمات لإدخال المعطيات وإخراجها I/O Instruction.
- توجد مجموعة خاصة من تعليمات المعالج 8080A تكمل جملة تعليماته، مثل:

تعليمة XCHG – Exchange التي تبدل بين محتوى زوجين من المسجلات.

تعليمة NOP – No Operation لا تفعل شيء.

وتعليمة HALT التي توقف تنفيذ البرنامج

وتعليمة DAA– Decimal Adjust Accumulator التعديل الحسابي للنتيجة

وتعليمة STC– Set Carry bit (Flag) التي تضع بيت النقل بحالة الواحد.

وتعليمة CMC– Complement Carry bit التي تشكل متمم بيت النقل

وتعليمة CMA – Complement Accumulator تشكل المتمم الأحادي للمراكم

أما بخصوص صيغ المعطيات المعمول بها لدى المعالج 8080A، فإن المعطيات التي يتعامل معها المعالج تخزن بالذاكرة الرئيسية بصيغة 8 bit، وبالتالي المعطيات (من بيانات وتعليمات) التي تنقل عبر ممر المعطيات يجب أن تكون بواحدة من الصيغ المبينة أدناه،

حيث إن كلمة المعطيات Data Word للمعالج 8080A بطول 8 bit تأخذ الترتيب التالي: $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$.

تعليمات المعالج 8080A يمكن أن تكون بطول 1, 2, 3 Bytes. التعليمات متعددة البايتات يجب أن تخزن وراء بعضها البعض في ذاكرة البرنامج، بعد ذلك تحدد صيغة التعليمة التي ترتبط بالعملية التي تنفذ بواحدة من الصيغ التالية:

- الصيغة الأولى: التعليمات بطول 1 Byte ولها الشكل التالي:

$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	وتسمى Op code

وكأمثلة على هذا النوع من التعليمات نذكر: تعليمات النقل من مسجل إلى مسجل، الدوران المنطقي والحسابي، Return, Push, Pop وتعليمات السماح بالمقاطعة.

- الصيغة الثانية: التعليمات بطول 2 Bytes ولها الشكل التالي:

$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	Op code
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	Operand

وكأمثلة على هذا النوع من التعليمات نذكر: تعليمات النمط الفوري وتعليمات I/O

- الصيغة الثالثة: التعليمات بطول 3 Bytes ولها الشكل التالي:

$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	Op code
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	Low address or Operand1
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1D_0$	Low address or Operand2

وكأمثلة على هذا النوع من التعليمات نذكر: تعليمات Jump, Call, Load, ...

Damascus University

الفصل الخامس: دارات الدعم غير المبرمجة للمعالج 8080A

يمكن تصنيف ما تم إنتاجه في البدايات من دارات مختلفة من قبل شركة إنتل بالتالي:

- ميكروكمبيوتر من chip واحد مثل الدارات 8048/8748 لتطبيقات محددة، بحيث تحتوي كل دارة على ذواكر وبوابات مع المعالج على نفس الشريحة.
- معالجات ربط Microprocessor Interfacing مثل المعالجين 8041/8741.
- معالجات وحيدة البلورة للتطبيقات العامة مثل المعالجين 8085 ,8080.
- معالجات شرائحية Slice processor مثل عناصر السلسلة 3000 المصنعة بالتقانة TTL على عكس بقية معالجات إنتل المصنعة اعتمادا على تقنيات MOS .
- جملة من دارات الدعم البسيطة مثل: مرمز ات، مقويات إشارة، دارات ربط وغيرها.
- جملة من دارات الدعم المركبة القابلة للبرمجة

سبق وأشرنا إلى أن كل معالج صغري مثل 8080A لا يستطيع العمل بمفرده، ولابد له من بعض الدارات المساعدة لانجاز منظومة متكاملة قادرة على انجاز شيء ما. من هنا نجد أن معظم الشركات المصنعة للمعالجات الصغرية تقوم مع كل معالج تنتجه بطرح البعض من الدارات المتممة له، مشكلة بذلك تركيبة محددة. وبالتالي يقصد بتركيبة ما مثل التركيبة MCS – 80 دارة المعالج الصغري 8080A وكافة دارات الدعم العائدة له من شركة Intel. فمثلا بلغ عدد دارات الدعم من شركة إنتل لمنتجاتها من المعالجات حتى نهاية عام 1977 ما يقارب 40 دارة دعم، خصص منها:

- 21 دارة دعم للتركيبة MCS-08 و
- 28 دارة دعم للتركيبتين MCS-41 و MCS48،
- 35 دارة دعم للتركيبة MCS-80 و
- 34 دارة دعم للتركيبة MCS-85

أي أن بعض هذه الدارات يصلح للاستخدام مع كافة التراكيب، وأن البعض الأخر يصلح للاستخدام مع قسم من هذه التراكيب وثالث لا يصلح للاستخدام إلا مع تركيبة محددة.

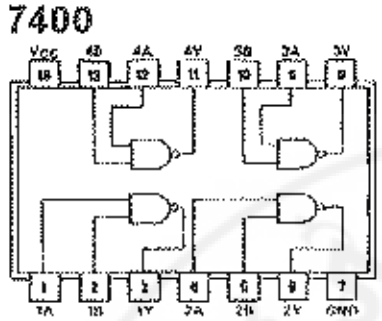
بالطبع أثناء تصميم منظومة متكاملة باستخدام معالج ما يمكن بالإضافة للدارات الخاصة من الشركة الصانعة للمعالج أيضاً استخدام بعض الدارات العامة المصنعة من شركات أخرى لا تنتج معالجات.

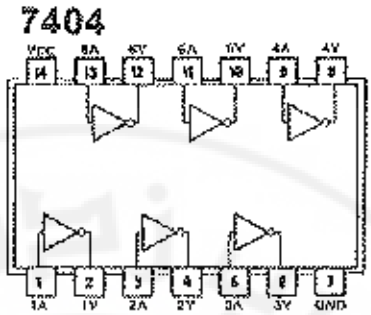
يظهر الجدول 1-5 بعض الدارات المتكاملة والمنتقاة من رتبة SSI و MSI بالتقانة TTL التي تصلح للاستخدام في تشكيل منطق تحكمي ضمن منظومات حاسوبية، أو للاستخدام كبوابات دخل/خرج مع المعالج 8080A وغيره من المعالجات الأخرى.

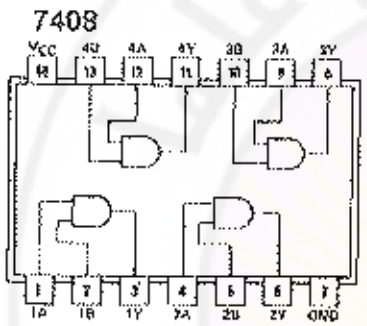
الجدول 1-5: بعض الدارات المتكاملة من رتبة SSI و MSI بالتقانة TTL

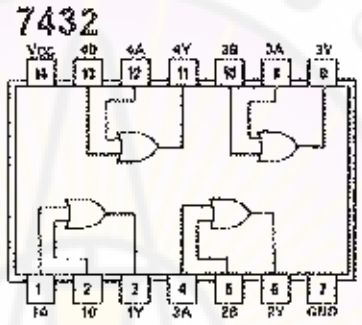
وظيفة الدارة	رقم الدارة	وظيفة الدارة	رقم الدارة
أربع بوابات XNOR (التكافؤ)	74135	أربع بوابات NAND كل منها بمدخلين	74 00
3 to 8-Line Decoder	74138	أربع بوابات NOR كل منها بمدخلين	74 02
Dual 2-to-4 Decoder/Demultiplexer	74139	ست بوابات NOT	74 04
4 to 16-Line Decoder	74154	أربع بوابات AND كل منها بمدخلين	74 08
Octal Buffer/Line Driver (Noninverted)	74244	أربع بوابات OR كل منها بمدخلين	74 32
Octal Bus Transceiver	74245	BCD-Seven Segment Decoder	74 46
8 bit universal Register Shift/Storage	74299	4 bits Comparator	7485
Octal Transparent Latch	74373	أربع بوابات XOR	74 86

يظهر الشكل 1-5 توزع مآخذ بعض هذه الدارات (بوابات ومفكك ترميز ومسجلات) التي ستستخدم لاحقا في بعض الأمثلة.

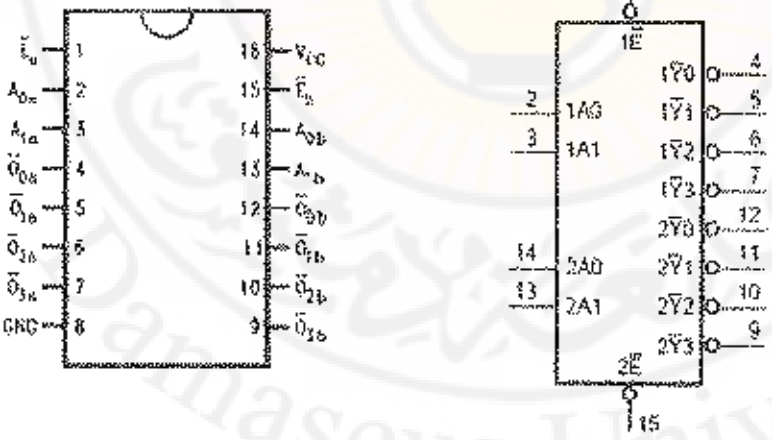






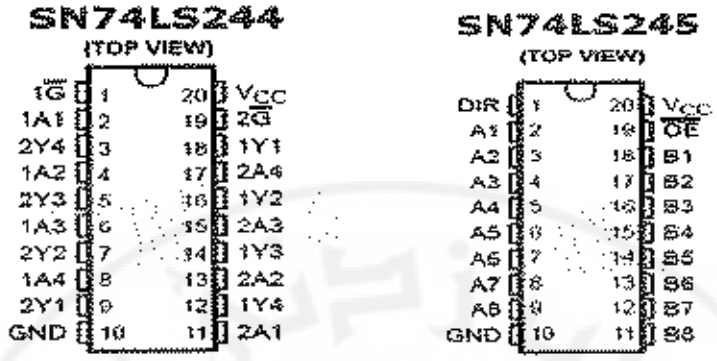


74139 :Dual 2-to-4 Decoder/Demultiplexer



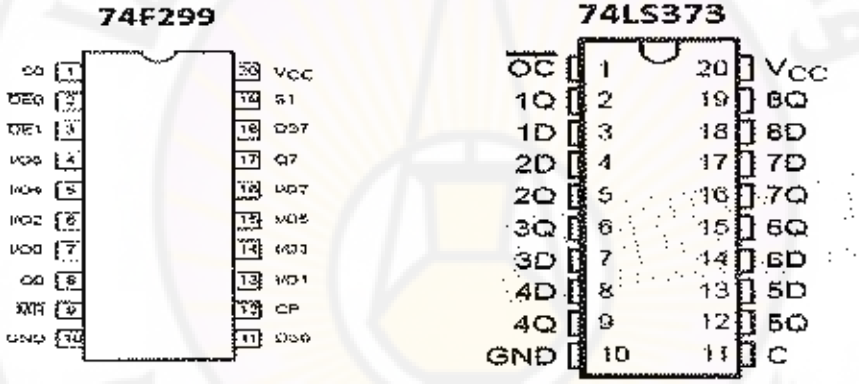
74LS244: Octal Buffer/Line Driver (Noninverted) and

74LS2455: Octal Bus Transceivers



74F299: 8 bit Universal Shift/Storage Register and

74LS373 Octal Transparent Latch



الشكل1-5: مخطط توزع مآخذ بعض الدارات المتكاملة

ويمكن بشكل عام تقسيم مجموعة دارات الدعم للمعالج 8080A إلى مجموعتين هما:

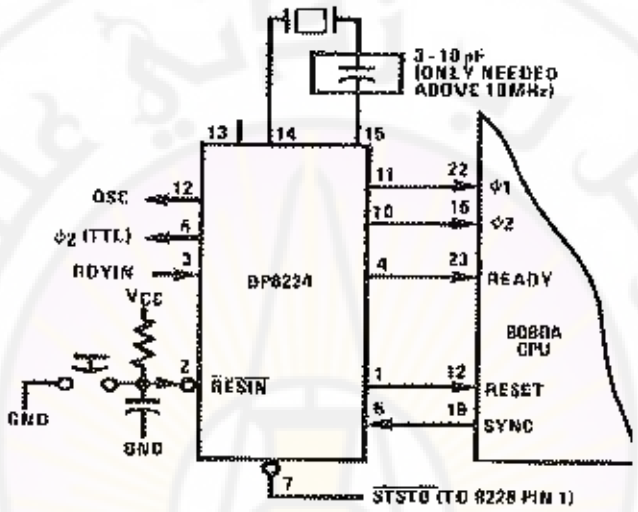
الأولى: الدارات غير المبرمجة وسنفرد لها بقية هذا الفصل لتعرّف البعض منها بالشكل الذي يسمح لنا بالاستخدام الأمثل لها.

الثانية: الدارات المبرمجة وسنخصص لها الفصل القادم لتعرّف التركيبة الداخلية للبعض منها وكيفية توصيل وبرمجة كل منها.

نتعرف فيما يلي بعض دارات الدعم البسيطة غير المبرمجة للمعالج 8080A، علما أن البعض منها ضروري جدا لعمل المعالج ومن هذه الدارات نذكر:

٥-١ الدارة 8224

تمثل هذه الدارة مولد ساعة (Clock Generator and Driver for 8080A) لتوليد الفازين ∅2 ،∅1 للمعالج وهي مثال لدارات الدعم الضرورية لعمل المعالج، حيث لا يمكن استخدام المعالج 8080A بدون هذه الدارة، كما أن الدارة 8224 لا تستخدم إلا مع المعالج 8080A. يظهر الشكل 2-5 الشكل القياسي لربط هذه الدارة مع 8080A.

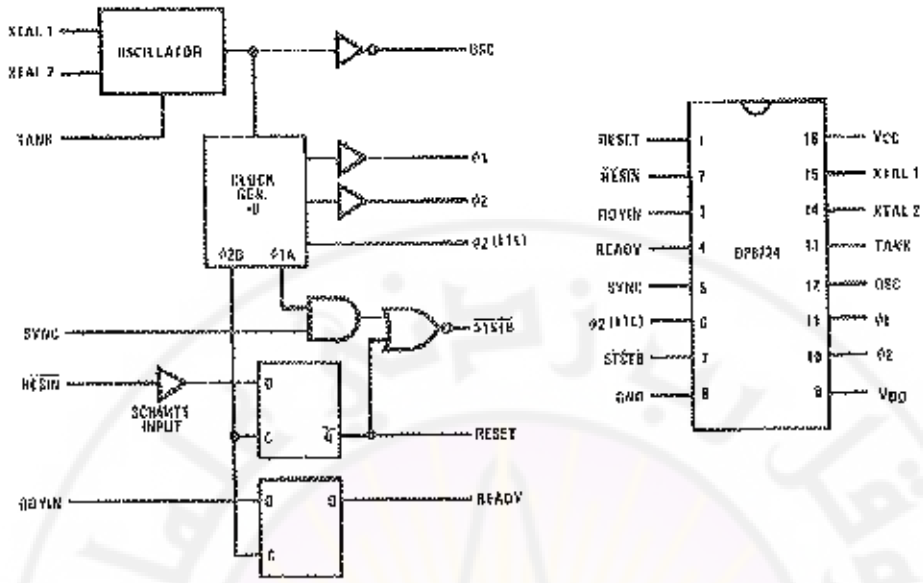


شكل2-5: الشكل العام لتوصيل الدارة 8224 مع المعالج الصغري 8080A

ويظهر الشكل3-5 التركيبة الداخلية لهذه الدارة مع توزع الإشارات على مأخذها. تحتوي الدارة 8224 على مقوٍ أو محرض إشارة يؤمن السوية الفولطية غير القياسية لهاتين الإشارتين كما يتطلبها المعالج أي: 0.8V ÷ 1- من أجل Log 0 و13V÷ 9 من أجل Log1، لذلك تتطلب هذه الدارة جهدي تغذية هما :

5V+ لعمل معظم الدارات الداخلية ضمن الدارة 8224 وتسمى V_{cc}

و12V+ لمحرض النبضات الساعية غير المتوافقة مع TTL وتسمى V_{dd} .



شكل3-5: المخطط الصندوقي وتوزع الإشارات للدارة 8224

تصنع الدارة 8224 بالتغليف DIL-16 ويلزم لتشغيلها إدخال إشارة الخرج SYNC من المعالج لتأمين التزامن. الدارة 8224 مدخلان $XTAL_1$ و $XTAL_2$ لوصل ثنائي قطب خارجي Crystal (يستخدم على الأغلب كريستال 18 MHz) يحدد تواتر الساعة اللازم لعمل المعالج. تقوم الدارة 8224 بتقسيم قيمة تواتر الكريستال على 9 ليتشكل الفازين Ø1, Ø2 ويمكن عوضا عن الكريستال استخدام عنصر RC أو LC ، حيث إن القيمة العظمى للتواتر المسموح به هو 27 MHz ليعمل المعالج فعليا بتواتر 3 MHz على الأكثر. يخرج من الدارة 8224 المخرج OSC بنفس قيمة التواتر الموضوع على الدخل ليستخدم من قبل أي عنصر آخر يلزمه Clock في المنظومة. تعطي الدارة 8224 أيضاً بالإضافة لـ Ø1, Ø2 الخرج $Ø_{TTL}$ المتوافق مع TTL لاستخدامه مع أي عنصر يحتاج لساعة متوافقة مع TTL ضمن التصميم. بوجود الدارة 8224 مع المعالج 8080A يتم وصل إشارتي Reset و Ready اللازمتين لعمل المعالج عبر هذه الدارة.

٥-٢ الدارة 8228/8238

عند دراسة دارتان معاً مثل الدارتان 8228 و 8238 بوجود الإشارة " /" بينهما، يعني أن الدارتين متماثلتان بكل شيء باستثناء خلاف بسيط يمكن تجاوزه أثناء الاستخدام، وبالتالي الحديث عن إحداها ينطبق على الأخرى. تمثل الدارة 8228 (المسماة: منظم أو متحكم النظام وقائد الممر System Controller and Bus Driver) محرض أو مقوي إشارة مركب لكل من ممر النظام وكلمة الحالة (التحكم) للمعالج 8080A. تصنع الدارة 8228 وفق التغليف DIL-28. يظهر الشكل 4-5 التركيبة العامة لها وتوزع إشاراتها، حيث إنها تحتوي على جزئيين رئيسيين هما:

a – محرض ممر بعرض 8 bit وبالاتجاهين بين المعالج وممر المعطيات للنظام.

b -- Decoder يرسل على ممر المعطيات كلمة الحالة التي تضم مجموعة من الإشارات التعريفية بحالة المعالج.

تتولد إشارات الحالة للمعالج دوما بالزمن T_1 من كل دورة آلة للمعالج 8080A وفي الأزمنة المتبقية من دورة الآلة تكون البيانات على الممر معطيات Data. نعلم أن للمعالج 8080A إشارتي تحكم للقراءة DBIN والكتابة 'WR ولكن هاتين الإشارتين لا تحددان من أين القراءة وإلى أين الكتابة.

تتولد إشارات التحكم بالممر (إشارات الحالة للمعالج) عن طريق مفكك ترميز داخلي بالتركيبة 3x8 Decoder بالمداخل DBIN, WR' and HLDA. إشارات الحالة للمعالج تحدد ذلك وفق إشارات الخرج لمفكك ترميز الدارة 8224 وفق التالي:

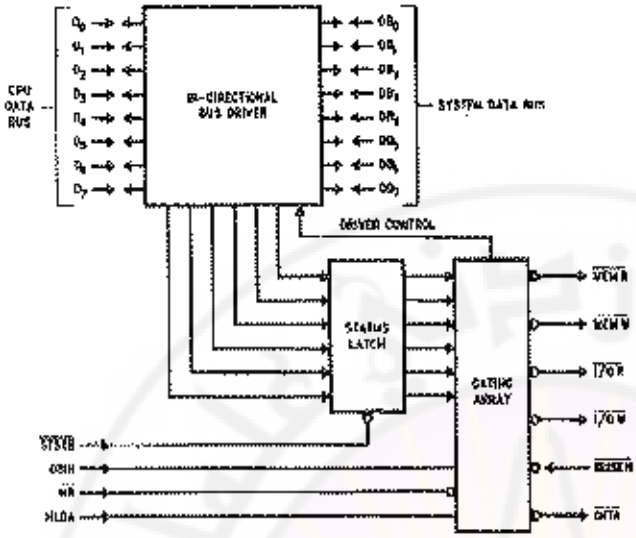
MEMR اختصارا 'MR من Memory Read وتعني القراءة من الذاكرة.

MEWR اختصارا 'MW من Memory Write وتعني الكتابة في الذاكرة.

'IOR من Input/output Read وتعني القراءة من بوابات الدخل.

'IOW من Input/output Write وتعني الكتابة على بوابات الخرج.

يمكن استخدام الدارة 8228 كدارئ أو مقوي ممر مع معالج آخر على عكس الدارة 8224 التي لا تستخدم إلا مع المعالج 8080A.



Black Diagram

D7-D0	Data Bus (8080 Side)
DB7-DB0	Data Bus (System Side)
I/OR'	I/O Read
I/OW'	I/O Write
MEMR'	Memory Read
MEMW'	Memory Write
DBIN	DBIN (From 8080)

Pin Configuration

INTA	Interrupt Acknowledge
HLDA	HLDA (From 8080)
WR'	WR' (From 8080)
BUSEN'	Bus Enable Input
STSTB'	Status Strobe (8224)
Vcc	5 V
GND	0 Volts

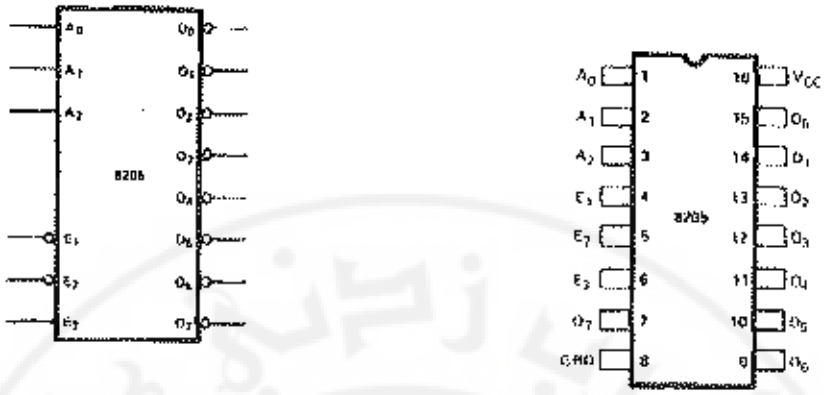
شكل4-5: المخطط الصندوقي وتوزع الإشارات للدارة 8228/8238

٥-٣ الدارة 8205

تصنع الدارة 8205 بالتقانة MOS، وتمثل مفكك ترميز (3x8 Decoder)، وقد تلزم بكل تصميم، وهي تقابل من العائلة TTL الدارة ذات الرقم 74138، ومن دارات السلسلة Intel3000 الدارة 3205. تصنع الدارة بالتقانات الثلاثة بالتغليف DIL-16 ولها نفس توزع الإشارات بالدارات الثلاثة.

يظهر الشكل5-5 الرمز المنطقي للدارة وتوزع إشاراتها، حيث إن للدارة ثلاثة مداخل لتحديد العنوان، وثلاثة مداخل للتحكم بعمل الدارة هي: E1', E2', E3. تكون كافة مخارج الدارة عند تفعيلها بحالة الواحد باستثناء مخرج واحد يكون بحالة الصفر (يسمى CS')

يقابل العنوان الموجود على مداخل الدارة $A_2A_1A_0$

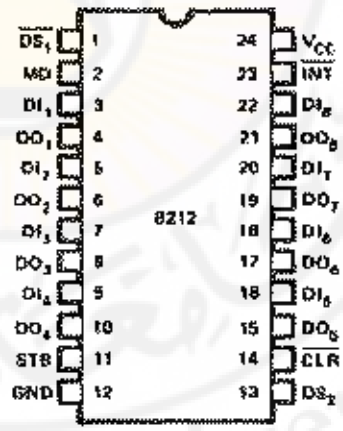


شكل5-5: توزع إشارات الدارة 8205

٥-٤ الدارة 8212

تمثل الدارة 8212 (8 bit Input/output Port) إحدى دارات الدعم العامة التي يمكن استخدامها مع المعالج 8080A وغيره من المعالجات الأخرى، لأنها تمثل مسجل (latch) بعرض 8bit. تحتوي الدارة على بعض مقويات الإشارة العازلة وتصنع بالتغليف DIL-24. يظهر الشكل 6-5 توزع الإشارات على مأخذ الدارة.

DI_1-DI_8	DATA IN
DO_1-DO_8	DATA OUT
$\overline{DS_1}$, DS_2	DEVICE SELECT
MD	MODE
STB	STROBE
INT	INTERRUPT (ACTIVE LOW)
CLR	CLEAR (ACTIVE LOW)



شكل6-5: توزع إشارات الدارة 8212

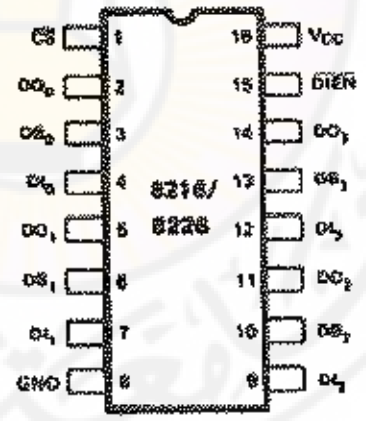
للدارة 16 خط لنقل المعطيات (بمعدل 8 خطوط للدخل و 8 خطوط للخرج). وتعمل بنمطين مختلفين، وتحتوي بالإضافة إلى إشارة الانتقاء 'CS على إشارة 'CLR لمسح

محتوى الدارة، وإشارتين لانتقاء التجهيزات Device Select إحداهما 'DS1 فعّالة بالصفر على عكس الإشارة الثانية DS2 الفعّالة دوما بالواحد. يمكن التحكم باتجاه نقل المعطيات عبر هذه الدارة بالقيم التي تطبق على كل من 'DS1 و DS2. تتميز هذه الدارة أيضاً بوجود إشارة Strobe اختصارا STB للدلالة على المعطيات فيما إن كانت هي المعطيات المطلوبة أم لا. بمعنى إذا كانت STB = 1 يعني ذلك أن البيانات على ممر المعطيات هي صحيحة وهي المعطيات التي يطلب تقويتها بوساطة الدارة. وفي الحالة المعاكسة البيانات الموجودة على الممر غير مفيدة وتهمل.

٥-٥ الدارة 8216/8226

تصنع هذه الدارة بالتغليف DiL- 16 وتمثل محرض أو مقوي ممر بعرض 4 bit (4-bit Bidirectional Buffer/Latch). الدارتين 8216 و 8226 متماثلتين مع فارق أن الدارة 8216 تحتوي على ثماني مقويات غير عاكسة (أربعة للدخل وأربعة للخرج)، بينما الدارة 8226 تحوي على ثمانية عواكس. يظهر الشكل 7-5 توزع إشارات الدارة ومفهوم كل منها.

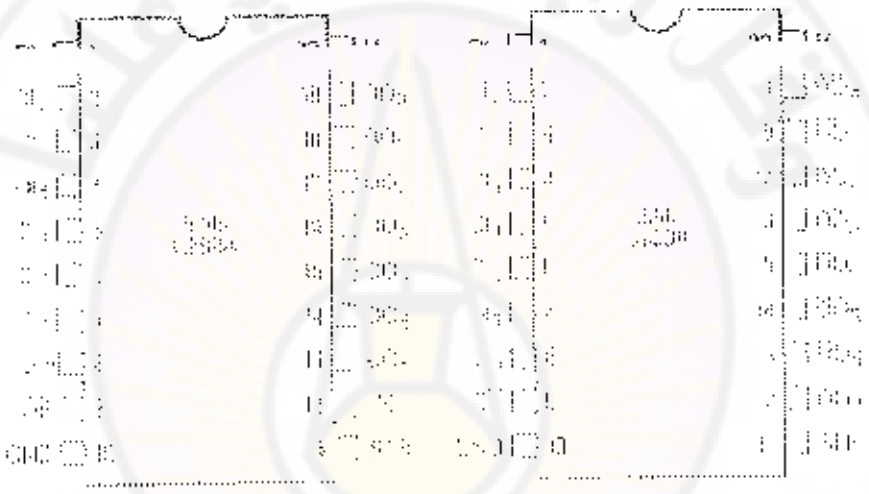
DB_0-DB_3	DATA BUS BI-DIRECTIONAL
DI_0-DI_3	DATA INPUT
DO_0-DO_3	DATA OUTPUT
DIEN	DATA IN ENABLE DIRECTION CONTROL
CS	CHIP SELECT



شكل7-5: توزع إشارات الدارة 8216/8226

٥-٦ الدارة 8282/8283

تعدّ الدارة 8282 (Octal Latch) واحدة من دارات الدعم واسعة الاستخدام في التطبيقات الحاسوبية وهي تصنع بالتغليف DIL-20 وتمثل مقوي إشارة بعرض bit 8. تعمل هذه الدارة بشكل مشابه لعمل الدارة 8216 بفارق أنها تنقل معطيات بعرضbit 8. يظهر الشكل 8-5 توزع إشارات هذه الدارة التي تحتوي على 8 خطوط معطيات للدخل و 8 خطوط معطيات للخرج. للدارة أيضاً إشارتي تأهيل هما STB المعروفة وإشارة 'OE التي يمكن وصلها بإشارة 'CS من خرج مفكك الترميز ليتم استخدامها عند الحاجة.



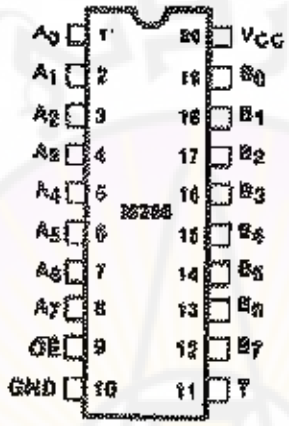
شكل8-5: توزع إشارات الدارة 8282/8283

تستخدم الدارة 8282 عادة مقوياً للممر أياً كان نوعها (ممرات المعطيات أو العناوين أو التحكم) وفيها الخطوط المخصصة للدخل منفصلة عن الخطوط المخصصة للخرج. ويوجد العديد من الدارات المماثلة لهذه الدارة التي تستخدم لتقوية الإشارات المطبقة على الدخل وتختلف عن بعضها بالإشارات التحكمية.

تختلف الدارة 8283 عن الدارة 8282 في أن خرج الدارة 8282 يكون غير معكوس، بينما خرج الدارة 8283 يكون معكوساً (المتمم الأحادي لقيمة الدخل).

٥-٧ الدارة 8286/8287

تسمى الدارة 8286 بـ Octal Bus Transceiver وتعدّ أيضاً واحدة من الدارات واسعة الاستخدام وهي تمثل سواقة عزل Buffer Driver لممر المعطيات، أي دارة إرسال واستقبال للمعطيات بعرض 8 bit بالاتجاهين. تصنع الدارة بالتغليف DIL-20 ويظهر الشكل 9-5 توزع إشاراتها.



الشكل9-5: توزع إشارات الدارة 8286

للدارة ثمانية مآخذ $A_0 - A_7$ تمثل Local Bus Data

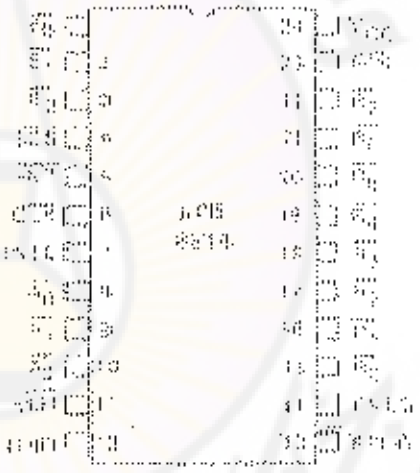
وثمانية مآخذ $B_0 - B_7$ تمثل System Bus Data

وإشارتي تحكم هما:

Output Enable – OE' وهي إشارة دخل ويمكن استخدامها كـ CS' للدارة، وإشارة Transmit – T وهي إشارة دخل قيادية تستخدم للتحكم باتجاه عمل الدارة، بحيث إنه عندما تكون T = 1، تكون المآخذ $A_0 - A_7$ دخل والمآخذ $B_0 - B_7$ خرج وعندما تكون T = 0 تصبح المآخذ $B_0 - B_7$ دخل والمآخذ $A_0 - A_7$ خرج. الفرق بين الدارتين 8286 و 8287 هو أن الدارة 8286 تمرر ما هو على الدخل إلى الخرج بشكل مقوى دون أن تعكس القيمة، أما الدارة 8287 فهي تعكس قيمة الدخل.

٥-٨ الدارة 8214

تمثل الدارة 8214 منظم للمقاطعة (Interrupt Controller) بثمانية مداخل ومخرج واحد. نعلم بأن المعالج 8080A يملك مأخذ وحيد للمقاطعة INTR ، وبالتالي لا يمكن وصل أكثر من جهاز لإجراء المقاطعة الخارجية له وهذه هي إحدى نقاط ضعف المعالج 8080A. ولتجاوز ذلك قامت شركة إنتل بطرح الدارة 8214 مع المعالج كدارة دعم تسمح بوصل ثمانية أجهزة بأن واحد على مدخل المقاطعة INT لإجراء المقاطعة باستخدام التعليمة RSTn، حيث n رقم صحيح من 0 وحتى 7 وكل منها يقابل عنوان محدد ابتداء من 0 وحتى 56 بفارق 8 بين كل عنوان والعنوان الذي يليه، بمعنى أن RST 5 مثلا تقابل العنوان 5x8 = 40H. تصنع الدارة بالتغليف DIL-24، ويظهر الشكل 10-5 توزع إشاراتها.



تحتوي الدارة 8214 على مرمز أفضلية بعرض 3 bit ومقارن Comparator وذاكرة . إذا طلبت محيطية ما مقاطعة وكان المعالج قادر على استقبالها (بحسب حالة INTE) فإن الدارة 8214 تولد الإشارة لمدخل INT للمعالج وتنقل رقم المقاطعة n من التعليمة RST n للمدخل المقابل لـ Rn إلى الكود الثنائي.

شكل10-5: توزع إشارات الدارة 8214

إذا أتى بنفس الوقت على مدخلين أو أكثر من المداخلR7 ÷ R0 مطلب مقاطعة فإن الدارة تأخذ بعين الاعتبار المطلب بالشعاع الأعلى لـ n باستخدام الذاكرة التي تحوي عنوان آخر شعاع، وتستخدم المقارن لتحدد أي من المداخل سيوصل بمدخل المعالج INT حيث إن الأولوية Priority تصاعدية (للرقم الأعلى).

٥-٩ أمثلة حول تصميم أنظمة مايكرو كومبيوتر باستخدام المعالج الصغري 8080A وبعض دارات الدعم البسيطة (غير المبرمجة)

مثال1-5: صمم منظومة مايكرو كومبيوتر متكاملة تضم بالإضافة للمعالج 8080A ذاكرة للقراءة فقط من نوع EPROM بسعة 4 KB وذاكرة للقراءة والكتابة RAM بسعة 4 KB مع بوابتي دخل وبوابة خرج واحدة، كل منها بعرض 8bit.

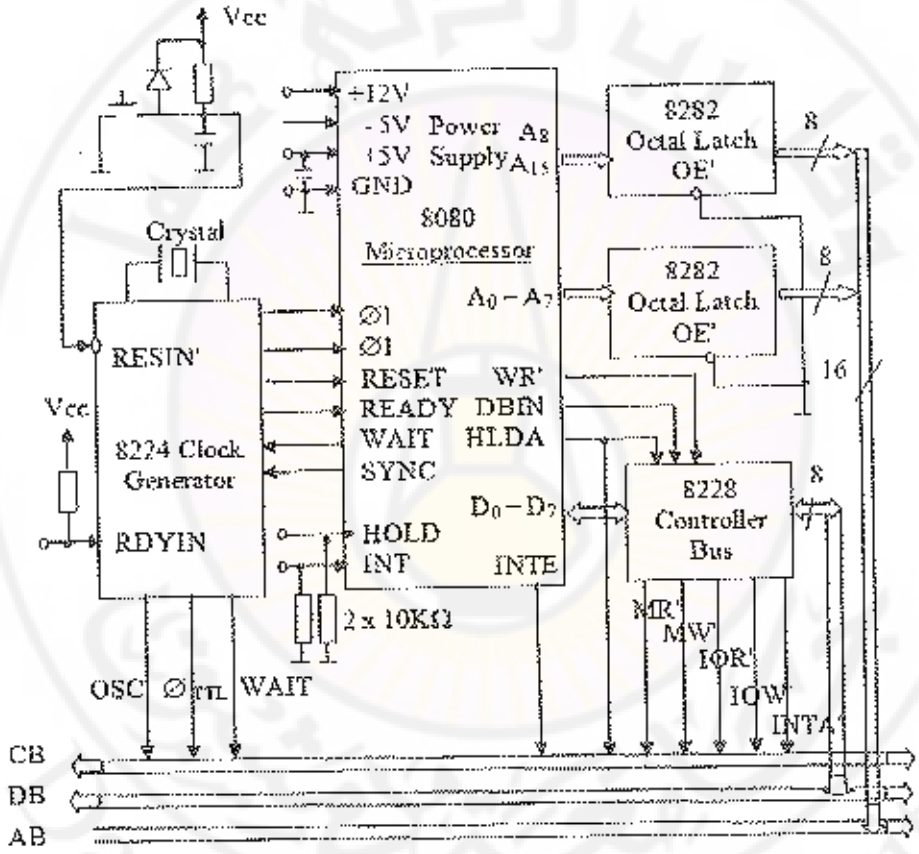
الجواب: لإجراء عملية التصميم لأي منظومة حاسوبية صغرية ننطلق بعد فهم التطبيق مما هو معطى أو متوفر من العناصر (الدارات المتكاملة) لتحديد ما يلزم التصميم بالحد الأدنى من الدارات لتأمين درجة معينة من الوثوقية والتكلفة والتوصل قدر الإمكان إلى تصميم بسيط وواضح خلال تتبعه من قبل أي طرف آخر.

وبحسب معطيات المثال، بما أن المعالج المحدّد للتصميم هو المعالج 8080A (لأننا لم نتعرف بعد إلا على هذا المعالج) نلتفت لتحديد بقية المتطلبات وفق التالي: بالنسبة لذاكرة EPROM بسعة 4 KB، سنستخدم بهذا المثال الدارة 2716 بالتنظيم (2 Kx 8 bit) عدد 2 ، للتوفيق مع أنواع أخرى من الدارات (ويمكن استخدام أنواع أخرى). وبالنسبة لذاكرة RAM بسعة 4 KB، قد يكون من المناسب استخدام RAM من النوع الاستاتيكي (تجنبا لتصميم دارات الإنعاش للنوع الديناميكي) مثل الدارة 6116 بالتنظيم (2 Kx 8 bit) عدد 2.

أما بخصوص البوابات بغض النظر عن عددها ونوعها إن كانت من نوع دخل أو خرج يمكننا انتقاء أي من دارات الدعم التي سبق وأشرنا إليها مثل: 8286, 8282, 8212 حيث كل منها يصلح لان تكون بوابة دخل أو خرج بعرض 8 bit، وتبدأ عملية التصميم بتوصيل دارة المعالج لتأمين الممرات الثلاثة للمنظومة، أي AB, DB, CB ويجب التركيز خلال ذلك على تأمين السوية الفولطية المناسبة لكل من إشارات الدخل إلى القيمة الصحيحة. أما بالنسبة لإشارات خرج المعالج، ليس الأمر بنفس الأهمية إذ يكفي تحديد إشارات الخرج التي تلزم في التطبيق.

ويأتي بالقسم الثاني من التصميم توصيل بقية الدارات اللازمة من ذواكر بنوعيها والبوابات بنوعيها بشكل قياسي.

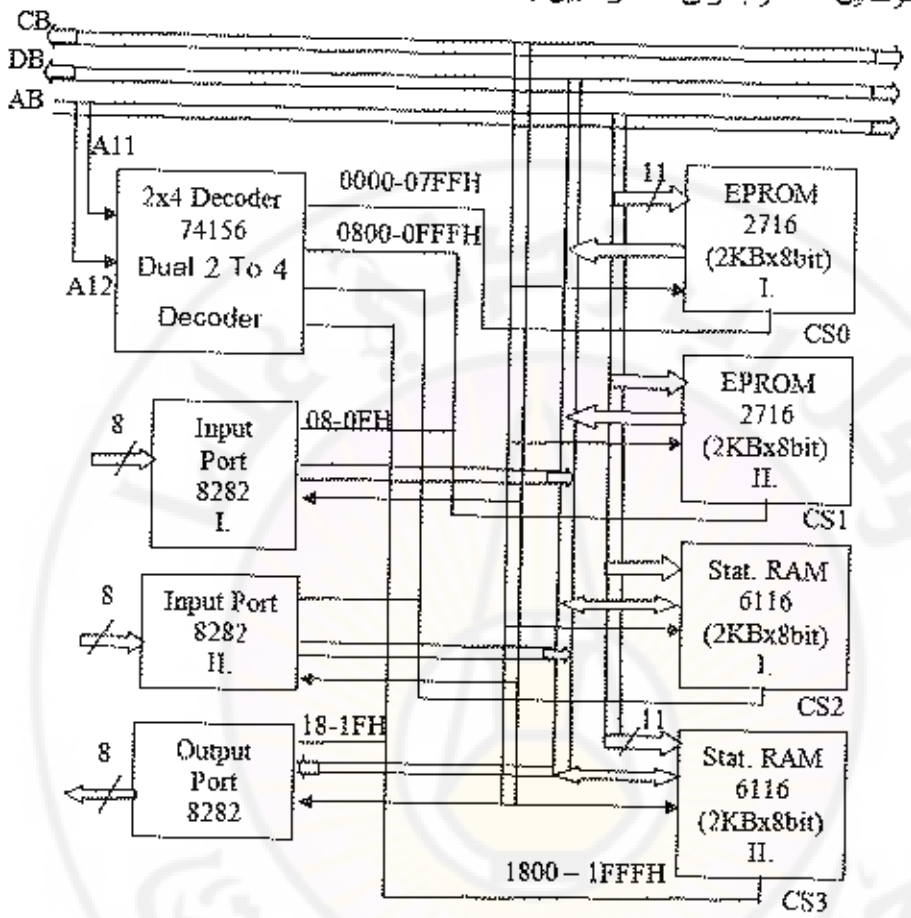
أولا: توصيل دارة المعالج 8080A- بما أن المعالج الذي سيبنى عليه الحاسوب الصغري (المايكرو كومبيوتر) هو المعالج 8080A يلزم معه بغض النظر عما هو مطلوب من ذواكر وبوابات دارتي الدعم 8224 و 8228 لتوصيلها مع المعالج وفق ما هو مبين بالشكل 11-5.



الشكل 11-5: توصيل المعالج 8080A لتشكيل الممرات AB, DB, CB

تم في الشكل أعلاه رسم ممر العناوين بعرض 16 bit وممر المعطيات بعرض 8 bit. أما بالنسبة لممر التحكم فإن عدد الخطوط متغير بحسب الطلب.

ولتوصيل ما تبقى من ذواكر وبوابات ننطلق من الشكل النهائي لممرات المعالج الثلاثة وتوصيل المطلوب وفق ما هو مبين بالشكل 12-5.



الشكل12-5: توصيل 4KB ذاكرة EPROM و 4 KB ذاكرة RAM وبوابتي دخل وبوابة خرج مع المعالج 8080A لتشكيل منظومة حاسوب صغري

بالطبع يجب أثناء التصميم الفعلي وضع الكاتالوكات الخاصة بكل دارة تستخدم لتحديد أرقام الـ Pin كما هي معطاة من الشركة الصانعة للدارة والتقيد بذلك أثناء التوصيل. يظهر الشكل 12-5 توصيل كل من الذواكر بنوعيها (EPROM and RAM) والبوابات بنوعيها (دخل وخرج). وبالتالي فان الشكلين 11-5 و 12-5 يشكلان معا كامل التصميم المطلوب وفق المثال 1-5.

شرح التصميم: بداية نشير إلى بعض الملاحظات العامة بخصوص هذا المثال التي تصلح لغيره من أمثلة التصميم ومنها:

١- يمكن أن يتغير الشكل العام لكل تصميم بحسب نوع الدارات المستخدمة ومواصفاتها، حيث إن كل كلمة في نص المثال قد يقابلها شكل معين للتصميم، لكن المبدأ العام في التصميم هو نفسه ويتلخص بعد تحديد العناصر التي ستستخدم في التصميم تحديد إشارات الانتقاء Chip Select (CS) لكل دارة مستخدمة.

٢- قد يتم اختيار بعض دارات الدعم التي تقابل المطلوب، مع الأخذ بعين الاعتبار أنه في كثير من الأمثلة يمكن استخدام أنواع أخرى من الدارات التي تحقق الغرض.

٣- سنحاول في كل تصميم الإشارة لعرض الممر بخط مائل (/) مع الرقم المقابل لعدد هذه الخطوط بكل ممر.

٤- سنركز بهذا التصميم وغيره من التصاميم التي سنوردها لاحقا على توصيل الإشارات الأساسية لكل دارة (أو ما يكافئها من حيث المفهوم).

٥- سوف لن نتوقف عند السوية الفولطية لتفعيل بعض الإشارات بالنسبة لبعض الدارات إن كان Low أو High ، حيث يمكن الانتقال من أي مستوى لآخر عند معرفته بشكل فعلي باستخدام بوابة NOT.

٦- يمكن أن نزيد من قيود تفعيل دارة ما باستخدام CS مع تفعيل إشارات أخرى معها شرطاً لعملها باستخدام البوابة المنطقية المناسبة. وبخصوص المثال المعطى نطبق ما سبق وفق التالي:

بما أن سعة كل Chip من ذواكر EPROM المستخدمة في التصميم هو 2 KB فقط، معنى ذلك أن كل Chip يلزمه 11 خط للعنونة هي $A_0 - A_{10}$. ونفس الشيء بالنسبة لذواكر RAM باعتبار أن سعة كل Chip هو أيضاً 2 KB.

يوصل ممر المعطيات بعرض 8 خطوط على كل دارة من الـ Chips الأربعة لان تنظيم الذواكر المستخدمة هو قياسي (حرف بطول 8 bit بكل موضع ذاكرة)، لكن اتجاه ممر المعطيات بالنسبة لذواكر EPROM هو دوما من الذاكرة إلى المعالج، أما بالنسبة لذواكر

RAM فيكون بالاتجاهين لان المعالج يقرأ من الذاكرة معطيات وتعليمات ويكتب بها المعطيات. وبالنسبة للبوابات فإن عرض ممر المعطيات هو 8 bit وفق ما هو مطلوب، وقد تم اختيار الدارة 8282 التي تحقق ذلك.

ولتحديد مفكك الترميز المناسب الذي يحدد العنوان المقابل لكل Chip مستخدم، ننطلق من خطوط العنونة اللازمة لكل دارة على حدة. وبما أنه يلزم كل من الذواكر المستخدمة في التصميم خطوط عنونة من A_0 وحتى A_{10} لعنونة المواقع ضمن كل Chip، يمكن استخدام خطوط العنونة التي تلي ذلك للتمييز بين الدارات نفسها، لتكون المدخل المناسب لمفكك الترميز ولتحدد وزن كل خرج لمفكك الترميز. فمثلا بالمثال المعطى لدينا أربع دارات أي يلزم أربعة عناوين مختلفة، لذلك قد يكفي بهذا المثال مفكك ترميز 2x4 Decoder مثل الدارة المتكاملة 74156 التي تحتوي بدورها على مفككي ترميز بنفس الدارة (يستخدم إحداها فقط) ويتم وصل الخطين A_{11} و A_{12} كمدخل لمفكك الترميز المنتقى ويكون وزن كل خرج لمفكك الترميز 8/4 = 2 KB وهذا يعني أن كل خرج يكفي لعنونة ذاكرة من الذواكر الأربعة المستخدمة، ونعبر عن ذلك بتخصيص CS مختلف لكل من هذه الدارات. وبذلك تكون خريطة الذاكرة بالمثال المعطى وفق ما هو مبين بالجدول 2-5.

الجدول2-5: مخطط توزيع ذواكر EPROM و RAM مع المعالج 8080A

A_{12} A_{11}	$A_{10}A_9A_8$	$A_7...A_4$	$A_3...A_0$	Address	For
0 0	0 0 0	0 ... 0	0 ... 0	0000H	EPROM I.
0 0	1 1 1	1 ... 1	1 ... 1	07FFH	(First 2 KB)
0 1	0 0 0	0 ... 0	0 ... 0	0800H	EPROM II.
0 1	1 1 1	1 ... 1	1 ... 1	0FFFH	(Second 2 KB)
1 0	0 0 0	0 ... 0	0 ... 0	1000H	RAM I.
1 0	1 1 1	1 ... 1	1 ... 1	07FFH	(Third 2 KB)
1 1	0 0 0	0 ... 0	0 ... 0	1800H	RAM II.
1 1	1 1 1	1 ... 1	1 ... 1	1FFFH	(Forth 2 KB)

انطلاقا من أن المعالج 8080A يبدأ عمله مع الذاكرة بالعنوان 0000H بعد كل عملية إقلاع أو RESET، حيث يفترض وجود كافة برامج التحكم والتعريف بالمكونات الأساسية للمنظومة. وبالتالي يفترض وجود ذاكرة من نوع غير متطاير كذاكرة EPROM على العنوان 0000H لحفظ الـ BIOS ، يليها وجود ما هو متوفر من ذواكر RAM. بمعنى أن أول إشارة انتقاء CS0 بوزن 2 KB تخصص للذاكرة رقم I من نوع EPROM يليها إشارة CS1 بوزن 2 KB للذاكرة رقم II من نوع EPROM وبعد أن ينتهي الحجم المطلوب من ذواكر EPROMs تبدأ ذواكر RAM بالخط CS2 بوزن 2 KB للذارة رقم I وبالخط CS3 بوزن 2 KB للذاكرة رقم II من نوع RAM وفق ما هو مبين بالشكل 2-5، حيث إنه قد كتب بالتصميم المجال العنوني لكل خط بجانبه.

أما بالنسبة لعنونة البوابات فالأمر يختلف قليلا، لان المعالج 8080A يستطيع عنونة 256 بوابة للدخل والخرج ولهذه الغاية يكفي المعالج 8 خطوط للعنونة. وبالطبع عندما يتعامل المعالج بلحظة ما مع الذاكرة لا يمكنه التعامل مع البوابات والعكس صحيح. لذلك عندما يتعامل المعالج مع البوابات يرسل العنوان المطلوب على الخطوط A_0 وحتى A_7 أو على الخطوط A_8 وحتى A_{15} وهذا يسمح باستخدام مفكك ترميز واحد في التصميم لكل من الذواكر والبوابات بدلا من استخدام مفكك ترميز منفصل لكل منهما. وتكون خارطة توزيع البوابات وفق ما هو مبين بالجدول 3-5.

لا يهم هنا ترتيب البوابات على خطوط الانتقاء كما هو الحال بالنسبة للذواكر إن كانت البوابة دخل أو خرج. ويمكننا تصور قيمة العنوان الموجودة على A_0 وحتى A_7 هي نفس القيمة الموجودة على خطوط العنونة A_8 وحتى A_{15}، أي أن مداخل مفكك الترميز بالنسبة للذواكر A_{11} و A_{12} تقابل بالنسبة للبوابات الخطين A_3 و A_4.

وبما أنه تم في التصميم أعلاه وصل أول بوابة دخل بخط الانتقاء الثاني CS1 فإن عنوان هذه البوابة يكون كل عنوان ضمن المجال 0FH – 08 كما في الجدول 3-5، وهكذا الحال لمعرفة عنوان أي بوابة أخرى.

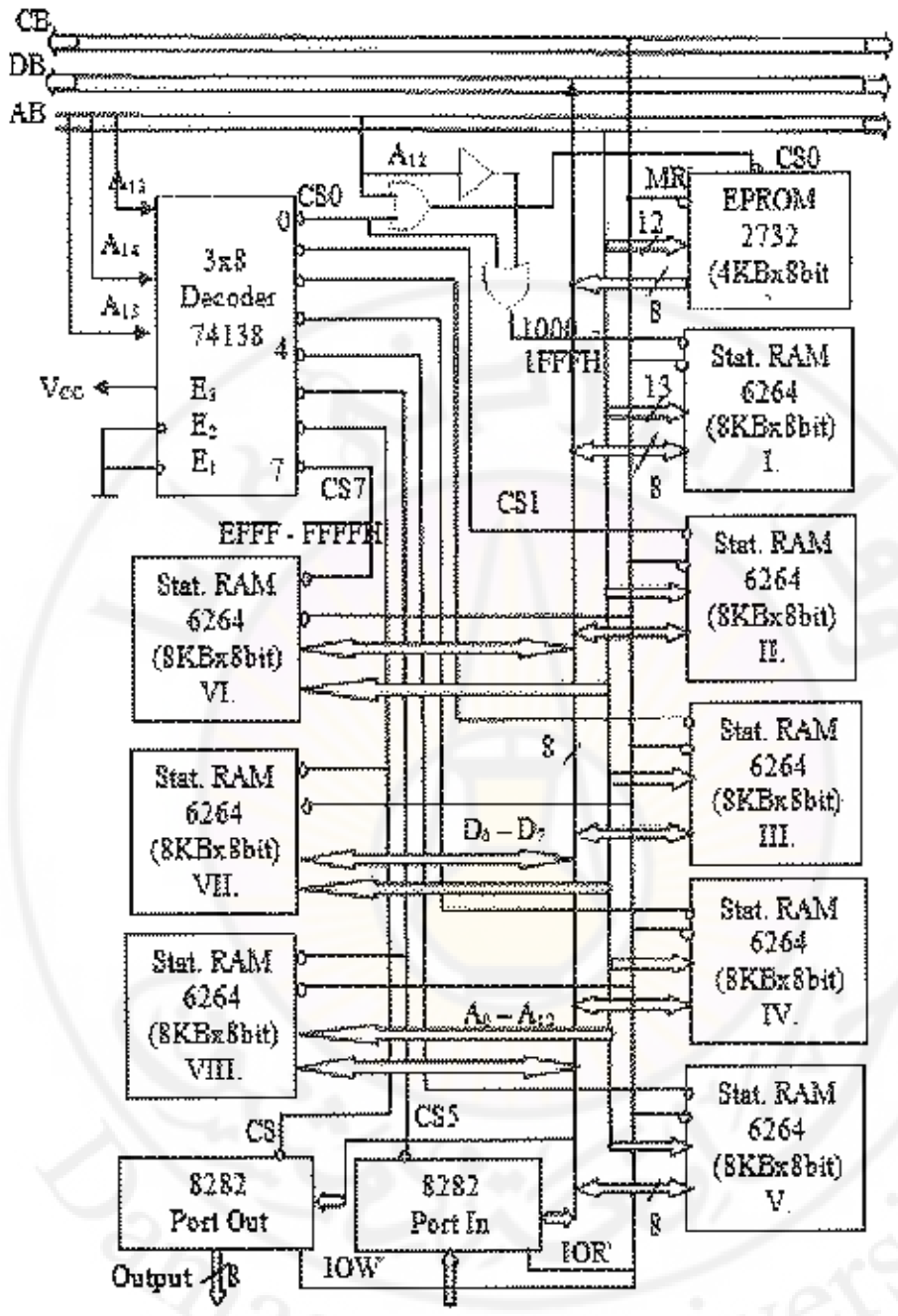
الجدول3-5: مخطط توزيع البوابات مع المعالج 8080A

$A_{15}A_{14}A_{13}A_{12}$ A_{11} $A_{10}A_9A_8$ A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	Address	For
d d d 0 0 0 0 0	00H	None
d d d 0 0 1 1 1	07H	used
d d d 0 1 0 0 0	08H	First
d d d 0 1 1 1 1	0FH	port In
d d d 1 0 0 0 0	10H	Second
d d d 1 0 1 1 1	17H	port In
d d d 1 1 0 0 0	18H	Port
d d d 1 1 1 1 1	1FH	Out

مثال2-5: صمم منظومة مايكرو كومبيوتر متكاملة باستخدام المعالج الصغري 8080A تضم بالإضافة للمعالج ذاكرة للقراءة فقط من نوع EPROM بسعة 4 KB وذاكرة للقراءة والكتابة من نوع RAM بسعة بقية المجال العنوني للمعالج، مع بوابة دخل واحدة تقع على العنوان A5H وبوابة خرج واحدة . ويطلب تحديد مجال العنوان لكل Chip في التصميم؟.

الجواب: بهذا المثال المماثل للمثال السابق، نوجز الشرح بأن توصيل دارة المعالج بالمثال السابق (شكل 5-11) لا يتغير ويمكن اعتماد نفس الشكل لهذا المثال أيضاً كالجزء الأول من التصميم. أما بخصوص الجزء الثاني من التصميم، ننطلق من المعطيات بنص المثال فنلاحظ وجود فارق حول المجال العنوني للمعالج 8080A وهو 64 KB. وبالتالي يلزم 60 KB ذاكرة من نوع RAM.

ويوجد فارق آخر يتمثل بتحديد العنوان الذي ستربط عليه إحدى البوابات (الخرج). يبين الشكل5-13 منظومة الحاسوب الصغري (المايكرو كومبيوتر) بالمعطيات المعطاة، حيث يوجد مشكلة تتعلق باختلاف حجوم الذواكر مابين EPROM و RAM بشكل يجعل وجود فراغ عنوني بين النوعين المختلفين من الذواكر وهذا ما سنحاول استدراكه في هذا المثال



الشكل13-5: الجزء الثاني من التصميم المطلوب بالمثال 2-5.

بضغط الفراغ لتتمكن من استخدام كامل المجال العنواني للمعالج بشكل فعلي. وانطلاقا مما هو مطلوب (بما أن حجم الذاكرة من نوع EPROM هو 4 KB) نختار الدارة 2732 بسعة 4 KB ولا داعي لاختيار دارة بسعة أكبر من ذلك.

وبما أن حجم الذاكرة المطلوب من نوع RAM هو 60 KB نحاول أن نختار دارة بأكبر سعة ممكنة ولتكن الدارة 6264 بسعة 8 KB لكل Chip (بالطبع كان بالإمكان اختيار ذاكرة RAM بنفس سعة الذاكرة EPROM وبهذه الحالة يلزم مفكك ترميز بـ 4x16 ويلزم 15 Chips من RAM مما ينعكس سلبا على درجة تعقيد التصميم). أي لنحاول بهذا المثال استخدام دارات مختلفة بالحجوم انطلاقا من أنه قد لا يكون متوفراً حجم الذاكرة الذي نرغب باستخدامه.

ولنحاول بهذا المثال إظهار آلية إلغاء الفراغ: عند استخدام ذاكرة مثل 6264 يلزم سبع دارات ونصف لتأمين الحجم المطلوب، إلا أنه لا يمكننا تقسيم الـ Chip الواحد وبالتالي سنستخدم 8 دارات على أن نستخدم من إحداها نصف الحجم فقط وبهذه الحالة سيبقى 4KB في إحدى الدارات موجود فيزيائيا وغير مستخدم منطقيا. وبما أن EPROM يجب أن تكون على العنوان0000H فإن دارة RAM التي سيبقى نصفها فارغا هي الدارة الأولى يلزم بهذا المثال استخدام مفكك ترميز 3x8 Decoder وسيكون وزن خرجه 8 KB لكل دارة باستثناء الخرج الأول وهو أيضاً بوزن 8 KB لكنه يقسم إلى خطين هما: الأول بوزن 4 KB يخصص لعنونة الذاكرة EPROM.

والثاني بوزن 4 KB لعنونة النصف الأول من أول دارة من دارات RAM بحيث يبقى النصف الثاني من هذه الدارة بسعة 4 KB موجودة فيزيائيا وغير مستخدمة فعليا. يبين الجدول 4-5 عن خارطة الذاكرة آلية عنونة كل دارة في التصميم.

تحتاج الدارة 6264 من A_0 وحتى A_{12} للعنونة ضمن الدارة. بقية خطوط ممر العناوين AB أي الخطوط A_{13}، A_{14}، A_{15} تلزم كمدخل لمفكك الترميز للتمييز بين الدارات.

الجدول4-5: عنونة 64 KB ذاكرة (بمعدل 4KB EPROM و 60KB RAM)

$A_{15}A_{14}$ $A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7 \ldots A_4$	$A_3 \ldots A_0$	Addr.	For
0 0 0 0	0 0 0 0	0 ... 0	0 ... 0	0000H	(4 KB)
0 0 0 0	1 1 1 1	1 ... 1	1 ... 1	0FFFH	EPROM
0 0 0 1	0 0 0 0	0 ... 0	0 ... 0	1000H	+(4 KB)
0 0 0 1	1 1 1 1	1 ... 1	1 ... 1	1FFFH	RAM I.
0 0 1 0	0 0 0 0	0 ... 0	0 ... 0	2000H	RAM II.
0 0 1 1	1 1 1 1	1 ... 1	1 ... 1	3FFFH	(8 KB)
.....	.....	.....	.....	.....	.....
1 1 1 0	0 0 0 0	0 ... 0	0 ... 0	EFFFH	RAM VII
1 1 1 1	1 1 1 1	1 ... 1	1 ... 1	FFFFH	(8 KB)

يتضح من الجدول أعلاه، أنه طالما A_{12} = 0 سيكون عمل المعالج بأول 4KB وهي تخصص للذاكرة EPROM ، وعندما يصبح A_{12} = 1 يتم الانتقال إلى ثاني 4 KB ويخصص للذاكرة RAM لذلك تم في التصميم إضافة منطق بسيط لتحقيق ذلك. وبهذا الشكل يتم تجاوز مشكلة الفراغ في المجال العنوني للمعالج، بمعنى أن كل عنوان للمعالج من 0000H وحتى FFFFH يقابل موضعاً محدداً في الذاكرة سواء كانت من نوع EPROM أو RAM.

أما بخصوص البوابات، بما أن عنوان بوابة الدخل محدد بشكل مسبق على العنوان A5H نعمل على تحقيق ذلك وفق التالي:

$A_{15}A_{14}A_{13}A_{12}$ $A_{11}A_{10}A_9A_8$ A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	Address	For
1 0 1 0 0 1 0 1	A5H	Port In

أي البوابة موصولة على CS_5. أما بوابة الخرج فيمكن وصلها على أي عنوان (CS_6).

٥-١٠ أسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

1- أي من العبارات التالية صحيحة؟

a- يستطيع المعالج 8080 العمل بمنظومة متعددة المعالجات

b- يملك المعالج 8080 مأخذ للنقل التسلسلي للمعطيات

c- يحتوي المعالج 8080 بداخلة على ذواكر وبوابات دخل/ خرج

d- لا شيئ مما مما سبق

2- تقسم الدارة 8224 قيمة الكريستال الخارجي للحصول على التردد اللازم لعمل المعالج 8080 على:

a -2 b -3 c -9 d- None of all

3- يحتاج المعالج 8080 لكي يعمل على:

a-One Clock phase b -Two Clock phases

c- Three Clock phases d-Four Clock phases

4- يعنون المعالج 8080 ذاكرة بسعة:

a -16K Byte b- 64K Byte

c -64M Byte d -16M Byte

5- يعمل المعالج 8080 بقيمة فولطية للساعة

a- -1V ÷ 0.8V للمستوى صفر 9V ÷ 13V للمستوى واحد

b- - 0.5 ÷ 0.6V للمستوى صفر 3.9 V ÷ Vcc + 1V للمستوى واحد

c- 0 V للمستوى صفر 5 V للمستوى واحد

d- TTL متوافقة مع قيم

6- جواب المطلب HOLD بالنسبة لمعالج صغري من INTEL هو

a -RESET b -HLDA

c -DMA d -None of all

الفصل السادس: دارات الدعم المبرمجة للمعالج 8080A

نتعرف بهذا الفصل بعض دارات الدعم المبرمجة التي يمكن استخدامها مع المعالج 8080A وغيره من المعالجات الأحدث.

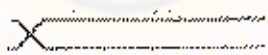
يقصد بالدارة المبرمجة كل دارة يمكن قيادة عملها عن طريق المعالج الصغري باستخدام بعض كلمات التحكم. قبل الشروع بتعرّف بعض هذه الدارات المبرمجة، نستعرض المفهوم العام لكل من النوعين الأساسين لنقل المعطيات وهما: النقل التفرعي (المتوازي) والنقل التسلسلي للمعطيات لنتمكن من تعريف الدارات المقابلة لكل نوع من أنواع النقل.

٦-١ النقل المتوازي للمعطيات

يقصد بالنقل المتوازي للمعطيات نقل كافة البيانات الموجودة بلحظة ما على ممر المعطيات دفعة واحدة. وهنا نميز بين ثلاثة أنواع من النقل المتوازي للمعطيات هي:

*- نقل بدخل وخرج بسيطين Simple I/O

ويستخدم هذا النوع من النقل عند نقل معطيات رقمية من أحد المفاتيح البسيطة إلى معالج صغري، وهذا النوع من النقل لا يحتاج إلى أية إشارات تحكم إضافية، إذ يكفي وصل المفاتيح إلى منفذ دخل Input Port ثم قراءة المنفذ. ويتم نفس الشيء عند إخراج معطيات إلى جهاز إظهار مثل ديود ضوئي LED وهذه الحالة يمكن تمثيلها على شكل موجة التوقيت التالية :



حيث إن الخطوط المتقاطعة بالموجة تمثل بايت المعطيات بصيغة صحيحة valid.

*- نقل بدخل وخرج بوساطة إشارة تمكين Simple Strobe I/O

توجد عادة المعطيات في العديد من التطبيقات على جهاز خارجي في وقت معين ويجب أن نقرأ عند ذلك الوقت كما هو الحال بالنسبة للوحة المفاتيح . فمثلا عند ضغط مفتاح ما من الـ Keyboard ترسل شفرة ASCII المقابلة للمفتاح المضغوط، وترسل جملة دارات لوحة المفاتيح إشارة تمكين Strobe على خط تحكمي منفصل لتشير بأن المعطيات الصحيحة موجودة الآن على ممر المعطيات DB. أي أن النقل بهذه الحالة يعتمد على

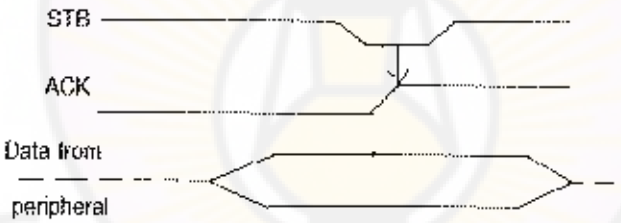
إشارة تحكم إضافية، وفق ما هو مبين بالشكل 1--6، بحيث إن عملية النقل لا تتم ما لم تفعل إشارة التحكم المرفقة.



الشكل1--6: دخل خرج ببضة تمكين بسيطة

*- نقل بدخل وخرج بإشارة مصافحة وحيدة Simple Handshaking I/O

تصلح إشارة التمكين للاستخدام في بعض الحالات البسيطة. لكن في بعض التطبيقات الخاصة يلزم إشارة ثانية تخبر المرسل بالوقت المناسب لإرسال بايت المعطيات التالي، ولذلك يستخدم خط آخر لنقل المعطيات بالمصافحة Handshaking. يبين الشكل 2--6 شكل موجة التوقيت من أجل نقل المعطيات بإشارة مصافحة من جهاز محيطي Peripheral إلى معالج صغري، حيث يظهر بالموجة إشارتي تحكم.



الشكل2--6: دخل/خرج بمصافحة وحيدة

الأولى STB: لتخبر منظومة الاستقبال (المعالج) بأن البيانات صحيحة وجاهزة والثانية ACK: جواب (Acknowledge) من منظومة الاستقبال لتشير بأنها جاهزة لاستقبال البيانات Data.

ويمكن تفسير المخطط بالتالي: يخرج الجهاز المحيطي بعض المعطيات المتوازية (بايت)، ويرسل معه إشارة تمكين S T B' إلى المعالج.

عندما يكتشف المعالج فعالية الإشارة 'S T B على أساس المقاطعة، يقوم بقراءة البايت. بعد ذلك يرسل المعالج إشارة اعتراف ACK إلى الجهاز المحيطي ليشير إلى أن المعالج أخذ البايت ويستطيع المرسل إرسال البايت التالي للمعطيات . يشار إلى هذه العملية من وجهة نظر المعالج على أنها دخل ممكّن Strobed In.

إن نفس أشكال الموجة قد تمثل خرجاً ممكناً (مصافحاً) من معالج إلى طابعة أيضاً. وبهذه الحالة يخرج المعالج محرفاً إلى الطابعة ويمكن لها ذلك بالإشارة 'STB ليخبرها " هنا محرف لك " وعندما تكون الطابعة جاهزة فإنها تجاوب بالإشارة ACK لتخبر المعالج " لقد استلمت ذلك الحرف ، أرسل إلي محرفاً آخر" . أي أن المنظومة لا تستطيع إرسال بايت المعطيات التالي حتى يشير الجهاز المستقبل بإشارة ACK إلى أنه جاهز لاستقبال البايت التالي .

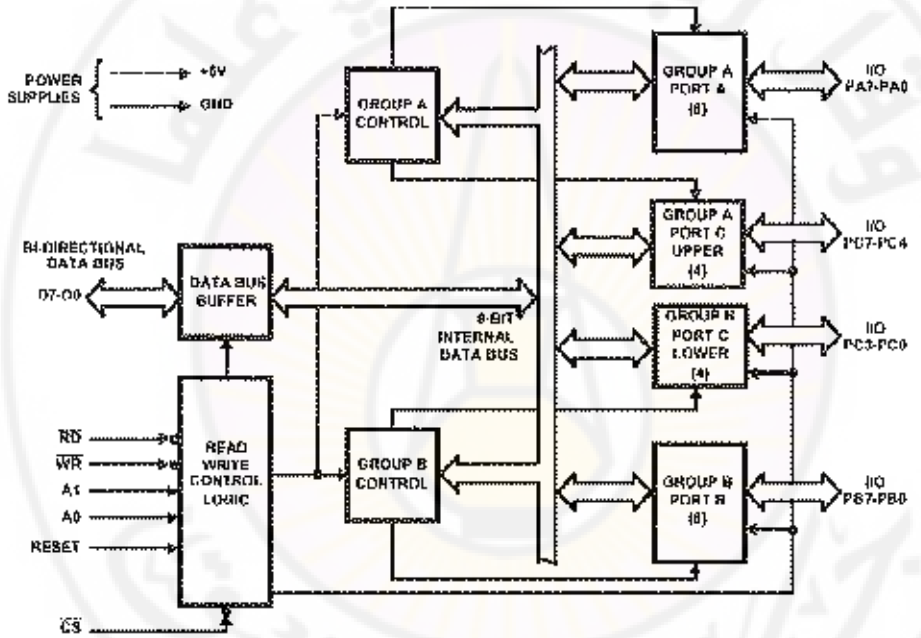
بالإضافة لأنواع النقل المبينة أعلاه يوجد أيضاً " نقل المعطيات بمصافحة مزدوجة Double Handshaking Data Transfer " وهي تماثل المحادثة التي تجري بين شخصين، حيث يخفض الجهاز المرسل خطة 'S T B (أي يفعّله) ليقول لدي معطيات صحيحة ويسأل " هل أنت جاهز " ترفع المنظومة المستقبلة في حال جهوزيتها، خطها ACK (أي تفعّله) لتقول " أنا جاهزة " .

بعد ذلك يبدأ الجهاز المحيطي بإرسال بايت المعطيات ويرفع خطه 'S T B ليقول " هنا بعض المعطيات الصحيحة لك خذها" . وبعد أن تنتهي المنظومة المستقبلة من قراءة المعطيات ، تعيد خطها ACK إلى الحالة المنخفضة وتقول " حصلت على المعطيات ، وأنا انتظر طلبك لإرسال البايت التالي" . يمثل ذلك في موجات التوقيت بحيث كل حافة لكل إشارة لها معنى خاص.

صممت شركة إنتل دارة دعم مبرمجة بالاسم 8255A لتحاكي مختلف طرق نقل المعطيات المتوازية المشار إليها أعلاه.

٦-١-١ دارة الدعم المبرمجة 8255A للنقل التفرعي للمعطيات

تصنع الدارة 8255 (والتي تسمى Programmable Peripheral Interface) بالتقانة NMOS وتوجد بالأشكال الثلاثة: 8255 A2 , 8255A , 8255. تعد الدارة 8255 أي كان شكلها من دارات الدعم المبرمجة للمعالج 8080A وتصلح للعمل مع غيره من المعالجات الأحدث كما سنجد لاحقاً كبوابات دخل/خرج بنقل بسيط أو بالمصافحة. توجد الدارة 8255 بالتغليف DIL-40 وهي تضم ما يقارب 1500 ترانزستور. يظهر الشكل3-6 البنية الداخلية للدارة 8255 وهي تعمل بتغذية 5V+.



شكل3-6: التركيبة الداخلية للدارة القابلة للمبرمجة 8255

تصنف الدارة 8255 من دارات الـ VLSI ورغم تعقيداتها لا تصلح لإجراء عمليات حسابية أو منطقية مثل المعالج، بل هي ومثيلاتها من دارات الدعم تعدّ من الدارات المركبة والمهمة للتعامل مع المعالج كتابع . تستخدم الدارة 8255 A للنقل المتوازي للمعطيات وهي تضم وفق ما هو مبين من تركيبتها الداخلية 24 خط للدخل أو للخرج موزعة على ثلاث بوابات A, B, C كل منها بعرض 8 Bits .

تحتوي الدارة 8255 A بالإضافة إلى البوابات الثلاثة على مسجل داخلي للتحكم بعمل الدارة ويسمى CWR – Control Word Register وهو يستخدم لوضع كلمة الحالة للدارة أي لبرمجتها، بالتالي يلزم الدارة أربعة عناوين مختلفة لعنونة البوابات الثلاثة والمسجل CWR. خصص لتشكيل هذه العناوين الأربعة خطي العنونة A_0 و A_1 لتحديد البوابة (المنفذ) التي يتم التعامل معها من الدارة وفق التالي:

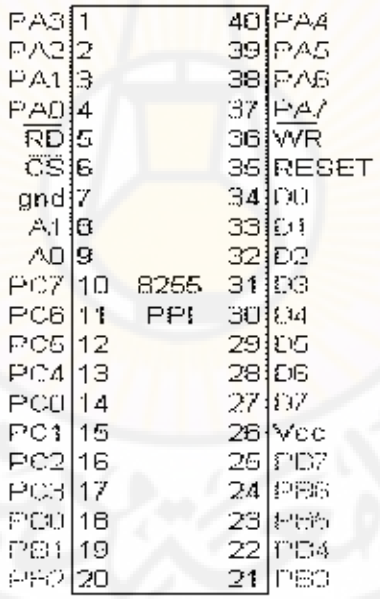
A_1 A_0 = 00 تقابل البوابة PA (PA_7 ..... PA_0)

A_1 A_0 = 01 تقابل البوابة PB (PB_7 ..... PB_0)

A_1 A_0 = 10 تقابل البوابة PC (PC_7 ..... PC_0)

A_1 A_0 = 11 تقابل المسجل CWR

يظهر الشكل4–6 توزع المآخذ Pin configuration للدارة 8255.



شكل4–6: توزع إشارات الدارة 8255

يمكن التعامل مع كل من البوابتين A, B على أن طول كل منهما 8 Bits ، أما البوابة C فيمكن التعامل معها على أنها بوابة واحدة بطول 8 Bit ويمكن التعامل معها على أنها

بوابتين كل منهما بطول 4 Bit وهي مخصصة للتحكم بالبوابتين A , B عند النقل بالمصافحة.

وبما أن إشارة Reset فعالة بالواحد، توصل بشكل مباشر مع إشارة Reset out للمعالج ليتم تصفيرها عند تصفير المعالج. وعند تصفير هذه الدارة يتم تصفير كافة المسجلات الداخلية كما توضع كافة مآخذ البوابات على أنها دخل بأبسط شكل. ويظهر الشكل 5-6 مجمل التراكيب الممكنة لإشارات التحكم بالدارة 8255.

شكل5-6: مفهوم التراكيب المختلفة لإشارات التحكم بالدارة 8255

أنماط عمل الدارة 8255 A

تعمل الدارة 8255 بنمطين أساسيين يقابل كل منهما كلمة حالة خاصة به وهما:

– نمط الدخل والخرج I/O Mode ويقابل كلمة الحالة الأولى (التي يمكن بواسطتها تشكيل ثلاثة أشكال للنقل) و

– نمط التعبئة والتصفير BSR Mode – Bit Set–Reset Mode ويقابل كلمة الحالة الثانية.

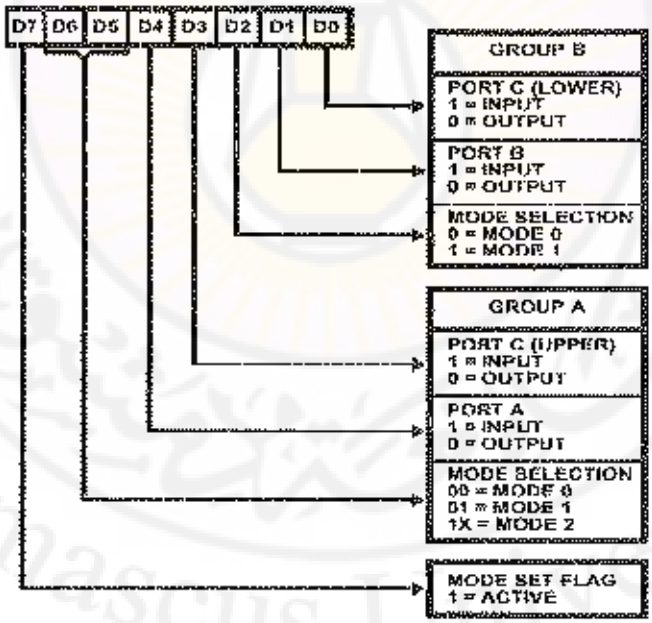
تحوي الدارة على مسجل حالة ويمكن برمجتها من خلال وضع كلمة في هذا المسجل

– في نمط الدخل والخرج I/O Mode، وفيه تعمل بوابات الدارة 8255 بشكل مبرمج بواحد من ثلاثة أنماط هي: Mode0, Mode1, and Mode2.

– في نمط BSR Mode وفيه يمكن استخدام البوابة C فقط لتعبئة أو تصفير كل بيت.

كلمات الحالة للدارة 8255

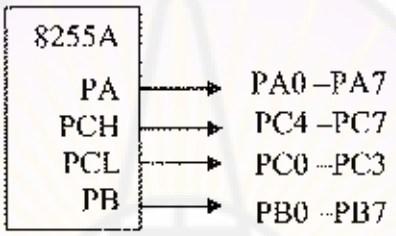
1– كلمة التحكم الأولى: وتقابل النمط I/O Mode وشكلها العام مبين بالشكل 6–6.



الشكل6–6: كلمة الحالة الأولى للدارة 8255

وهي تغطي صيغ الأنماط الثلاثة لعمل الدارة حيث إن

1- النمط Mode0 ويسمى أيضاً النمط الأساسي، يعدّ أبسط أشكال التعامل مع الدارة، لأنه بهذا النمط تستخدم كافة منافذ الدارة الـ 24 لنقل المعطيات بشكل بسيط كمدخل أو خرج أو الاثنين معاً بدون مصافحة. يظهر الشكل أعلاه لكلمة الحالة الأولى للدارة 8255 مفهوم كل بيت من بيتات كلمة الحالة التي ترسل لمسجل الحالة CWR عند برمجة الدارة. يحدد نوع الكلمة بالبت السابع والذي يأخذ القيمة واحد من أجل الكلمة الأولى وصفر من أجل كلمة الحالة الثانية. فمثلا كلمة حالة بالشكل 80H = 0000 1000 تقابل التوصيلة المبينة بالشكل 7-6



شكل7-6: توصيل بوابات الدارة 8255 بالـ Mode0 كخرج

أي أن بوابات الدارة الثلاثة بحالة خرج بالنمط Mode0 لان قيمة الصفر بالبيتات المقابلة للبوابات تعني أن البوابات بحالة خرج وأن قيمة صفر بالبت الثاني D_2 تعني أن البوابة PB تعمل بالنمط Mode0 ونفس الشيء بالنسبة لوجود قيمة الصفر في البتين D_5 و D_6 يعني أن البوابة PA تعمل بالنمط Mode0. وأن البوابة C تستخدم بالنمط Mode0 للنقل البسيط للمعطيات (خرج) مثلها مثل البوابتين A,B ولا تستخدم للتحكم، وأخيرا وجود واحد بالبيت D_7 يعني الكلمة الأولى حول تهيئة عمل الدارة.

2 - النمط Mode1 (Strobed Input / output Mode)

يستخدم بهذا النمط البوابتين A , B من أجل عملية إدخال المعطيات أو إخراجها بالمصافحة Handshaking وتخصص بيتات البوابة C للتحكم بهما. منافذ البوابة C تعمل كخطوط مصافحة بمعدل ثلاثة خطوط للتحكم بالبوابة A وثلاثة خطوط للتحكم بالبوابة B والمأخذين المتبقيين يستعملان للدخل والخرج العادي .

تخصص الخطوط PC_0, PC_1, PC_2 للمصافحة مع البوابة B

وتخصص الخطوط PC_3, PC_4, PC_5 للمصافحة مع البوابة A

والخطين المتبقيين PC_6, PC_7 يخصصان للنقل العادي للمعطيات بدون مصافحة. ويكون المفهوم التحكمي لبيتات البوابة C لكلا البوابتين A, B بالنمط Mode1 عندما تكون كل منهما بحالة الدخل، حيث D3 من كلمة الحالة تخصص لـ PC6 و PC7 بحيث واحد في هذا البيت يعني أن كلا من البتين دخل وأن صفر يعني خرج. وبخصوص بقية بيتات البوابة C فهي معرفة كالتالي:

PC_0 تخصص للبوابة B بمفهوم إشارة INTR–Interrupt وهي إشارة خرج من الدارة

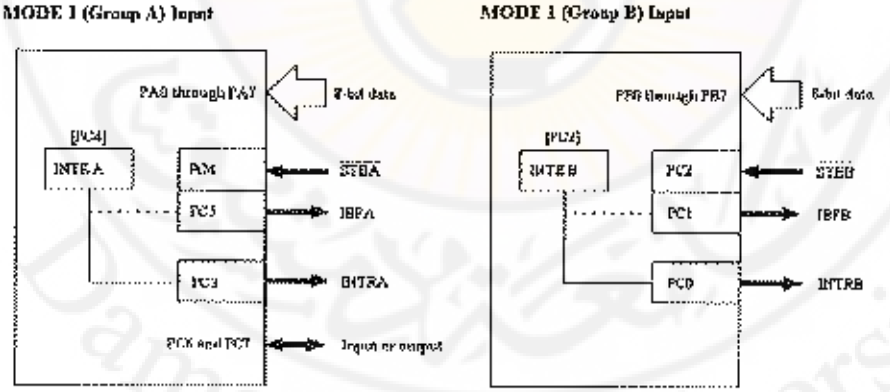
PC_1 تخصص للبوابة B بمفهوم إشارة IBF– Input Buffer Full وهي إشارة دخل

PC_2 تخصص للبوابة B بمفهوم إشارة STB'–Strobe وهي إشارة خرج من الدارة.

وبأسلوب مماثل تكون الإشارات المخصصة للتحكم بالبوابة A على النحو التالي:

PC_3 تخصص لـ INTR و PC_4 لـ IBF و PC_5 لـ STB'.

ويبقى البيتان PC_6, PC_7 للنقل البسيط كدخل أو خرج وهذا يحدد بقيمة البيت الرابع من كلمة الحالة، أي D_3 حيث إن قيمة صفر بهذا البت تعني أن الخطين بحالة خرج. يظهر الشكل 8–6 كلمة الحالة للدارة 8255 في الـ Mode1.

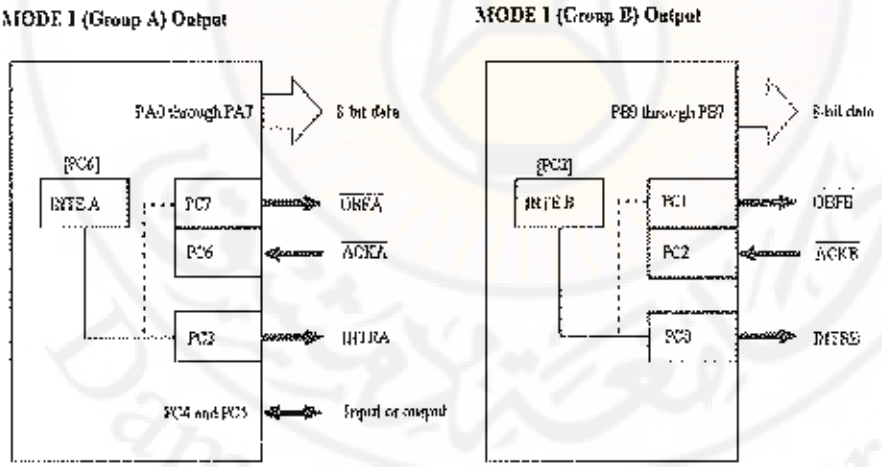


شكل8–6: كلمة الحالة وتوصيل بوابات الدارة 8255 بالـ Mode1 كدخل

يظهر الشكل8-6 الـ Mode1 للدارة 8255 الذي تكون فيه البوابتان A, B بحالة الدخل مع الشكل المقابل لتوصيل مأخذ البوابة C للتحكم بهما.

ويمكننا بأسلوب مماثل إظهار مفهوم بيتات البوابة C لكلا البوابتين A, B بالنمط Mode1 عندما تكون كل منهما بحالة الخرج، أي الكتابة من المعالج في البوابة، يخصص PC_0 للبوابة B بمفهوم إشارة $INTR_B$. و PC_1 تخصص للبوابة B بمفهوم OBF'– Oulput Buffer Full وهي إشارة خرج، وتخصص PC_2 للبوابة B بمفهوم ACK'– Acknowledge وهي إشارة دخل الدارة وباسلوب مماثل تكون الإشارات المخصصة للتحكم بالبوابة A على النحو التالي:

PC_3 بمفهوم إشارة $INTR_A$ و PC_4 بمفهوم إشارة OBF'_A و PC_5 بمفهوم ACK'_A ويبقى البيتان PC_7, PC_6 للنقل البسيط كدخل أو خرج وهذا يحدد بالقيمة الموجودة بالبيت الرابع من كلمة الحالة، أي D_3. يظهر الشكل9-6 كلمة الحالة للدارة 8255 في الـ Mode1 الذي تكون فيه البوابتان A, B بحالة الخرج مع الشكل المقابل لتوصيل مأخذ البوابات الثلاثة للدارة.



شكل9-6: كلمة الحالة وتوصيل بوابات الدارة 8255 بالـ Mode1 كخرج

3- النمط Mode2 (Strobed bidirectional I/O)

يمكن بهذا النمط أن تعمل البوابة A لدى الدارة 8255 لنقل المعطيات بالاتجاهين (Read, Write) وبهذه الحالة يلزمها خمس إشارات من البوابة PC للمصافحة هي: PC_3 يطبق عليها INTR الذي يستخدم عند كل عملية قراءة أو كتابة

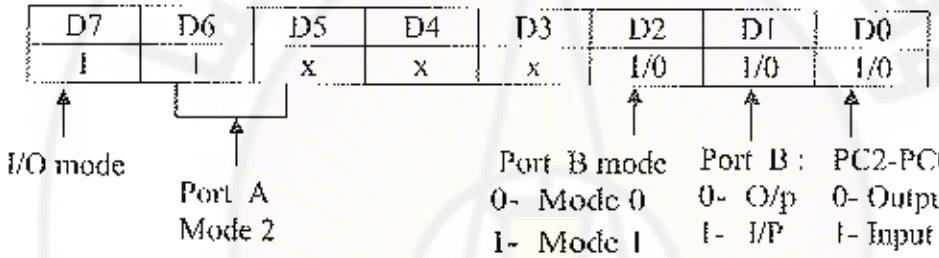
PC_4 يطبق عليها 'STB الذي يستخدم عند عملية القراءة

PC_5 يطبق عليها IBF الذي يستخدم عند عملية القراءة

PC_6 يطبق عليها 'ACK الذي يستخدم عند عملية الكتابة

PC_7 يطبق عليها 'OBF الذي يستخدم عند عملية الكتابة

يظهر الشكل10-6 الأشكال الممكنة لكلمة التحكم بالدارة 8255 بالـ Mode2

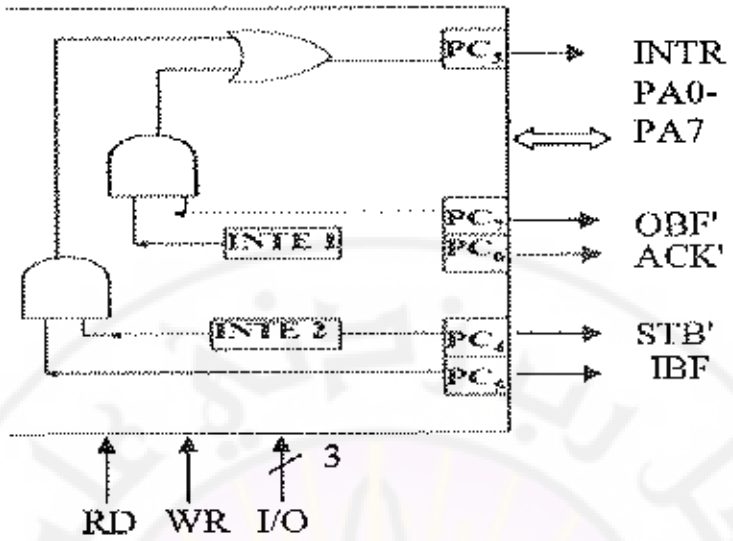


شكل10-6: الشكل العام لكلمة التحكم بالدارة 8255 بالـ Mode2.

حيث نصادف حالتين هما:

الأولى: هي أن تعمل البوابة A بالنمط Mode2 والبوابة B بالنمط Mode 0 (ويخصص قسم من بتات البوابة C للتحكم بالبوابة A والقسم المتبقي منها للنقل العادي). يظهر الشكل 11-6 التوصيل المقابل لكلمة الحالة المقابلة.

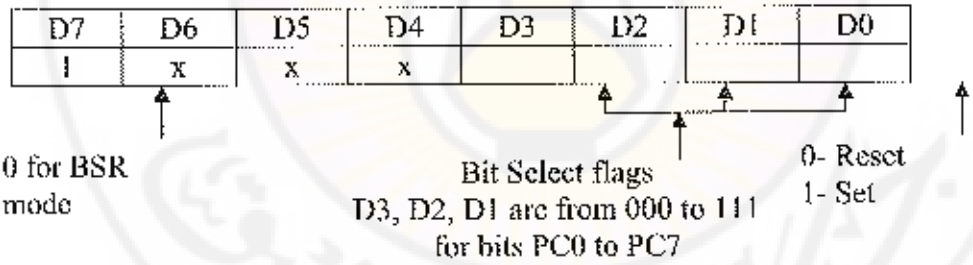
والثانية: هي أن تعمل البوابة A بالنمط Mode2 والبوابة B بالنمط Mode 1 دخل أو خرج وفي هذه الحالة يلزم البوابة A خمس اشارات للتحكم من البوابة C والاشارات الثلاثة المتبقية من C (PC_0, PC_1, PC_2) تخصص للتحكم بالبوابة B.



شكل 11-6: توصيل الدارة 8255 بالـ Mode2 والبوابة B بالـ Mode0

II- كلمة التحكم الثانية بالدارة 8255

وتقابل النمط BSR وهي مخصصة للتعامل مع البوابة C فقط. يظهر الشكل 12-6 الشكل العام لهذه الكلمة



الشكل12-6: كلمة الحالة الثانية للدارة 8255

يمكن بالنمط BSR جعل كل بيت من بيتات البوابة C في حالة 0 أو 1 وذلك بالارتباط مع قيمة البت الأول D_0 من كلمة التحكم. ويحدد رقم البيت الذي سيوضع بحالة Set أو Reset بالتوافقية الموجودة في البيتات $D_3D_2D_1$ من كلمة التحكم.

٦-٢ النقل التسلسلي للمعطيات

يقصد بالنقل التسلسلي للمعطيات مهما كان عرضها نقل كل بيت وراء بيت، وبالتالي يلزم بالنقل التسلسلي للمعطيات تحويل المعطيات من شكلها المتوازي بالحاسوب إلى صيغة تسلسلية وإرسالها على زوج من الأسلاك.

الفرق الجوهري بين النوعين الأساسين لنقل المعطيات هو أن النقل التفرعي يستخدم للمسافات القصيرة وهو مكلف إلا أنه سريع، أما النقل التسلسلي فيستخدم لنقل المعطيات إلى مسافات بعيدة ، إلا أنه بطيء. ويصنف النقل التسلسلي إلى ثلاثة أنواع نقل هي:

١ - النقل البسيط Simplex : وفيه يتم نقل المعطيات باتجاه واحد مثل الإرسال الإذاعي.

٢- نقل نصف ثنائي الاتجاه Half Duplex: وفيه يتم النقل بالاتجاهين، لكن النقل يحصل بوقت واحد باتجاه واحد، مثل النظام اللاسلكي.

٣- نقل ثنائي الاتجاه Full Duplex : وفيه إمكانية لإرسال المعطيات واستقبالها بآن واحد، مثل منظومة الهاتف أو الشبكات الحاسوبية.

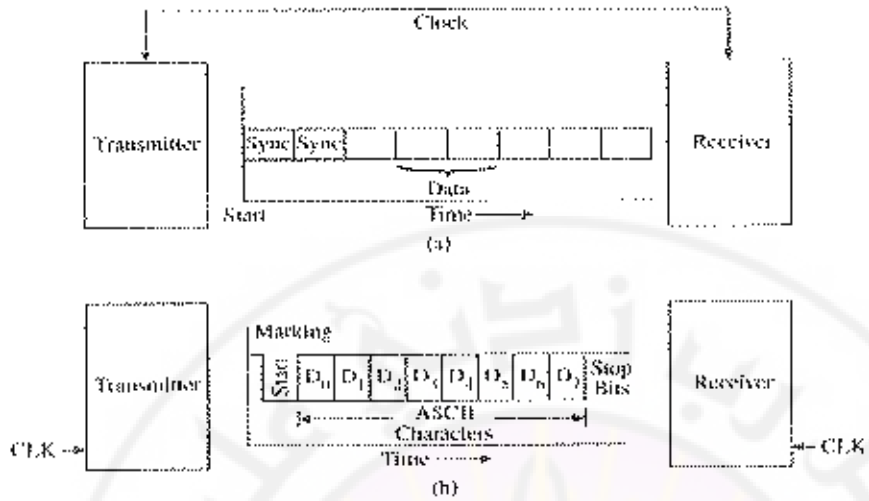
ونصادف نوعين من النقل التسلسلي للمعطيات هما:

النقل المتزامن Synchronous Serial Data Communication

والنقل غير المتزامن Asynchronous Serial Data Communication

يبين الشكل3-6 بشكل صندوقي الشكل العام لكل من النوعين، حيث إن الشكل a) يبين النقل المتزامن وفيه ترسل المعطيات على شكل كتل Blocks بين المرسل والمستقبل بمعدل ثابت وتعرف بداية ونهاية الكتلة بنماذج محددة من البتات (تسمى إشارات التزامن) ويكون كل من المرسل والمستقبل مقاد بنفس الساعة.

b) يبين الإرسال غير المتزامن، حيث إن كل حرف Character له بيت يعرفه في البداية وبيت واحد (أو بيتين) يعرفانه في النهاية . وبما أن كل حرف يكون معرفاً بشكل فردي ، فإنه يمكن إرسال هذه المحارف في أي وقت بشكل غير متزامن وأثناء النقل دوماً يرسل البيت LSB أولا .



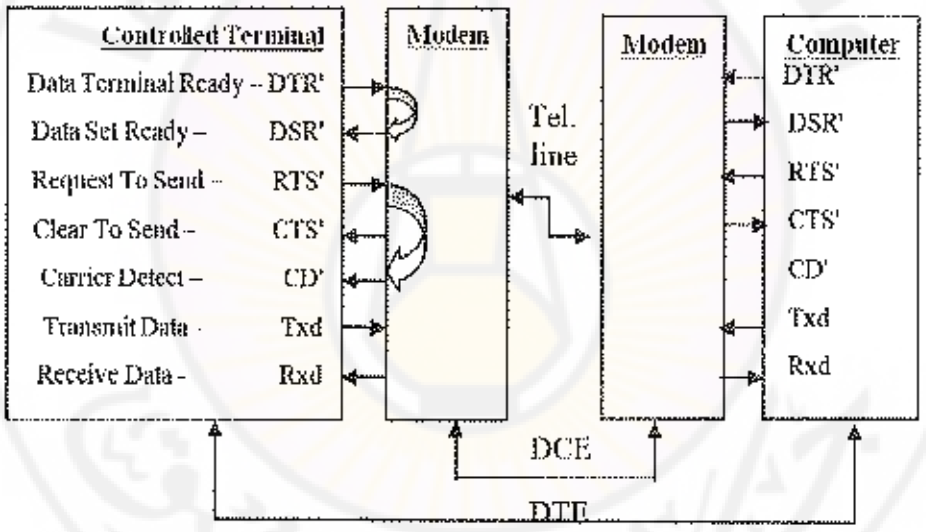
شكل13-6: النقل التسلسلي المتزامن وغير المتزامن للمعطيات

يمكن بالنقل التسلسلي أن تتألف الكلمة المرسلة من 5 , 6 , 7 , or 8 Bits يلحق بها بيتات التكافؤ Parity . يستخدم بالنقل التسلسلي للمعطيات مصطلح " معدل البود Baud Rate' ليدل على معدل إرسال المعطيات التسلسلية ، حيث إن المعدلات المستخدمة هي بشكل عام تبدأ من القيمة 110 BD وتتزايد وفق الشكل:

110 , 300 , 1200 , 2400 , 4800 , 9600 ,19200, ... BD

بالطبع لكي يتم ربط الحاسوب مع خطوط المعطيات التسلسلية يجب أن تحول المعطيات إلى الشكل التسلسلي. ويمكن تحقيق ذلك بعدة طرق ، منها استخدام بعض المسجلات الخاصة (مثل مسجلات دخل متوازي – خرج تسلسلي PISO أو ...) يوجد عدة طرق لإرسال المعطيات التسلسلية منها طريقة استخدام الحلقة التيارية Current Loop أي استخدام التيار في خط الإشارة لتمثيل " 1 " وعدم وجود التيار لتمثيل " 0 " إلا إن إمكانيات هذه الطريقة محدودة ببضعة آلاف الأمتار . لإرسال المعطيات التسلسلية لمسافات أطول يمكن استخدام نظام الهاتف القياسي لممر مناسب ، حيث يشار إلى خطوط الهاتف القياسية " بالخطوط المتبدلة أو المحولة Switched Lines " لان أي نقطتين يمكن أن توصلا معاً عبر سلسلة من المفاتيح ولهذه الخطوط عرض حزمة Width ... يتراوح من 300 إلى 3000 Hz فقط. لهذا السبب ولأسباب أخرى فإن الإشارات الرقمية لا يمكن

أن ترسل مباشرة على خطوط الهاتف القياسية (علماً أن خطوط الهاتف قادرة على حمل المعطيات الرقمية مباشرة ويمكن استئجارها ، لكن هذا مكلف ومقيّد بالغرض). يتم حل هذه المشكلة بتحويل الإشارات الرقمية إلى نبرات أو نغمات Tones ترددية سمعية تكون في المجال الترددي، بحيث تستطيع خطوط الهاتف إرسالها. يدعى الجهاز المستخدم لهذا التحويل ولتحويل النبرات المرسلة ثانية إلى معلومات رقمية بالموديم MODEM (Modulator – Demodulator). يوضح الشكل 14-6 كيفية وصل مود يمين ليسمحا لطرفية أن تتصل على خط هاتف مع حاسوب، حيث إن أسماء الإشارات المبنية بالشكل هي جزء من المفاهيم العائدة لاتصال المعطيات التسلسلية والتي تدعى بالوصلة RS – 232C ومن هذه الإشارات:



شكل14-6: ربط طرفية بحاسوب باستخدام الـ Modem

DTE– Data Terminal Equipment ترمز لمعدات (تجهيزات) طرفيات المعطيات

DCE – Data Communication Equipment ترمز لمعدات اتصال المعطيات.

٦-٢-١ دارة الدعم المبرمجة 8251A للنقل التسلسلي للمعطيات

تسمى الدارة 8251A بـ USART أي " المرسل- المستقبل العام المتزامن- غير المتزامن" (Universal Synchronous – Asynchronous Receiver – Transmitter) ولها مأخذين للنقل التسلسلي للمعطيات هما RxD و TxD.

الأول: مأخذ مستقل لاستقبال المعطيات التسلسلية عند القراءة على المدخل RxD (اختصارا لـ Receive Data)

والثاني: لإرسال المعطيات التسلسلية إلى المحيطيات عند الكتابة عبر المخرج TxD (اختصارا لـ Transmitted Data).

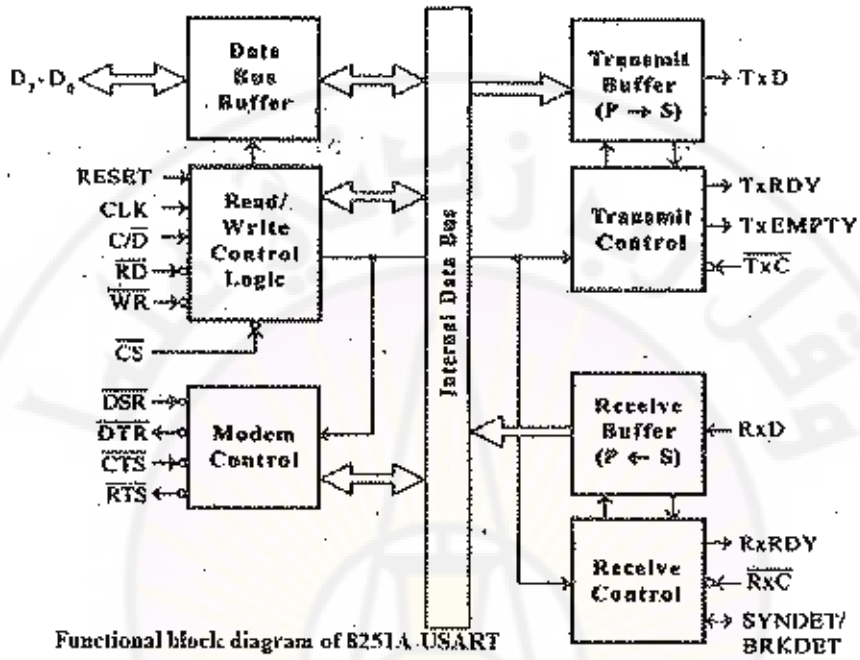
تضم الدارة 8251A مجموعة من الإشارات التحكمية مبينة بالشكل 14-6 ونوضح فيما يلي باختصار مفهوم كل منها كالتالي:

DTR–Data–Terminal Ready: تمكن الطرفية إشارة ' جاهزية طرفية المعطيات " لتخبر المودم بأنها جاهزة . وعندما يكون المودم جاهزاً لإرسال واستقبال المعطيات ، فإنه سيمكن إشارة DSR – Data Set Ready ' جاهزية وضع المعطيات ' إلى النهاية بطرفية وعندها يتخاطب المودم (Dials Up) مع الحاسوب .

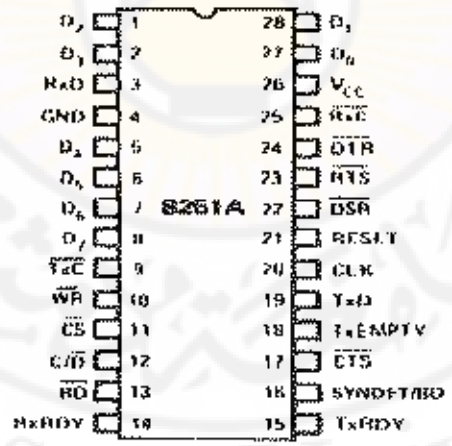
إذا كان الحاسوب موجوداً فإنه سيرسل نغمة محددة (عندما تمتلك النهاية الطرفية محرفاً حقيقاً جاهزاً للإرسال) تفعّل (تمكن) إشارة RTS – Request to Send " الطلب بقصد الإرسال" وبعد ذلك سيمكن المودم CD – Carrier Detect ' إشارة اكتشاف الحامل " إلى النهاية الطرفية ليدل بها على أنه يملك تماساً مؤسساً Established Contact للاتصال مع الحاسوب. عندما يكون المودم جاهزاً بشكل تام لإرسال المعطيات ، فإنه يمكن إشارة CTS – Clear To Send " المسح بقصد الإرسال العائدة إلى النهاية الطرفية، وبعدئذ ترسل النهاية الطرفية محارف المعطيات المتسلسلة إلى المودم. عندما ترسل النهاية الطرفية جميع المحارف اللازمة فإنها تجعل إشارتها RTS مرتفعة وهذا ما يسبب للمودم بعدم تمكين إشارته CTS ويوقف الإرسال.

إشارات وتوصيلات الدارة 8251A

يبين الشكل 15–6 المخطط الصندوقي التركيبية الداخلية لـ Intel 8251A USART ومآخذ الدارة Pin Configuration



Functional block diagram of 8251A USART



الشكل15–6: المخطط الصندوقي للدارة 8251A وتوزع الإشارات على المآخذ

خصصت الدارة 8251A من قبل شركة إنتل لتقوم بالاتصال المتزامن وغير المتزامن مابين المعالج المصغري والوسط الخارجي بشكل قابل للبرمجة وهي تمثل الجهاز النموذجي Serial Interface Device لربط الحاسوب (المعالج الصغري) مع مودم او أي جهاز آخر بحاجة إلى معطيات تسلسلية .

تمتلك الدارة 8251A عنوانين داخليين هما: عنوان التحكم Control وعنوان المعطيات Data حيث يتم انتخاب أي منهما بالقيمة على المأخذ 'C / D.

تصنع الدارة 8251A بالتغليف DIL – 28 أي ما يقابل 28 إشارة، أصبح معروفا منها المفهوم العام لـ 14 إشارة هي:

- إشارتين للتغذية UCC = + 5V و GND = 0V (pin 4, 26)
- ثمانية إشارات لنقل المعطيات D_0 ÷ D_7 مشكلة بذلك ممر المعطيات DB Bidirectional (Pin 1,2, 5,6,7,8, 27, 28)
- إشارتي تحكم بالقراءة والكتابة هما RD و WR (Pin 10,13)
- إشارة التصفير RESET (Pin 21) وإشارة الانتقاء 'CS (Pin 11)

وفيما يلي نشير بالجدول 1-6 إلى أسماء بقية إشارات الدارة وهي:

الإشارات 'RTS , 'CTS , 'DTR , 'DSR الموصولة بصندوق التحكم بالمودم Modem Control تمثل إشارات مصافحة ولها نفس المفهوم الذي سبق و أشرنا إليه عند الحديث عن النقل التسلسلي للمعطيات باستخدام المودم .

تتطلب معالجة المعطيات التسلسلية تزامن دقيق بين المرسل والمستقبل، بالتالي يفترض بجزء الإدخال والإخراج أن يحتوي على كافة الدارات اللازمة لهذا التزامن، حيث المدخل 'TxC : يمثل مدخل "ساعة مسجل إزاحة الإرسال"

والمدخل 'RxC : يمثل مدخل : "ساعة مسجل إزاحة الاستقبال"

ويربط هذان المدخلان عادة مع بعضيهما ليعملا بنفس تردد الساعة. ويجب أن يكون التردد المطبق لإشارة الساعة بمقدار 1 أو 16 ، أو 64 مرة من معدل البود للإرسال والاستقبال وذلك اعتمادا على النمط الذي يهيأ به الدارة .

الجدول1-6: إشارة الدارة 8251A المبرمجة للنقل التسلسلي

Pin20: CLK- Clock Pulse (TTL), In
Pin12: C/D'- Control Data – Is to be written or read, In
Pin9 : TxC'- Transmitter Clock, In
Pin19: TxD- Transmitter Data, Out
Pin25: RxC'- Receiver Clock, In
Pin3 : RxD- Receiver Data, In
Pin14: RxRDY- Receiver Ready (has character for CPU) Out
Pin15: TxRDY- Transmitter Ready(ready for character), Out
Pin16: SYNDE/BD- Sync Detect/Break Detect, Bidirectional
Pin18: TxEmpty- Transmitter Empty, Out
Pin22: DSR'- Data Set Ready, In , For control Pin24: DTR'- Data Terminal Ready, Out, For Modem Pin17: CTS'- Clear to Send Data, In, For Modem Pin23: RTS'- Request to Send Data, Out, For Modem

تستخدم الإشارة TxD للخرج التسلسلي والإشارة RxD للدخل التسلسلي للمعطيات. يصبح الخرج RxRDY فعالا (بالقيمة 1) عندما يكون عازل المسك فارغاً، ويمكن إرسال المحرف الآخر من الوحدة CPU . تصبح Tx Empty فعالة (بالقيمة 1) عندما يكون كل من عازل المسك ومسجل إزاحة الإرسال فارغين وسوف يصبح المأخذ RxRDY فعال (بالقيمة 1) عندما يزاح المحرف إلى داخل عازل المستقبل ويكون جاهزاً لقراءته من قبل الوحدة CPU .

تهيئة الدارة 8251A

إن هذه الدارة وغيرها من الأجهزة المحيطية الأخرى القابلة للبرمجة تكون عادة عند توصيل التغذية الكهربائية في "حالات غير معروفة Undefined States" وتهيئة أي دارة لابد في البداية من تعرّف التركيبة العامة لها وعلى صيغة كلمة (كلمات) التحكم الواجب إرسالها. وبعد ذلك فإن عملية التهيئة ليست بالأمر الصعب. وعند تهيئة أي دارة يجب سلوك خطوات محددة يأتي في مقدمتها تحديد عنوان الدارة في المنظومة (عن طريق دارة مفكك شيفرة) وبعد ذلك تحديد العناوين الداخلية لكل من مسجلات التحكم، المنافذ، المؤقتات وغيرها بحسب نوع الدارة.

وبما أن الدارة 8251A تعمل بالنمطين المتزامن واللامتزامن يمكن النظر إليها على أنها مكونة من جزأين منفصلين في نفس الـ Chip. ولكي تتم تهيئة الدارة 8251A ينبغي أولا إرسال كلمة النمط Mode Word وبعدئذ كلمة الأمر Command Word إلى عنوان مسجل التحكم. يبين الشكل 16-6 صيغ هذه الكلمات التي تقرأ من نفس العنوان.

1- كلمة الأمر (صيغة لتعليمة الأوامر) Command instruction Format كالتالي

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
EH	IR	RTS	ER	SBRK	RxE	DTR	TxEN

TxEN – Transmit Enable (1- Enable, 0- Disable)

DTR – Data Terminal Ready (1 will Force DTR' Output to Zero)

RxE – Receive Enable (1- Enable, 0- Disable)

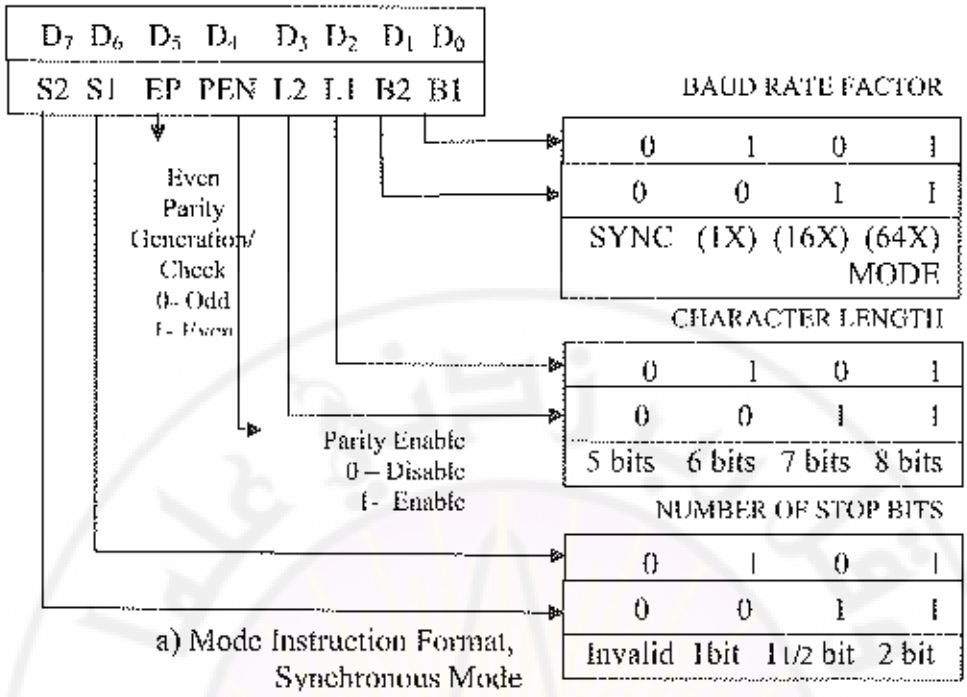
SBRK – Send Break Character (1 Force TxD "low", 0- Normal)

ER – Error Reset (1 Reset Error Flag PE, OE, FE)

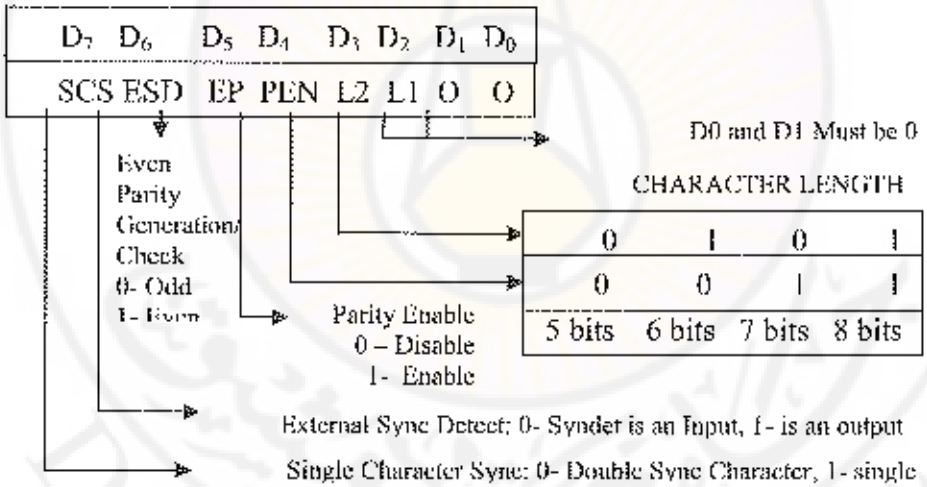
RTS – Request to Send (1 will Force RTS' Output to Zero)

IR – Internal Reset (1 Return 8251 to Mode Instruction Format)

EH – Enter Hunt Mode (1 Enable Search for Sync, Characters).



a) Mode Instruction Format, Synchronous Mode



b) Mode Instruction Format, Asynchronous Mode

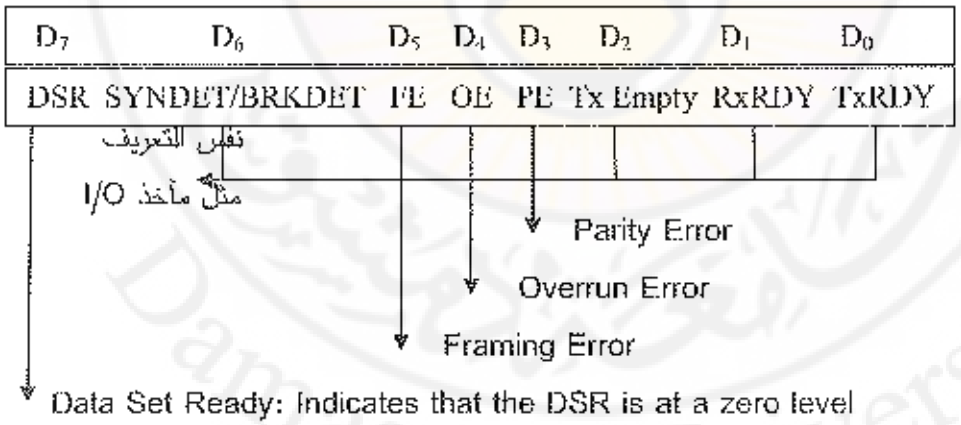
الشكل 16-6: صيغ كلمات التحكم للدارة 8251A

يحدّد البيتان D_0,D_1 (B_2B_1) من كلمة النمط عامل Baud Rate Factor معدل البود أي النسبة بين إشارة الساعة المطبقة على المداخل 'RxC'–'TxC ومعدل البود. فمثلاً إذا أردنا أن نستخدم 'TxC بقيمة 19200 Hz وأرسلت المعطيات عند معدل 1200 Bd يكون عامل معدلا البود مساوياً إلى 1200 / 19200 أو X 16.

حالة خاصة : إذا كان $D_1\ D_0 = 00$ فإن الدارة تبرمج من أجل نقل معطيات متزامنة وفي هذه الحالة سيكون معدل البود نفسه مطبقاً على الخطين 'TxC و 'RxC (بقية التراكيب الثلاثة لهذين البيتين تعمل للنقل غير المتزامن). يمكن استخدام عامل معدل بود قيمته واحد فقط من أجل نقل غير متزامن إذا كان كلّ من نظام الإرسال ونظام الاستقبال يستخدمان نفس الخطوط 'TxC و 'RxC . ويحدّد البيتان D_2 و D_3 (أي L_2 L_1 في كلمة النمط) طول المحرف بدون البيتات الإضافية (من بيت الإقلاع ، بيت التكافؤ وبيتات التوقف).

D_4 تقابل الرمز Parity Enable – PEN وتعني إذا كان هذا البيت بالقيمة (1) أنه يسمح باستخدام الـ Parity أما واحد بالبيت D_5 يعني Even Parity – EP وأخيراً البتين D_6 , D_7 يدلان على عدد بيتات التوقف بالنسبة للنقل اللاتزامني .

II– كلمة الأحوال (صيغة لقراءة الحالة) Status Read Format



بعد الانتهاء من إرسال كملة النمط ، ينبغي إرسال كلمة الأمر ، حيث إن وجود قيمة (1) في البيت D_0 ~ TxEN يمكن القسم المرسل بالدارة 8251 A والمخرج TxRDY وهذا يعني أن المخرج TxRDY سيصبح فعالا إذا أصبح المدخل 'CTS فعالا. ويكون عازل المسك للمرسل جاهزا من أجل حرف آخر من CPU.

إن وضع (1) بالبيت D_1 لكلمة الأمر سوف يسبب المخرج 'DTR أن يصبح منخفضاً وهذا بدوره يخبر المودم أن النهاية الطرفية أو الحاسوب جاهزاً للعمل. وإن وضع (1) بالبيت D_2 يمكّن مأخذ الخرج RxRDY وإذا مكنت فإنها ستصبح مرتفعة عندما يكون لدى الدارة 8251A محرف في عازل مستقبلها جاهز للقراءة . ويتم تصغير RxRDY عندما تتم قراءة محرف من الدارة 8251 A .

وإن وضع (1) بالبيت D_3 المقابل لـ SBRK يجعل مخرج الخرج للدارة كله أصفاراً ويستخدم للدلالة على نهاية كتلة من المعطيات المرسلة .

وإن إرسال كلمة الأمر بوجود (1) في البيت D_4 المقابل (Error Reset ~ ER) يسبب الدارة في تصغير أعلام التكافؤ PE و التجاوز OE--Overrun وخطأ الإطار FE-- framing error في مسجل الأحوال للدارة 8251A.

وإن وجود (1) في البيت D_5 المقابل لـ RTS يسبب للدارة تمكين طلب الإرسال 'RTS بخرج منخفض.

وإن وضع (1) بالبيت D_6 المقابل لـ IR يسبب بتصفير الدارة داخلياً عندما ترسل كلمة الأمر . وبعد أن يرسل أمر التصفير برمجياً بهذه الطريقة ، فإنه يجب إرسال كلمة نمط جديدة . وأخيراً يستخدم البيت D_7 المقابل لـ EA فقط عندما تعمل الدارة بالنمط المتزامن تكمن مشكلة النقل المتزامن في المولد المركزي للنبضات لقيادة كل من المرسل والمستقبل. يقرر ماذا سيكتب في الدارة من تعليمات أوامر أو معطيات عن طريق الإشارة 'C/ D ويمكن تغير نظام برمجة الخرج التسلسلي فقط بالتصفير أو الإجبار على التصفير عبر البيت D_6 (IR) بكلمة التحكم. يمكن بعد التصفير إرسال كلمات التحكم والنمط المطلوبة إلى الدارة التي تميز كلمة الأمر من كلمة النمط بالترتيب الذي ترسل به إلى الدارة ، حيث بعد التصفير يجب إرسال كلمة النمط إلى عنوان الأمر Command Address لان أي

كلمة ترسل إلى عنوان الأمر بعد كلمة النمط ستعالج كأمر حتى تصفير الدارة من جديد. وبالإضافة إلى كتابة كلمة النمط والأمر يمكن بأي وقت قراءة حالة الدارة. تعمل الدارة 8251 A بالنقل المتزامن من أجل 5 وحتى 8 رموز ، بحيث إن توضع الرموز الداخلية أو الخارجية يتم بشكل تلقائي تزامني.

أما بالنسبة للنقل اللاتزامني فإن الدارة تعمل من أجل 5 وحتى 8 رموز لكن بتدرج من واحد وحتى 16 ، أو 64 مرة من سرعة النقل . سرعة النقل خلال النقل التزامني تتراوح من 0 وحتى 56 Kb/s ، وبالنقل اللاتزامني من 0 و حتى 9,6 Kb/s.

يمتلك المأخذ " اكتشاف التوقف / اكتشاف التزامن (SYNDET / BD) استخدامين : فعندما تعمل الدارة في النمط غير المتزامن يصبح هذا المأخذ بحالة الواحد إذا بقي RxD غير فعال لزمن أكبر من زمن محرفين، وعندما تبرمج الدارة من أجل إرسال معطيات متزامنة ، فإنها ستصبح مرتفعة عندما تجد الدارة 8251 محرف تزامن .

٦-٣ دارة الدعم المبرمجة 8253/8254 للتوقيت/العد

تحتوي كل من الدارتين 8253 و 8254 (Programmable Timer/ Counter) على ثلاث مسجلات كل منها بطول 16 bit يمكن برمجتها لتعمل في عدة أنماط مختلفة. الدارتان 8253 و 8254 متماثلتان، إلا أن الدارة 8254 تملك زيادة عن الدارة 8253 ميزة "القراءة العكسية Read Back" التي تسمح بحفظ العَد في كل العدادات، وكذلك حالة العداد عند أي نقطة. وبالتالي نكتفي بالإشارة للدارة 8254 لان الدارتين متوافقتان في المأخذ والوظيفة. الاختلاف البارز بين الدارتين هو أن تردد ساعة الدخل الأعظمي للدارة 8253 هو 2.6 MHz، بينما التردد الأعظمي للدارة 8254 هو 8 MHz.

تملك الدارة 8254 ممر معطيات بعرض 8 bit ومدخل تمكين فعّال بالصفر، إضافة إلى مدخلي عنوان A_0, A_1 لعنونة واحد من العدادات الثلاثة أو مسجل كلمة التحكم الموجود في داخل الدارة. يبين الشكل 17-6 المخطط الصندوقي للدارة 8254 الذي يظهر مداخل العدادات ومخارجها، حيث يمكن تطبيق إشارة بأي تردد بالمجال من0 وحتى8 MHz على مدخل ساعة العداد CLK.

شكل17-6: المخطط الصندوقي وتوزع الإشارات للدارة 8254

يعدّ وجود الدارة 8253/8254 مفيداً جدا في منظومة الحاسوب الصغري لأنه يمكن برمجتها بتعليمات معينة لتقسيم تردد الدخل على أي عدد مطلوب (خاصة وأن استخدام دارات مؤقت زمني مثل الدارة 555 لا يحقق الدقة المطلوبة للعديد من التطبيقات، كما أن استخدام إشارة مشتقة من هزاز مقاد بكريستال مثل إشارة ساعة المعالج يكون ترددها مرتفع جدا لقيادة مدخل مقاطعة المعالج مباشرة).

يسمح المدخل Gate الموجود على كل عداد بأن نبدأ أو نوقف ذلك العداد بإشارة ناتجة من مكونات صلبة خارجية. فإذا كان المدخل Gate لأحد العدادات مرتفعا، يكون ذلك العداد ممكنا للعَد. والتواتر الناتج من كل عداد يظهر على مخرجه Out. لتهيئة الدارة 8254 يجب إرسال كلمة تحكم إلى مسجل التحكم لكل عداد يراد استخدامه، حيث إن الشكل 18-6 يبين صيغة كلمة التحكم للدارة 8254. يتضح من الشكل العام للكلمة أن أعلى بتين من كل كلمة تخصص لإخبار الدارة 8254 بالعداد الذي يراد تهيئته.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

شكل18-6: صيغة كلمة التحكم للدارة 8254

يستخدم البتان RW1, RW2 لتعين كيفية كتابة العدد أو قراءته من عداد ما، حيث يمكن أن نقرأ العدد من أحد العدادات في أي وقت.

أما البتات M0, M1, M2 فتحدد نمط العدّ، حيث إن العدد الثنائي الذي نضعه في هذه البتات من كلمة التحكم، يعين التأثير الذي سيجري على عملية العدّ بوساطة مدخل البوابة، وعلى شكل الموجة التي سيتم إنتاجها على المأخذ Out.

فعلى سبيل المثال إذا عينت النمط 3 لعداد ما بوضع القيمة 011 في هذه البتات الثلاثة، سيوضع العداد في نمط الموجة المربعة، وفي هذا النمط سيكون الخرج مرتفعاً من أجل النصف الأول للعدّ المحمل، ومنخفضا من أجل النصف الثاني للعدّ المحمل. وعندما يصل العدّ إلى الصفر، يعاد تحميل العدد الأصلي آليا ويكرر العد التنازلي، وعندئذ سيكون شكل الموجة على المأخذ Out في هذا النمط عبارة عن موجة مربعة بتردد مساوي لتردد ساعة الدخل مقسوما على العدد الذي كتب في العداد. وبخصوص البت الأخير BCD في كلمة التحكم، إن العدادات ذات الطول 16 bit الموجودة في الدارة 8254 هي عبارة عن عدادات تنازلية Down-Counter وهذا يعني أن العدد المحمل في العداد سيتم إنقاصه عند كل نبضة ساعة.

ويمكن برمجة الدارة 8254 لتعدّ تنازليا عدد محمل بصيغة الـ BCD (عشري) أو بالثنائي. فإذا وضعنا $D_0 = 0$ بكلمة التحكم سيتعامل العداد مع العدد المحمل على أنه عدد ثنائي، ومن أجل هذه الحالة فإن أضخم عدد ممكن أن نحمله هو FFFFH. أما إذا جعلنا $D_0 = 1$ فإن أضخم عدد ممكن أن نحمله في العداد هو 9999H.

أنماط العدادات للدارة 8254

يمكن لعدادات الدارة 8254 أن تبرمج لتعمل بواحد من ستة أنماط مختلفة هي:

النمط 0 – المقاطعة على عدّ نهائي

النمط 1 – المكونات المصلبة القابلة لإعادة القدح مرة واحدة (نبضة واحدة)

النمط 2 – مولد مقاطعة مؤقتة

النمط 3 – نمط الموجة المربعة

النمط 4 – نبضة تمكين مطلقة صادرة بوساطة مكونات لينة

النمط 5 – نبضة تمكين صادرة بوساطة مكونات صلبة.

مثال6-1: صمم منظومة مايكرو كومبيوتر متكاملة باستخدام المعالج 8080A للتحكم (مثلا بمرجل كبير) من خلال مراقبة كمية المواد والحرارة والضغط، بحيث تضم المنظومة بالإضافة للمعالج ذاكرة للقراءة فقط من نوع EPROM بسعة 32 KB وذاكرة من نوع RAM بسعة 32KB مع بوابة دخل للنقل بالنمط البسيط وبوابتي خرج كل منها بعرض 8 bit تعمل بنمط المصافحة.

الجواب: يوجد بهذا المثال مقارنة مع المثالين (5-2 ،5-1) نقطتين جديدتين هما: وجود حاجة لوصل مقاطعات خارجية ووجود حاجة لاستخدام دارات مبرمجة. أي يجب تعديل توصيلة المعالج 8080A بالشكل 5-11 بإضافة دارة للتحكم بالمقاطعة وفق ما هو مبين بالشكل 6-19 مع مراعاة مايلي:

عند ربط الحساسات العائدة للكمية والضغط والحرارة بالدارة 8214 لتوليد الإشارة INT لمقاطعة عمل المعالج والذهاب لتخديم أي طلب مقاطعة من هذه الحساسات يجب:

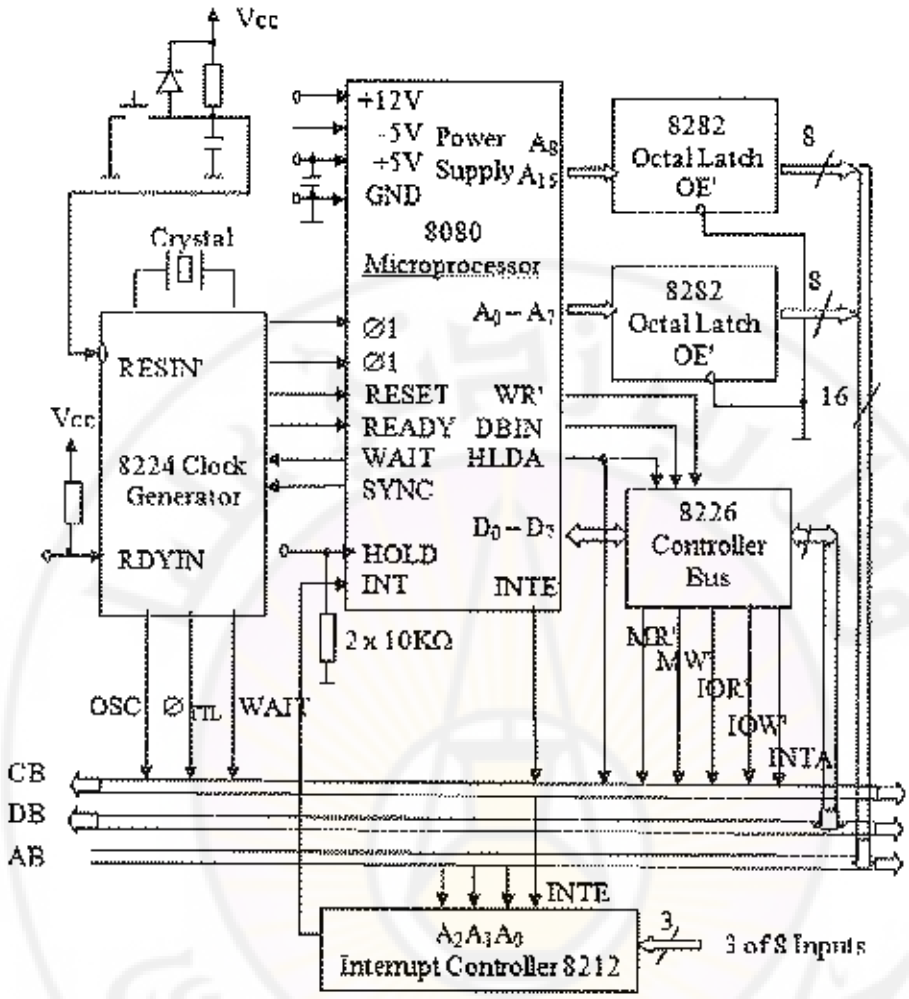
- وصل الحساسات على مداخل الدارة 8214 الثمانية بحسب الأفضلية المقترحة.
- توصيل مداخل مفكك الترميز الداخلي على خطوط العنونة A_0, A_1, A_2 لتحديد أي من المآخذ الثمانية التي يمكن بلحظة ما وصلها بالدارة.
- التحكم بعمل الدارة برمجيا من خلال التعليمة RST n

أما بخصوص النقطة الثانية حول استخدام الدارات المبرمجة نعدل الجزء الثاني من التصميم بحيث يضم بالإضافة لما هو مطلوب من ذواكر، دارة دعم مبرمجة لتأمين نمط النقل بالمصافحة وفق ما هو مبين بالشكل 6-20. وبهذه الحالة يتم وصل الدارات بشكل قياسي ويتم التحكم بها عن طريق تغير كلمة الحالة فقط.

بما أن المطلوب هو تأمين بوابتي خرج، يجب اختيار دارة دعم مثل الدارة 8255 لتعمل بالـ Mode1 ويجب تحديد كلمة الحالة وفق التالي: للمجموعة B نستخدم الكلمة

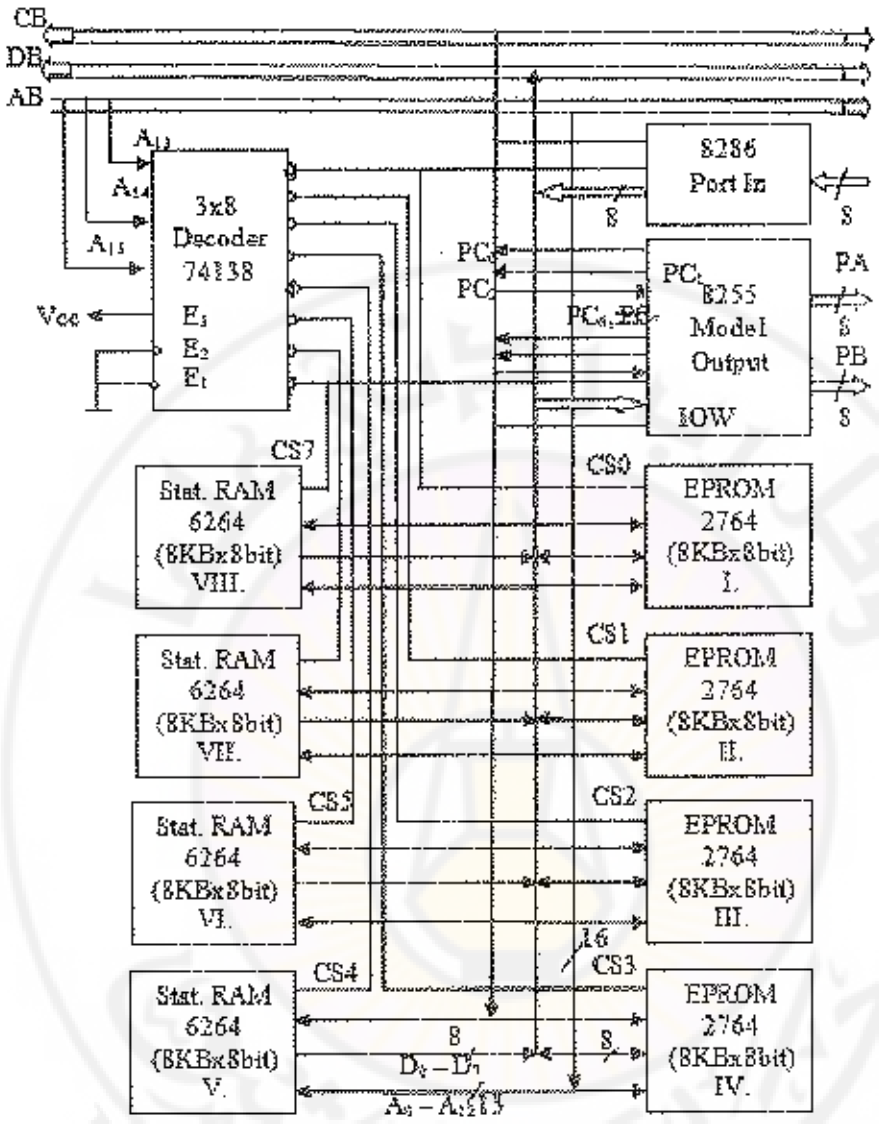
$D_7D_6D_5D_4D_3D_2D_1D_0$ = 1 x x x x 1 0 x وللمجموعة A نستخدم الكلمة

1 0 1 0 1/0 x x x حيث إن D_3 إذا كان 0، يعني أن البتين PC_6, PC_7 يكونان خرجاً، وإن كان يساوي 1 يكونان دخلا بالنمط البسيط.



الشكل19–6: توصيل دارة المعالج 8080 بوجود متحكم للمقاطعة 8214

أما بخصوص بوابة الدخل المطلوب منها أن تعمل للنقل بالنمط البسيط، يكفي استخدام أي من البوابات العادية غير المبرمجة بعرض 8 bit ، أو أي بوابة إضافية من بوابات الدارة 8255 غير المستخدمة للعمل بالمود Mode 0.



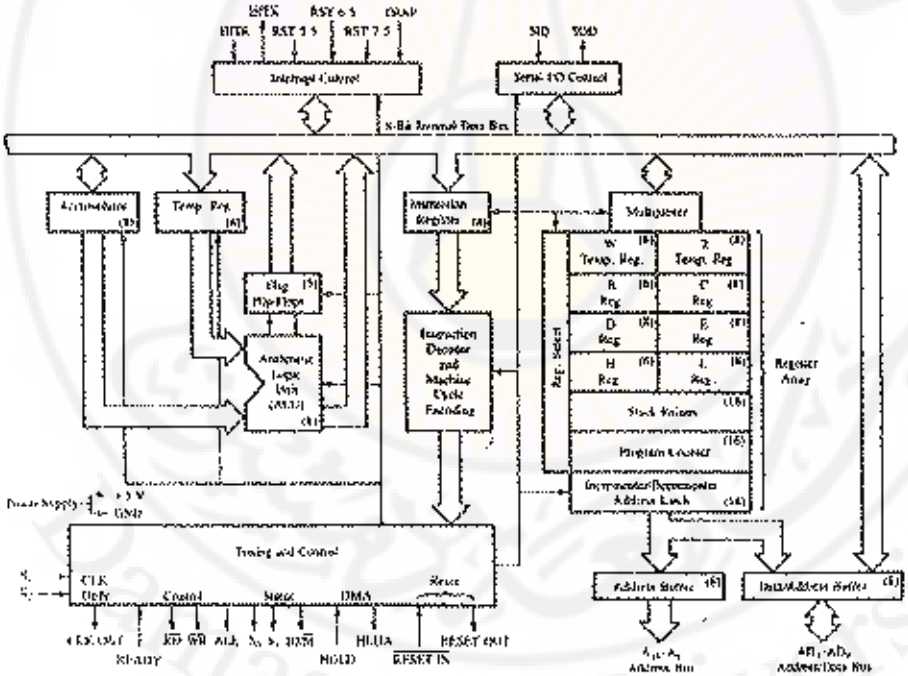
الشكل 20–6: تصميم مايكرو كومبيوتر باستخدام المعالج الصغري 8080A

وبوجود دارات دعم مبرمجة

يصنع المعالج الصغري 8085 بالتقانة NMOS ويظهر بعدة أشكال هي: 8085 , 8085A ، 8085A2 (كما ويصنع المعالج 8085 أيضاً بالتقانة CMOS بالاسم 80C85 بسحب تيار كهربائي يصل إلى 170 mA). يعدّ المعالج 8085A نسخة مطورة عن المعالج 8080A بالرغم من أنه يصنع بنفس التغليف DIL – 40 حيث تم فيه الاستغناء عن بعض الإشارات ودمج بعض الإشارات مع بعضها البعض وتم إضافة العديد من الإشارات الجديدة.

٧-١ التركيبة الداخلية للمعالج 8085

يظهر الشكل1-7 التركيبة الداخلية لهذا المعالج Functional Block Diagram وهي مشابهة لبنية المعالج 8080A مع بعض التعديلات (التطويرات).



شكل1-7: التركيبة الداخلية للمعالج 8085

يمتلك المعالج 8085A نفس ذاكرة المسجلات Registers للمعالج 8080A وله نفس المسجلين PC و SP وكذلك نفس مسجل المؤشرات Flag Register . بالرغم من أن للمعالج 8085A نفس عدد الـ Pins أو المآخذ التي يملكها المعالج 8080A إلا أنه يملك العديد من الإضافات والتحسينات. وبمقارنة أولية بين المعالجين 8080A و 8085A يمكننا ملاحظة التعديلات (التحسينات) التي طرأت على المعالج 8080A، حيث بالنسبة للمعالج 8085A، تم

- تركيب ممرات المعطيات الثمانية $D_0 - D_7$ على الخطوط الثمانية الدنيا من ممر العناوين $A_0 - A_7$ لتصبح مركبة مع بعضها Multiplexed Address/data Bus بالاسم $AD_0 - AD_7$.

- تخفيض عدد مآخذ الساعة من فازين لدى المعالج 8080A (∅1, ∅2) إلى فاز ساعي واحد ∅ لدى المعالج 8085A .

- تخفيض عدد مآخذ التغذية لدى المعالج 8080A من أربعة مآخذ إلى مأخذين فقط لدى المعالج 8085A هما: Vcc = +5V والـ GND وتم الاستغناء عن الباقي.

- كما تم الاستغناء عن إشارتي الخرج WAIT و SYNC الموجودتين لدى 8080A.

- توليد الإشارة ∅ (فاز واحد) بشكل مباشر على الـ Chip التي تؤمن بوجود المداخل X1 , X2 لوصل الكريستال أو عنصر RC أو LC (عوضاً عن الفازين لدى المعالج 8080A) مع العلم أن المعالج 8080 لايملك مآخذ لوصل الكريستال بشكل مباشر وإنما يتم تأمين الـ Clock اللازم للمعالج من الدارة الخارجية 8224. بمعنى أنه يكفي بالنسبة للمعالج 8085 وصل كريستال خارجي ليعمل المعالج مباشرة وأنه لا داعي لاستخدام دارات دعم مثل 8228 ,8224 (وكأنها موجودة مع المعالج على نفس الشريحة). يقوم المعالج بتقسيم قيمة الكريستال الخارجي ضمن الـ Chip على 2 وأن أكبر كريستال يمكن وصله مع 8085 A2 هو 10 MHz بزمن ولوج قدره 200 ns.

- أضيف للمعالج 8085A إشارة ALE (Address Latch Enable) لفصل المعطيات عن العناوين، حيث إن اشارة ALE تفعّل بالزمن T1 من كل حلقة آلة لتشير إلى أن البيانات على خطوط الممر (Bus)هي عناوين،

- يمتلك المعالج 8085A إشارتين جديدتين غير موجود مماثل لهما لدى المعالج 8080A وهما S_0, S_1 والذي يمكن من خلالهما معرفة حالة المعالج.

- يوجد كتلتان جديدتان ضمن المعالج 8085 غير موجودتين لدى المعالج 8080A:

الأولى: الكتلة الإضافية بالاسم Serial Control I/O الخاصة بالنقل التسلسلي للمعطيات ويقابل هذه الكتلة مأخذين جديدين هما:

SID– Serial Input Data لإدخال المعطيات بشكل تسلسلي

SOD – Serial Output Data لإخراج المعطيات بشكل تسلسلي

وبالتالي لتخديم هذه الكتلة يوجد لدى المعالج 8085A تعليمتان إضافيتان، هما:

RIM من Read Interrupt Mask و SIM من Set Interrupt Mask.

وهكذا أصبح المعالج 8085A يملك 80 تعليمة مع بقاء الـ OP.Code (الرمز العملياتي) لتعليمات المعالجين 8080A و 8085A نفسه.

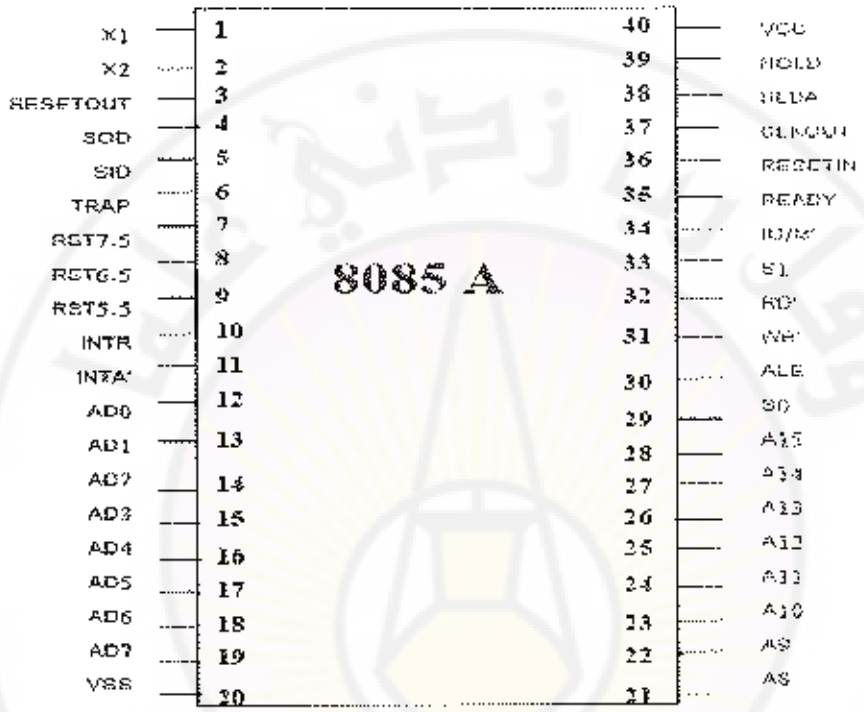
الثانية: الكتلة الإضافية بالاسم Interrupt Control التي تضم بالإضافة للإشارتين INT و INTE الموجودين أربعة مداخل أخرى إضافية للمقاطعة مبينة بالجدول 1-7 مع العنوان المقابل لكل منها وترتيب تنفيذها (Priority).

الجدول1-7: المقاطعات لدى المعالج 8085A (العنوان والأفضلية)

Name	Priority	Address Branched
TRAP	1	24H
RST 7.5	2	3CH
RET 6.5	3	34H
RST 5.5	4	2CH
INTR	5	Like 8080

٢-٧ إشارات المعالج 8085A

يظهر الشكل 2-7 توزع إشارات المعالج 8085A (Pin Configuration)، حيث يمتلك المعالج ممر للمعطيات والعناوين مركبة (أو مشتركة) مع بعضها البعض.



شكل2-7: توزيع الإشارات للمعالج 8085A

بالتالي في حال استخدام ذاكرة مع المعالج 8085A يجب في البداية فصل المعطيات عن العناوين بوساطة دارة إضافية مثل المسجل 8212 أو أي دارة أخرى بممر عرضه 8 bit وقابلة للقيادة بإشارة خارجية مثل ALE من المعالج 8085A. وهنا سنكتفي بالتعريف بإشارات المعالج 8085A الجديدة ومنها:

- SID, SOD وتستخدم لنقل المعطيات بشكل تسلسلي كل بيت وراء الآخر أو إدخالها عبر البيت السابع للمراكم.

- S_0 , S_1 إشارات الحالة Status Signals لدى المعالج 8085A. وهي اشارات خرج، حيث أن الإشارة S_0 تؤدي دور إشارة القراءة و S_1 تؤدي دور إشارة الكتابة، ويمكن لهذه الغاية استخدام مثل هذه الإشارات في الأنظمة الكبيرة . ترّمز الإشارتين S_1 ، S_0 عبر Decoder لإظهار حالة المعالج بأي لحظة وفق القيم المبينة بالجدول2-7.

الجدول2-7: ترميز إشارات الحالة لدى المعالج 8085A

S_1S_0	Status
0 0	Halt
0 1	Read
1 0	Write
1 1	Fetch

- TRAP, RST7.5, RST6.5, RST5.5: إشارات خاصة بالمقاطعة إضافة لإشارة INTR المماثلة لإشارة INT لدى المعالج 8080A، أي أن المعالج 8085A يملك خمسة مداخل للمقاطعة، تختلف عن بعضها البعض باختلاف الأفضلية لكل منها والعنوان الذي تخّدم منه وفق ما هو مبين بالجدول أعلاه.

مدخل المقاطعة TRAP غير القابل للتقنيع Nonmaskable Interrupt ويملك الأفضلية الأعلى من بين المقاطعات الأخرى.

- IO / M' وهي إحدى الإشارات الجديدة للمعالج 8085A تستخدم لتحديد مكان تعامل المعالج إن كان مع البوابات أو مع الذاكرة Input Output/Memory، بحيث إذا كانت هذه الإشارة IO / M' = 0 يكون التعامل مع الذاكرة. ويمكن ترميز هذه الإشارة مع إشارات القراءة RD' والكتابة WR' للمعالج 8085A وفق ما هو مبين بالجدول 7-3a للحصول على بعض إشارات كلمة الحالة للمعالج.

ويمكن ترميز الإشارتين S_1 , S_0 مع إشارة IO/M' عبر Decoder 3x8 للحصول على بعض الحالات وفق ما هو مبين بالجدول 7-3b.

الجدول3-7: ترميز إشارتي RD', WR' وإشارتي S0, S1 مع الإشارة IO/M'

IO/M'	WR'	RD'	Operation
0	0	1	Memory write = MW'
0	1	0	Memory Read = MR'
1	0	1	I/O Write = IOW'
1	1	0	I/O Read = IOR'

-a -

IO/M'	S_0	S_1	Operation
0	1	1	Op Code Fetch
0	0	1	Memory Write
0	1	0	Memory Read
HZ	0	0	Halt
1	0	1	I/O Write
1	1	0	I/O Read
1	1	1	Interrupt Acknowledge

-b -

- CLK Out: وهي إشارة خرج من المعالج 8085A جواباً لإشارة الساعة، بمعنى أنه عند توليد الساعة اللازمة للمعالج من الكريستال الخارجي يعطي المعالج إشارة ساعة على المخرج CLK Out بحيث يمكن منها إمداد أي دارة أخرى بالساعة اللازمة لها.
- Reset out: وهي إشارة خرج تقابل إشارة تصفير المعالج، يمكن من خلالها تصفير الدارات المرتبطة بالمعالج بدون تشكيل منطق إضافي للتصفير.

وهنا نشير إلى أن بعض الإشارات قد يتغير مستوى تفعيلها أو اسمها بين معالج وآخر، مثل إشارة التصفير بالمعالج 8080A (والتي تسمى RESET) وهي فعالة بالواحد أما نفس الإشارة لدى المعالج 8085A (والتي تسمى 'Reset in) هي فعالة بالصفر.

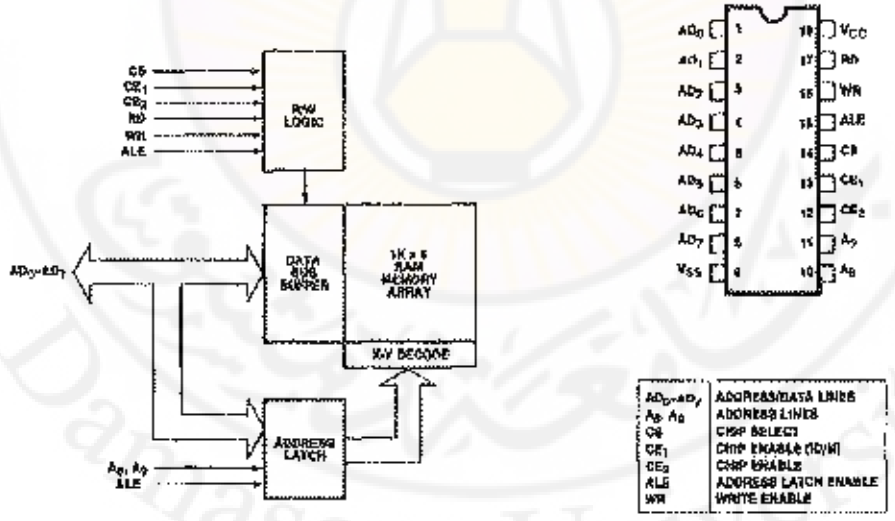
- RD' : تسمية جديدة لدى المعالج 8085A لأمر القراءة عوضا عن الإشارة DBIN لدى المعالج 8080A ويوجد أيضاً تغير في مستوى تفعيل الإشارة.

بقية الإشارات التي تعرفنا عليها لدى المعالج 8080A والمستخدمة مع المعالج 8085A (مثل HOLD و HLDA) لها نفس المفهوم بدون أي تغيير.

وتجدر الإشارة هنا أيضاً إلى أن للمعالج 8085A بعض دارات الدعم الخاصة به، كما أن معظم دارات الدعم للمعالج 8080A تصلح للاستخدام معه أيضاً.

٣-٧ دارات الدعم الخاصة بالمعالج 8085A

أنتجت شركة إنتل مع المعالج 8085A بعض دارات الدعم الخاصة به مشكلة ما يعرف بـ MCS-85. بحيث إن مثل هذه الدارات تستطيع التعامل مع ممرات مركبة بالعناوين والمعطيات. ومن بين هذه الدارات غير المبرمجة، نشير للدارة 8185 التي تمثل ذاكرة من نوع RAM بسعة 1 KB منظمة بالشكل 1024 x 8 bit. تصنع الذاكرة بالتغليف DIL-18 وهي من نوع استاتيكي وتعمل بتغذية واحدة. تملك الذاكرة ثلاثة خطوط انتخاب بالرموز CS', CE1', CE2 يظهر الشكل 3-7 التركيبة العامة لهذه الذاكرة.

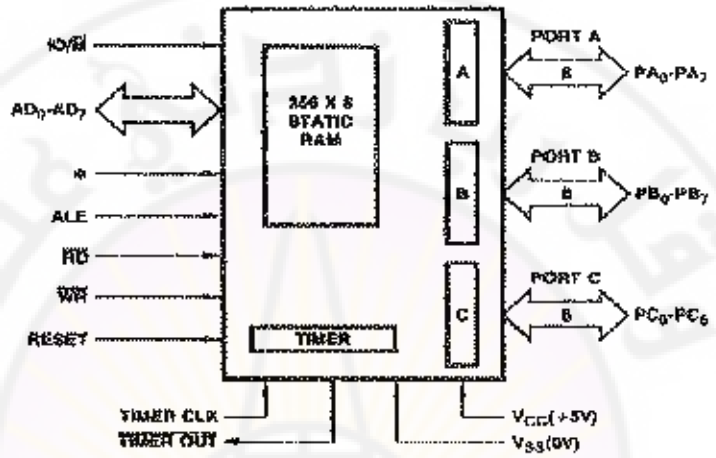


الشكل3-7: المخطط الصندوقي وتوزع الإشارات للدارة 8185

ومن بين دارات الدعم الخاصة بالمعالج 8085A (وغيره من المعالجات الأحدث المماثلة له من حيث وجود المعطيات والعناوين على نفس الممر) نتعرف بعض الدارات المبرمجة.

٧-٣-١ الدارة 8155/8156

يبين الشكل 4-7 التركيبة العامة (الداخلية والخارجية) للدارة 8155 مع مجمل إشاراتها.



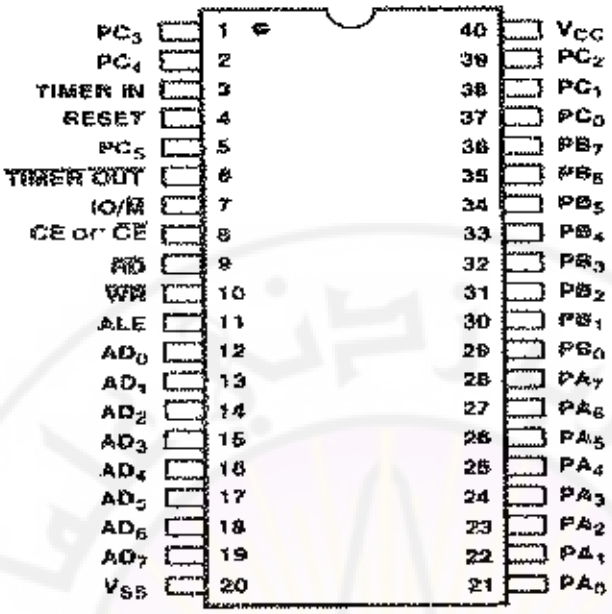
شكل4-7: المخطط الصندوقي للبنية العامة للدارة 8155

تعد الدارة 8155 (2048 bit static RAM with I/O ports and Timer) من دارات الدعم للمعالج 8085A القابلة للبرمجة وهي تحتوي على نفس الشريحة ذاكرة RAM وبوابات للدخل والخرج وعداد ومؤقت زمني.

تملك الدارة 8155 ممر للمعطيات بعرض 8 bit مركبة مع أول 8 خطوط للعناوين Multiplexed ومجموعة من إشارات التحكم.

تضم الدارة 8155A بداخلها ذاكرة بسعة 1/4 KB من نوع RAM الاستاتيكية منظمة بالشكل 8 bit × 256 word، وتضم كذلك ثلاث بوابات لإدخال المعطيات وإخراجها، منها بوابتين A,B كل منهما بطول 8 bits وبوابة ثالثة C بطول 6 bits. وتحتوي الدارة 8155 أيضاً على عداد ثنائي مبرمج بطول 14 bit يمكن استخدامه كمؤقت زمني.

تصنع الدارة 8155A بالتغليف DIL-40. يبين الشكل 5-7 توزع إشارات الدارة.



الشكل5-7: توزع إشارات 8155

توجد دارة أخرى مماثلة للدارة 8155 هي الدارة 8156 ، حيث إن الفرق بين الدارتين يكمن في أن إشارة الانتقاء للدارة 8155 أي 'CS فعالة بالصفر، بينما إشارة الانتقاء للدارة 8156 فهي فعالة بالواحد وتسمى CE، وبالتالي كل ما يرد عن الدارة 8155 ينطبق على الدارة 8156 باستثناء هذا الفرق.

بما أن الدارة 8155 قابلة للبرمجة فإنها تحتوي بداخلها على مسجل داخلي (غير ظاهر بالمخطط الصندوقي للدارة) خاص لهذه الغاية لحفظ كلمات التحكم. تعمل الدارة 8155 على نقل معطيات متوازية بالنمط المبسط Simple Mode أو بنمط المصافحة Handshaking وهذا يتحدد بمحتوى مسجل الحالة والتحكم للدارة.

تحتاج الدارة 8155 إلى خمسة عناوين داخلية على الأقل، وهذا يتطلب ثلاثة خطوط عنونة لتأمين العناوين اللازمة موجودة على المأخذ AD_0, AD_1, AD_2 ويمكن إظهار توزع العناوين الداخلية للدارة بالجدول 3-7.

الجدول4--7: عناوين المكونات الأساسية لدارة الدعم 8155 للمعالج 8085A

	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Internal Command / Status register	x	x	x	x	X	0	0	0
General purpose I/O port A	.	.				0	0	1
General purpose I/O port B	.	.				0	1	0
Port C general purpose I/O or control	.					0	1	1
Low--order 8bit of timercount	.	.				1	0	0
High 6bits of timer count and 2 bits of timer mode	x	x	x	x	X	1	0	1

تعد الدارة 8155 من حيث العنونة والبرمجة بعكس الدارة 8255 لان إشارة الانتقاء CS التي تقابل العنوان الداخلي الأول مخصصة لمسجل التحكم وليس للبوابة الأولى A كما في الدارة 8255. بوابات الدارة 8155 الثلاثة هي:

PA بعرض 8 bit وتمثل المآخذ PA_0, ..., PA_7

PB بعرض 8 bit وتمثل المآخذ PB_0, ..., PB_7

PC بعرض 6 bit وتمثل المآخذ PC_0, ..., PC_5

نتعرف فيما يلي على محتويات كل من مسجل التحكم Command Register والحالة Status Register للدارة 8155، علما أن العنوان الأول للدارة يكون مشتركاً للمسجلين.

أولا– مسجل التحكم أو الأوامر Command Register

يمكن بمسجل التحكم كتابة الكلمة المطلوبة للتحكم بالدارة ولا يمكن قراءة محتواه. ويتكون مسجل التحكم من 8 لواقط Latches، موزعة وفق التالي:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TM2	TM1	IEB	IEA	C2	C1	PB	PA

البتات الأربعة الأولى من كلمة التحكم (D_0--D_3) تعرف نمط عمل البوابات الثلاثة ونوعها إن كانت دخلا أم خرجاً.

والبتان (D_4, D_5) يحددان فيما إن كان يسمح بمقاطعة البوابتين A, B أو لا يسمح. وآخر بيتين (D_6, D_7) يخصصان للتحكم بالمؤقت الزمني.

يمكن أن يتغير محتوى مسجل الأوامر (يتناوب Altered) في أي وقت باستخدام العنوان xxxx x000 (الذي يشير إلى مسجل التحكم) خلال عملية الكتابة. وخلال عملية برمجة الدارة فإن 0 في أي من البتين PA, PB من بيتات مسجل الأوامر تعني دخل Input للبوابة المقابلة و 1 تعني خرج Output.

أما البتان: C_1, C_2 في كلمة التحكم والمخصصان لتعريف PC لهما أربع توافقيات مختلفة (Alt = C_2C_1) للتحكم بالبوابتين A, B وفق ما هو مبين بالجدول 5--7.

أي أن الدارة من أجل التركيبتين 00 = C_2C_1 و C_2C_1=01 تنقل البيانات كدخل وخرج بالنمط البسيط.

وبالتركيبة C_2C_1=10 تعمل البوابة A بنمط المصافحة والبوابة B بالنمط البسيط كخرج. والتركيبة C_2C_1= 11 تقابل عمل البوابتين A, B بنمط المصافحة.

Damascus University

الجدول5–7: برمجة بوابات الدارة 8155

Pin	Alt0 ~ 00	Alt1 ~ 01	Alt2 ~ 10	Alt3 ~ 11
PC_0			A – INTR	A – INTR
PC_1			A– BF	A– BF
PC_2	Input	Output	A– STB'	A– STB'
PC_3	Port	Port	Port	B– INTR
PC_4			Out	B– BF
PC_5				B– STB'

البت IEA (Interrupt Enable A) من كلمة التحكم يقابل Port A (بحيث إن 0 تعني Disable و 1 تعني Enable). ونفس الشيء بالنسبة IEB الذي يقابل البوابة B والبتان TM1, TM2 مخصصان لـ Timer Command وفق التالي:

00	NOP	أي لا يعمل شيئ No Operation
01	STOP – NOP	يعني NOP لا تعمل شيئاً إذا كان المؤقت لم يقلع بعد. ويكون STOP توقف العّد إذا كان المؤقت يعمل
10	STOP After TC	التوقف STOP مباشرة بعد الوصول إلى نهاية عدّ العدّاد Terminal Count – TC
11	START	الإقلاع، ويعني اشحن النمط وطول العّد وأبدأ مباشرة بعد التعبئة إذا كان المؤقت لم يقلع بعد. أما إذا كان المؤقت قد سبق وأقلع، فانطلق مباشرة بعد الانتهاء من عملية العّد الجارية.

ثانياً– مسجل الحالة Status Register

يختلف مسجل الحالة عن مسجل التحكم في أنه يمكن أن يقرأ محتواه ولا يمكن أن يكتب به. وهو يتكون من 7 لواقط Latches ستة منها (AD_0 – AD_5) عن حالة البوابات وبيت واحد (السابع AD_6) عن حالة المؤقت Timer، والبت الثامن غير مستخدم، حيث إن الصيغة العامة لكلمة الحالة هي:

AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	AD_1	AD_0
X	Timer	INTE B	B BF	INTR B	INTE A	A BF	INTR A

ويمكن قراءة محتوى مسجل الحالة من العنوان xxxx x()() إذ يخصص للبوابة PA ثلاثة بيتات هي (AD_0, AD_1, AD_2) بحيث:

AD_0 ~ INTR A – Port A Interrupt Request

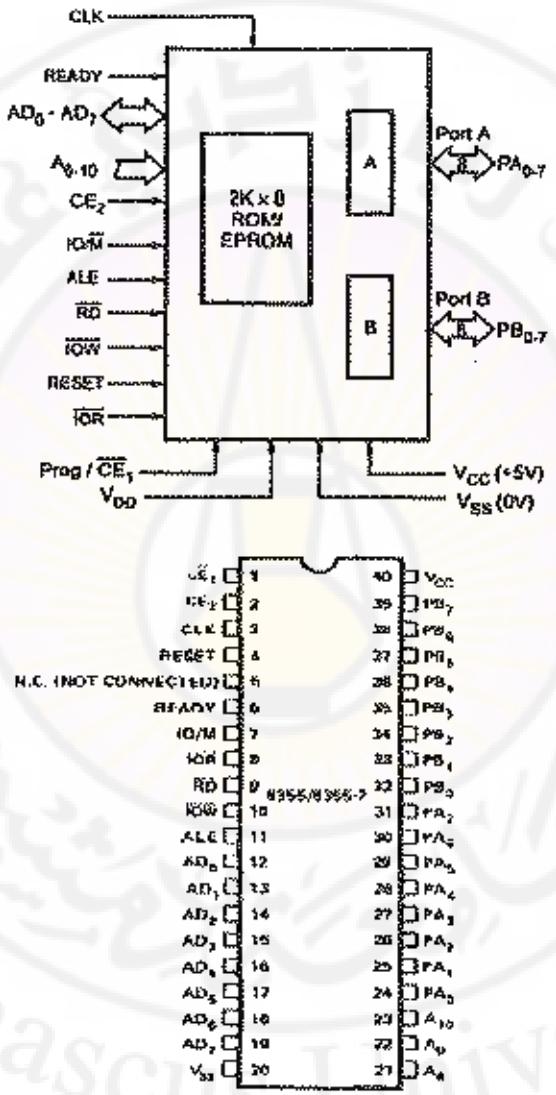
AD_1 ~ A BF – Buffer Full/Empty (Input/output)

AD_2 ~ INTE A – Port A Interrupt Enable

ونفس الشيء يخصص للبوابة PB ثلاثة بيتات AD3, AD4, AD5 بنفس الوظائف.

٧-٣-٢ الدارة 8355

تعّد الدارة 8355 من إحدى دارات الدعم للمعالج 8085 والمعالجات الأحدث المماثلة له بالتركيبة. تصنع الدارة بالتغليف DIL-40. يظهر الشكل 6-7 المخطط الصندوقي للدارة 8355A وتوزع الإشارات، وهي من رتبة VLSI.



الشكل6-7 المخطط الصندوقي وتوزع إشارات الدارة 8355

توجد الدارة 8355 بالنسخة العادية وبالنسخة المحسنة 8355A وبالنسخ الأسرع 8355A-2.

يتضح من الشكل 6-7 أن للدارة 8355 ممر معطيات بطول 8 bit مركب مع 8 bit من ممر العناوين بالاسم AD_7 ,..... ,AD_0 بالتالي يمكن وصلها مباشرة مع المعالج 8085.

تضم الدارة 8355 بوابتين كل منهما بعرض 8 bit هما: PA (PA_7 ,..... ,PA_0) و PB (PB_7 ,..... ,PB_0) لإدخال المعطيات وإخراجها بشكل مبرمج.

تحتوي الدارة على ذاكرة ROM بسعة 2KB بالتنظيم (2K×8bit) بالتالي يلزمها 11 خط للعنونة A_{10} ,..... ,A_0، منها أول ثمانية خطوط A_0 – A_7 مركبة مع ممر المعطيات وآخر ثلاثة خطوط للعنونة A_8, A_9, A_{10} غير مركب معها أي من الإشارات الأخرى وبالتالي هي موجودة بشكل مستقل لعنوية كامل المجال العنوني للذاكرة.

ونشير هنا أيضاً أن للدارة 8355 إشارتي تأهيل هما 'CE1 و CE2 وعند الحاجة للتعامل مع هذه الدارة يجب تأمين السوية الفولطية المناسبة لكل منهما، علما أن 'CE1 هو المأخذ الذي يقابل إشارة Prog عند برمجة الدارة بمعنى أن الجهد اللازم لبرمجة الدارة يطبق على المأخذين 'CE1 و V_{DD}. أما المأخذين V_{cc} و V_{SS} هما لتغذية الدارة بالحالة العادية أثناء ربطها مع المعالج. بالطبع الذاكرة ROM الموجودة بالدارة تأتي بشكل مبرمج من المنتج ولا يمكن اعادة برمجتها من قبل المستخدم بالتالي لادعي للاشارة إلى كيفية برمجتها.

يلاحظ بالنسبة للدارة 8355 أيضاً وجود مأخذ هو الـ pin5 بدون أي وظيفة وبالتالي يترك بدون توصيل ويسمى NC من Not Connected. أما بقية اشارة التحكم بالدارة توصل بالمعالج أو مع دارات الربط المرتبطة معه. بمعنى أن اشارة RESET الدارة مثلا توصل مع اشارة RESET OUT للمعالج وأن اشارة 'RD توصل مع اشارة 'RD المعالج خلال تعامله مع الذاكرة. أما عند التعامل مع البوابات فتستخدم الاشارتين 'IOR و 'IOW من خرج مفكك الترميز بالمداخل 'RD', WR', IO/M.

تملك الدارة 8355 مسجلين داخليين كل منهما بطول 8 bit مخصصين للبرمجة هما: المسجل DDR A لبرمجة البوابة PA والمسجل DDR B لبرمجة البوابة PB والتحكم بها.

بالطبع لا يمكن قراءة محتوى المسجلات DDR وإنما يمكن الكتابة فيها فقط.

تبرمج كل من بوابتي الدارة 8355 على حده، إما كدخل أو كخرج، حيث لا تعمل بنمط المصافحة، يلزم لبرمجة البوابتين PA, PB والمسجلين المعاندين لهما خطي عنونة هما AD_0 و AD_1 لتحديد العنوان وفق ما هو مبين بالجدول 6-7.

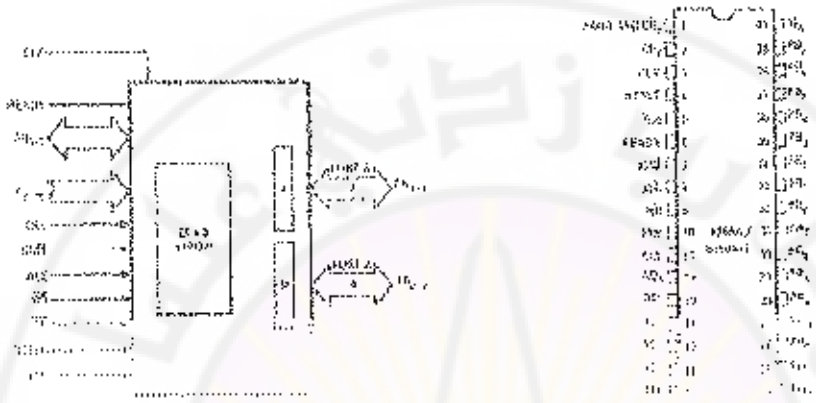
الجدول6-7: برمجة بوابات الدارة 8355

Selection	AD_1	AD_0
PA	0	0
PB	0	1
Port A data direction register (DDR A)	1	0
Port B data direction register (DDR B)	1	1

يقوم مسجلا اتجاه المعطيات DDR بتحديد حالة كل بيت من البوابتين، إن كان بحالة دخل أو خرج، حيث يبرمج كل بيت من البوابتين على حدة. فوجود قيمة "0" بيت محدد من المسجل DDR يشير إلى أن البيت المقابل له من البوابة هو في حالة دخل، فمثلا: وجود القيمة الثنائية 1 0 0 0 0 0 0 1 بالمسجل DDR A تعني أن البتات: الأول والسابع والثامن فقط من البوابة A هي في حالة خرج.

٧-٣-٣ الدارة 8755

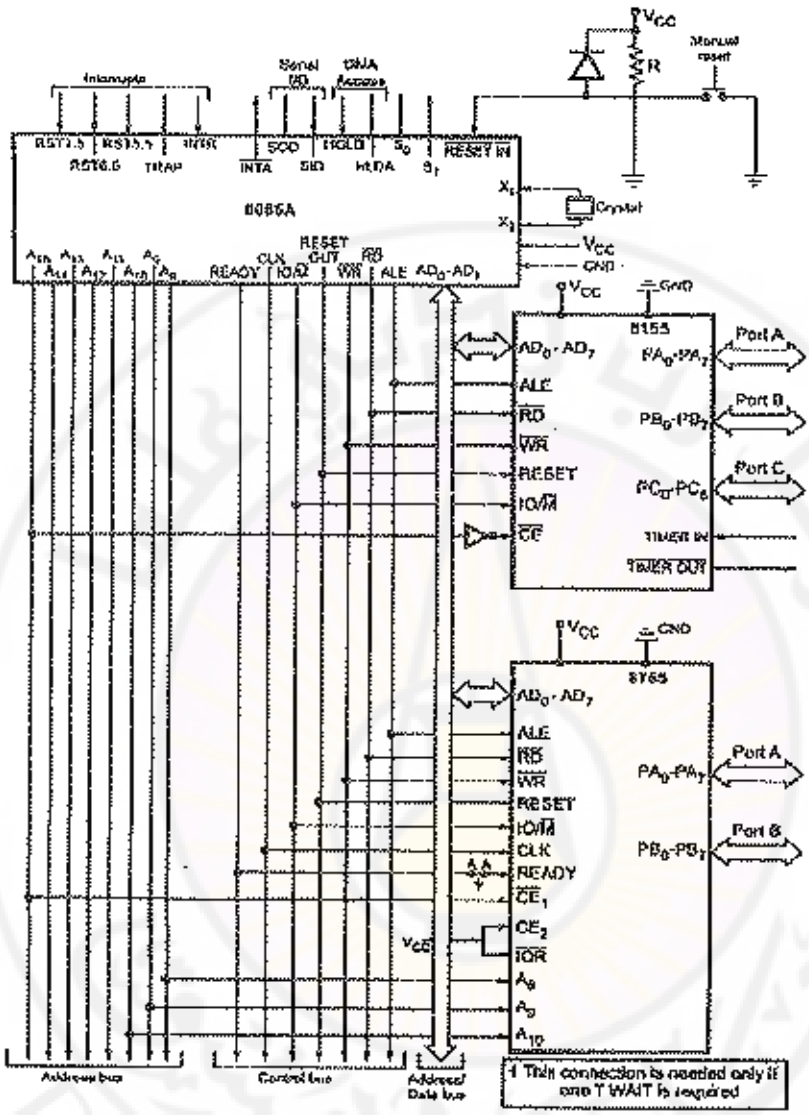
تعدّ الدارة 8755 مماثلة تماماً للدارة 8355 بكل شيء من حيث اسم الإشارات وعددها وعدد البوابات وطريقة البرمجة. يظهر الشكل 7-7 المخطط الصندوقي للدارة 8755 وتوزع الإشارات، حيث إن الـ pin5 مستخدم في هذه الدارة لوصل التغذية V_{DD}.



الشكل7-7: المخطط الصندوقي وتوزع إشارات الدارة 8755

الفارق الجوهري بين الدارتين هو أن الدارة 8755 تحتوي على EPROM بدلا من ROM (وهي بسعة 2KBx8bit أيضاً)، بالتالي يوجد مأخذ خاص لبرمجة الدارة. للدارة 8755 خطي تأهيل مثل الدارة 8355 هما CE_1' و CE_2. يستخدم منهما المدخل CE_1' لبرمجة الدارة. بالطبع الجهد اللازم لبرمجة الدارة يختلف عن جهد التغذية، حيث إن الدارة تغذى بـ 5V أما الجهد اللازم للبرمجة فهو 25V ± 1V.

وبهذا الشكل أصبح بامكاننا اظهار أبسط تركيبة لحاسوب مايكروي باستخدام المعالج 8085 ودارتي الدعم 8755 ,8355 لوجود مايلزم الحاسوب الصغري من ذواكر EPROM و RAM وبوابات يمكن برمجتها كدخل أو خرج موجودة بالدارتين المذكورتين. يظهر الشكل 8-7 هذه التوصيلة التي تصلح للاستخدامات البسيطة.



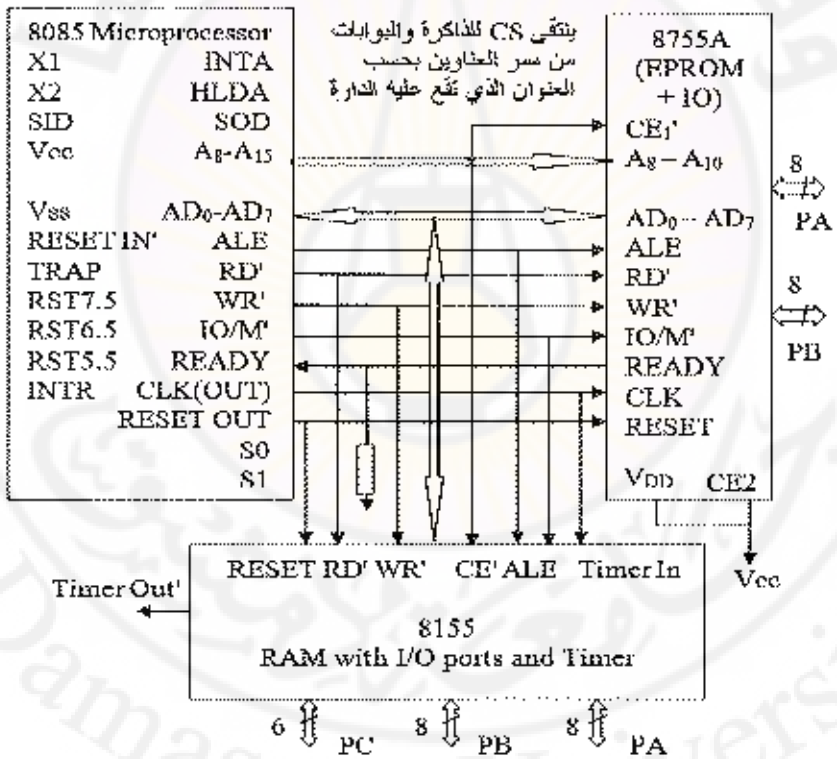
الشكل8-7: التركيبة الصغرى لحاسب مايكروي

٧-٤ أمثلة حول تصميم أنظمة مايكرو كومبيوتر باستخدام المعالج 8085A

نعرض فيما يلي بعض الأمثلة حول تصميم منظومة حاسوبية متكاملة باستخدام المعالج 8085A وبعض دارات الدعم الخاصة به.

مثال7-1: صمم باستخدام المعالج 8085A أبسط منظومة حاسوب صغري.

الجواب: بالنظر إلى أن التركيبة الأساسية لأي حاسوب صغري تضم وحدة معالجة مركزية (معالج صغري) وذاكرة (بنوعيها RAM و ROM) وبوابات (للدخل والخرج)، يمكننا باستخدام ثلاث دارات فقط هي: المعالج الصغري 8085A ودارتين من دارات الدعم الخاصة به 8155 و8355 (أو 8755)، تصميم منظومة حاسوبية متكاملة وفق ما هو مبين بالشكل 9-7 وهي أبسط تركيبة.



الشكل9-7: نظام مايكرو كومبيوتر متكامل بثلاث دارات متكاملة

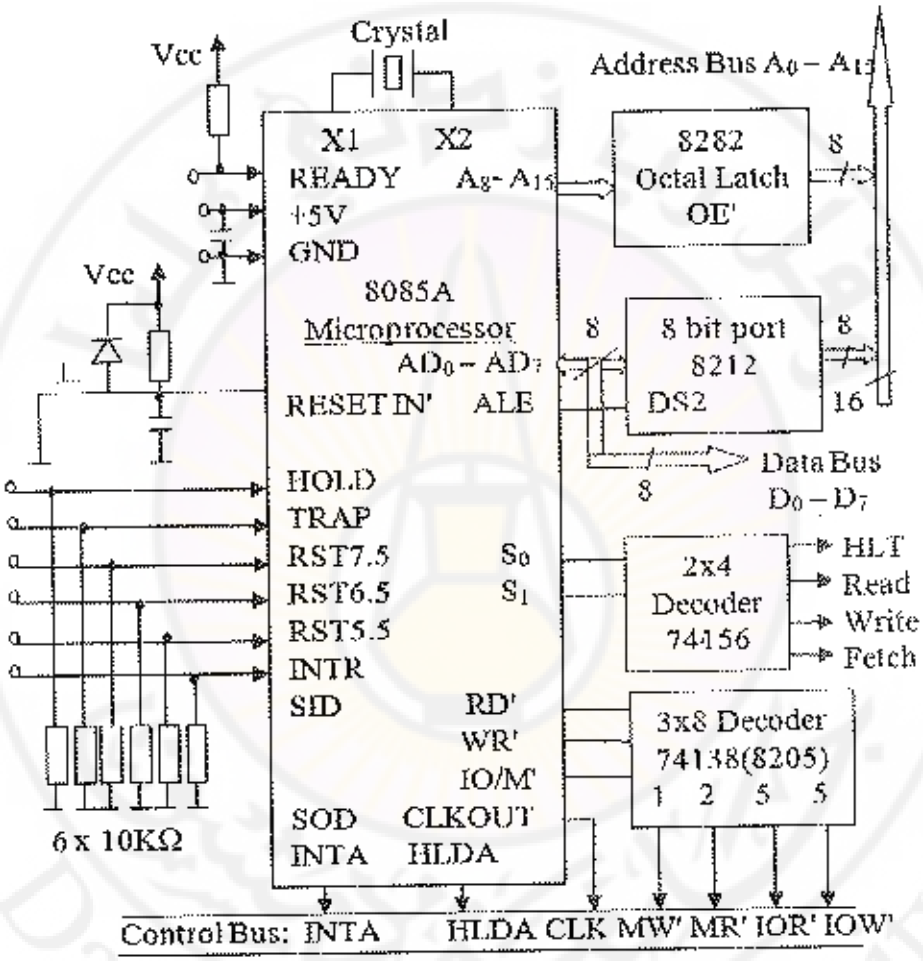
يضم المخطط بالإضافة للمعالج 8085A ذاكرة من نوع EPROM بسعة 2 KB (موجودة بدارة الدعم 8755) وذاكرة RAM من النوع الاستاتيكي بسعة 1/4 KB (موجودة بالدارة 8155) وخمس بوابات اثنتان منها موجودة بالدارة 8755 كل منهما بعرض 8 bit وثلاث بوابات بالدارة 8155 اثنتان منها بعرض 8 bit لكل منهما وثالثة بعرض 6 bit. يمكن استخدام أي من البوابات المتوفرة للنقل بالنمط البسيط كمدخل أو خرج أو للنقل بنمط المصافحة. إشارات المدخل التي لم ترسم بالمخطط، تعرف بأسلوب مماثل لما تم مع المعالج 8080A وإشارات الخرج تؤخذ لممر التحكم بحسب الحاجة.

مثال 2-7: صمم نفس منظومة المايكرو كومبيوتر المبينة بالمثال 1-5 من الفصل السابق، لكن باستخدام المعالج الصغري 8085A بدلا من المعالج 8080A (أي منظومة تضم بالإضافة للمعالج على ذاكرة من نوع EPROM بسعة 4KB وذاكرة RAM بسعة 4 KB مع بوابتي دخل وبوابة خرج، كل منها بعرض 8bit).

الجواب: نبين بالجزء الأول من التصميم رسم دارة المعالج 8085A بكل إشاراته لتشكيل الممرات الثلاثة AB, DB, CB وفق ما هو مبين بالشكل 10-7. حيث يتضح من المخطط أهم الفوارق بالتصميم بين معالجين (8080A و 8085A) كل منهما بطول كلمة 8 bit ومن نفس الشركة المنتجة ، إذ أن أول ما يلفت الانتباه هو عدم الحاجة لاستخدام الدارتين 8224 و 8226 مع المعالج 8085A لأنه بالنسبة للمعالج 8085A يكفي وصل الكريستال بشكل مباشر لتأمين الساعة اللازمة لعمله. كما أنه لا حاجة لتوليد كلمة الحالة للمعالج 8085A إذ إن الإشارات الرئيسية لممر التحكم يمكن توليدها باستخدام مفكك ترميز من إشارات المعالج IO/M', WR', RD'.

الشيء الإضافي لدى المعالج 8085A بالمقارنة مع المعالج 8080A هو عملية فصل ممر المعطيات عن ممر العناوين باستخدام إشارة ALE ، لنتمكن مع المعالج 8085A من استخدام دارات من غير دارات الدعم المخصصة له، لأنه باستخدام دارات دعم المعالج 8085A لا حاجة لفصل المعطيات عن العناوين، حيث إن كافة دارات الدعم الخاصة به تملك ممرات للمعطيات والعناوين مركبة مع بعضها. وقد تم إظهار عملية ترميز إشاراتي

الحالة S_0, S_1 لإظهار أربع حالات للمعالج، وبعد أن تم تحديد كل من ممر العناوين للمعالج 8085A بعرض 16 bit وممر المعطيات بعرض 8 bit بشكل مستقل لكل منهما يصبح الجزء الثاني من التصميم هو نفس التصميم بالنسبة للمعالج 8080A المبين بالشكل 12-5.



الشكل10-7: توصيل المعالج 8085A لتشكيل الممرات AB, DB, CB

مثال 3-7: صمم نظام مايكرو كومبيوتر باستخدام المعالج الصغري 8085A ودارات الدعم الخاصة به، بحيث يحتوي على 2KB ذاكرة من نوع EPROM وذاكرة RAM بسعة بقية المجال العنوني للمعالج، على أن لا يكون فراغ بالمجال العنوني للمعالج مع بوابتي دخل بالنمط البسيط وبوابتي خرج بنمط المصافحة.

مثال4-7: صمم منظومة مايكرو كومبيوتر متكاملة باستخدام معالج صغري بطول كلمة 8 bits تضم بالإضافة للمعالج ذاكرة من نوع EPROM بسعة 12 KB مثلا وذاكرة من نوع RAM بسعة 40 KB على أن لا يكون هناك أي فراغ في المجال العنوني للمعالج إضافة إلى بوابة واحدة بعرض 8 bits للنقل بالاتجاهين وبوابة دخل للنقل بنمط المصافحة وبوابتي خرج إحداها تعمل بنمط المصافحة والثانية تعمل بالنمط البسيط وتقع على العنوان 77H. ويطلب تحديد مجال العنوان لكل Chip في التصميم؟

الجواب: سنحاول فيما يلي الإجابة عن المسألتين معا نظرا للتشابه بينهما، ولان مبدأ التصميم واحد، والفرق الأساسي بين المثالين هو أنه في المثال 3-7 تم تحديد نوع المعالج ودارات الدعم، أما في المثال 4-7 لم يتم تحديد نوع المعالج أو دارات الدعم بل تم فقط تحديد طول كلمة المعالج. وبالتالي يمكننا النظر إلى التصميم بأنه مؤلف من جزأين هما:

الأول: رسم دارة المعالج وتحديد سوية تعريف كل من إشاراته (وبالأخص إشارات الدخل) لتشكيل الممرات الثلاثة الأساسية له، أي AB, DB, CB.

الثاني: تحقيق المتطلبات من التصميم من حيث تأمين حجوم الذواكر وعدد البوابات المطلوبة وربطها بالشكل الصحيح مع ممرات المعالج الثلاثة.

وبالنظر إلى أن الخيارات أمامنا محدودة إذ اقتصرنا بالاختيار على معالج من معالجات إنتل (أي اختيار المعالج 8080A أو المعالج 8085A)، سنفترض أنه تم اختيار المعالج 8085A وسنركز على المتطلبات الواردة بالمثال 4-7 دون التقيد بدارات الدعم الخاصة بالمعالج 8085A، إذ يمكننا استخدام بعض داراته وما يتوفر من دارات دعم. الجزء الأول من التصميم يمكن أن يكون نفس الرسم بالشكل 9-7، أما بالنسبة للجزء الثاني فمن المهم الانطلاق مما هو متوفر من دارات، وفي حال توفر كل ما يطلب من دارات،

يتركز الاهتمام على انتقاء الدارات المناسبة لتتوافق مع بعضها بحسب ما هو مطلوب في التصميم ، لان أي تغيير بنوع الدارات التي ستستخدم ، ينعكس على الشكل العام للتصميم، خاصة وأن خيارات انتقاء الدارات أصبحت واسعة جدا.

وبالعودة إلى المثال، ننطلق من أن مجمل السعة المطلوبة للذواكر بنوعيها من EPROM و RAM هو 52 KB أي أنه رقم غير قياسي. يمكننا تحديد الخيارات باقتصارنا على اختيار ذواكر بطول كلمة قياسي، أي من الشكل (X KB x 8 bit). وبما أن السعة المطلوبة من ذواكر EPROM هي 12 KB فإن الخيارات من الدارات التي سبق واشرنا إليها هي: 6 قطع من الدارة المتكاملة 2716 بسعة 2 KB لكل منها أو 3 قطع من الدارة 2732 بسعة 4 KB لكل منها أو قطعتين من الدارة 2764 بسعة 8 KB لكل منها وهكذا. لكن قبل تحديد أي من الدارات السابقة، ننظر إلى الحجم المطلوب من ذاكرة RAM وهو في هذا المثال 40 KB. ومن الدارات التي نعرفها وتناسب هذا الحجم لدينا خيار 20 قطعة من الدارة 6116 بسعة 2KB لكل منها أو 5 قطع من الدارة 6264 بسعة 8 KB لكل منها. بالطبع اختيار الدارة 6264 هو الأنسب في هذا المثال وبالتالي من الأفضل اختيار ذاكرة EPROM بحجم مماثل، أي الدارة 2764.

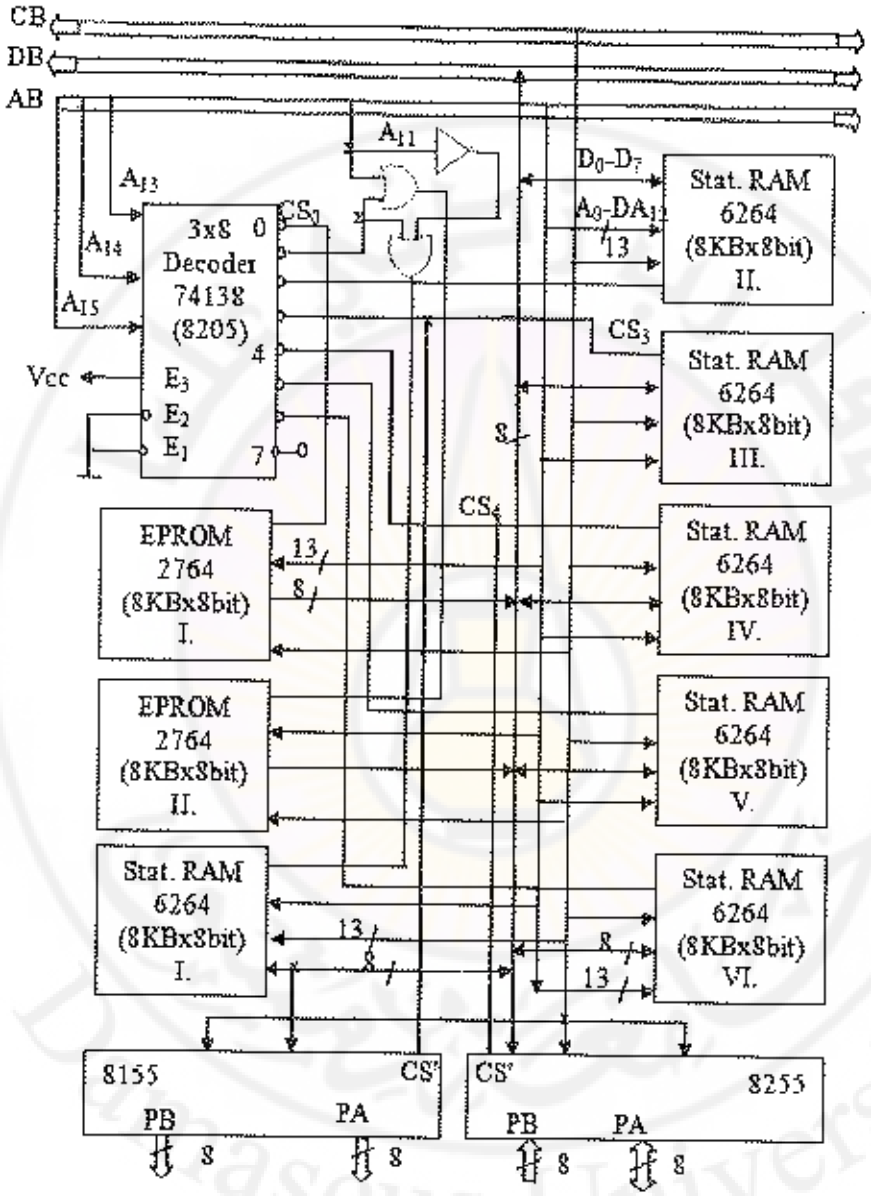
بعد الانتهاء من انتقاء الذواكر، نعود للتفكير بنوع البوابات بحسب نمط العمل المطلوب لكل بوابة على حده. وبما أنه من بين المطلوب يوجد بوابات للنقل بالاتجاهين أو للنقل بالمصافحة لابد من استخدام دارات مبرمجة.

يظهر الشكل 11-7 احد الأشكال الممكنة لهذا التصميم وفيه يستخدم ثماني دارات اثنتان من نوع EPROM و6 من نوع RAM، بحيث يستخدم من ذواكر RAM أربع دارات بكامل السعة ودارتان يستخدم من كل منهما 4 KB فقط.

يلزم ثلاث خطوط عنونة (من A_{13} وحتى A_{15}) لعنونة الدارات السبع، وبذلك يلزم استخدم مفكك ترميز 3x8 Decoder أي بمعدل خرج واحد لكل دارة.

وبما أن الحجم المطلوب من ذواكر EPROM هو غير قياسي، يلزم استخدام دارتان إحداها بشكل كامل والثانية يلزم نصفها فقط ويحدث فراغ في المجال العنوني بين

EPROM و RAM بالتالي لتجاوز مشكلة الفراغ في المجال العنوني للمعالج، يجب ربط أحد مخارج مفكك الترميز بدارتان بآن واحد (أي على نفس الـ Chip Select).



الشكل 11-7: نظام مايكرو كومبيوتر يوافق المثال 4-7.

الأولى من نوع EPROM لعنونة نصفها الأول والثانية من نوع RAM لعنونة نصفها الثاني باستخدام منطق إضافي يجعل المعالج ينتقل بعد 12 KB من ذاكرة EPROM (التي يجب وضعها اعتبارا من العنوان 0000H) إلى ما هو مطلوب من ذواكر RAM. يظهر الشكل 12-7 الذي يمكن للمعالج أن ينتقل عنده بعنونة النوعين المختلفين من الذواكر والمنطق الذي يجب اعتماده لتحقيق ذلك.

$A_{15}A_{14}$ A_{13}	A_{12} $A_{11}A_{10}A_9A_8$	A_7...A_4	A_3...A_0	Addr.	For
0 0 0	0 0 0 0 0	0 ...0	0 ...0	0000H	EPROM
0 0 0	1 1 1 1 1	1 ...1	1 ...1	1FFFH	Ch. 1
0 0 1	0 0 0 0 0	0 ...0	0 ...0	2000H	EPROM
0 0 1	0 1 1 1 1	1 ...1	1 ...1	2FFFH	Ch. 2
					+
0 0 1	1 0 0 0 0	0 ...0	0 ...0	3000H	RAM
0 0 1	1 1 1 1 1	1 ...1	1 ...1	3FFFH	Ch.1
0 1 0	0 0 0 0 0	0 ...0	0 ...0	4000H	RAM
0 1 0	1 1 1 1 1	1 ...1	1 ...1	5FFFH	Ch.2
.....	.....	.....	.....	.....	.....
1 1 0	0 0 0 0 0	0 ...0	0 ...0	C000H	RAM
1 1 0	1 1 1 1 1	1 ...1	1 ...1	DFFFH	Ch.7

الشكل12-7: آلية إلغاء الفراغ في المجال العنوني للمعالج

أي أن أول خط انتقاء CS_0 بالعنوان من 0000H وحتى 1FFFH بحجم 8 KB يخصص لدارة EPROM الأولى. أما خط الانتقاء الثاني CS_1 بوزن 8 KB يقسم إلى قسمين هما:

الأول: بالعنوان من 2000H وحتى 2FFFH أي بحجم 4 KB يخصص للدارة الثانية من EPROM لعنونة نصفها الأول على أن يترك النصف الثاني بدون استخدام.

الثاني: بالعنوان من 3000H وحتى 3FFFH أي بحجم 4 KB يخصص للدارة الأولى من RAM لعنونة نصفها الثاني على أن يترك النصف الأول بدون استخدام. أي وكأن داراتان من 2764 و 6264 فيزيائيا موجدتان فوق بعضهما البعض، أما منطقيا فيتم التعامل مع النصف الأول من 2764 والنصف الثاني من 6264.

تحدث عملية انتقال العنونة من EPROM إلى RAM عندما يصبح A_{13} بالقيمة 1 من مجال CS_1. وهكذا نكمل بعنونة بقية دارات RAM بمعدل خرج واحد من مفكك الترميز لكل من دارات RAM المتبقية باستثناء آخر دارة التي يستخدم منها 4 KB ولا يوجد مشكلة بعنونة قسم من الذاكرة غير موجود إذا كان ذلك الجزء في نهاية الذاكرة.

أما بخصوص البوابات فيتم التعامل معها كما تم التعامل بالمثال 2-5. ونكتفي بعد تحديد نوع الدارات بتحديد كلمة الحالة المناسبة لنمط عمل كل بوابة على حدة. وبقصد التنويع سنستخدم بهذا المثال إحدى دارات الدعم للمعالج 8085A ولتكن الدارة 8155 ودارة أخرى من دارات دعم المعالج 8080A التي تصلح أيضاً للاستخدام مع المعالج 8085A ولتكن الدارة 8255.

يمكن تخصيص الدارة 8255 لتضم البوابة التي تنقل البيانات بالاتجاهين والبوابة التي تعمل كمدخل بنمط المصافحة والدارة 8155 لتضم بقية البوابات المطلوبة. بالتالي يمكن برمجة الدارة 8255 بالـ Mode2 وفيه يمكن أن تكون البوابة PA للنقل بالاتجاهين وبهذه الحالة يلزمها خمس إشارات من البوابة PC ويمكن برمجة البوابة PB كمدخل للنقل بنمط المصافحة، حيث يلزمها ثلاث إشارات قيادية يمكن أخذها من PC. وبهذا الشكل تكون كلمة الحالة التي تهيىء الدارة للعمل (الأولى) على النحو التالي:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	d	d	d	1	1	1

حيث إن القيمة 1 ببعض البتات تعني دخلا والصفر يعني خرجاً، وأن الرمز d بالكلمة يعني Don't care (أي يمكن أن يأخذ أي من القيمتين 0 أو 1). ويمكن أن تكون كلمة الحالة للدارة التي ترسل من المعالج لبرمجتها هي أي كلمة ضمن المجال C7 وحتى FFH

والتي توضع ابتداء من البت السابع D_7 كالتالي:

$D_7 = 1$ هذا يعني كلمة الحالة الأولى بالنسبة للدارة 8255 التي تملك كلمتي حالة.

$D_6D_5 = 1d$ وتعني الـ Mode2 والذي فيه البوابة PA تنقل المعطيات بالاتجاهين

$D_4D_3 = dd$ هي تخصص عادة لبرمجة البوابة PA والقسم العلوي من البوابة PC ($PC_4 - PC_7$) ، وبما أن البوابة PA تنقل بالاتجاهين أي In و Out لاتهم القيم التي توضع بهذين البتين.

$D_2 = 1$ يخصص هذا البت لتحديد الـ Mode الذي تعمل به البوابة PB، وقد تم وضع 1 بهذا البت للدلالة على أن البوابة تعمل بنمط المصافحة (Mode1).

$D_1D_0 = 11$ وهي تخصص عادة لبرمجة البوابة PB والقسم السفلي من البوابة PC ($PC_0 - PC_3$) لذلك تم وضع 1 في كل منهما للدلالة على أن البوابة PB دخل. وبنفس الأسلوب يمكن كتابة كلمة الحالة للدارة 8155 لبرمجتها بحسب الطلب. من طرف آخر، بما أن عنوان البوابة للنقل بالنمط البسيط محدد مسبقا وهو 77H يجب أن تقع الدارة 8155 التي تضم هذه البوابة على خط الانتقاء CS_3 وفق ما هو مبين.

Address	$A_{15}A_{14}A_{13}$ A_{12} A_{11} $A_{10}A_9A_8$ A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	CS_1
77H	0 1 1 1 0 1 1 1	8155

ويمكن وضع الدارة 8255 (التي لم يحدد عنوانها) مثلا على CS_4

مثال5-7: صمم منظومة مايكرو كومبيوتر متكاملة باستخدام المعالج 8085A وما يلزم من دارات الدعم الخاصة به على أن يحوي ذاكرة من نوع EPROM بسعة 2 KB مثلا وذاكرة من نوع RAM بسعة 4 KB، على أن لا يكون هناك أي فراغ في المجال العنواني للمعالج إضافة إلى بعض بوابات الدخل والخرج.

الجواب: بما أنه تم في هذا المثال تحديد نوع المعالج، يجب التقيد بذلك ورسم الجزء الأول من التصميم باستخدام المعالج 8085A كما ورد معنا سابقاً. وسنكتفي في هذا المثال بالاشارة إلى الجزء الثاني من التصميم واستخدام دارات دعم خاصة بالمعالج 8085A.

حيث يلزم دارات متنوعة، مثل الدارة 8185 لاستخدام مايلزم من ذواكر RAM (أو استخدام ذاكرة RAM الموجودة بالدارة 8155) واستخدام ذاكرة EPROM الموجودة بالدارة 8755، إذ يلزم لتأمين ما هو مطلوب من ذواكر EPROM دارة واحدة 8755 توضع على العنوان 0H ويلزم 16 دارة من دارات 8155 لتأمين ما هو مطلوب من ذواكر RAM، لأن كل دارة تحوي على ربع كيلو فقط وهذا قد لايكون عملي، لذلك سنستخدم أربع دارات 8185 كل منها بسعة 1 KB توصل ابتداء من العنوان 800H لأن الـ 2 KB الأولى المخصصة للذاكرة EPROM تنتهي عند العنوان 7FFH. سنصادف هنا مشكلة في التصميم تتمثل باختلاف المجال العنوني للذواكر المستخدمة، حيث أن الذاكرة EPROM بالدارة 8755 بسعة 2KB وبالتالي يلزم لعنونتها 11 خط عنوان من A_0 وحتى A_{10} بينما الذاكرة RAM بالدارة 8185 هي بسعة 1 KB ويلزم لعنونتها 10 خطوط عنونة من A_0 وحتى A_9. بهذه الحالة نجعل خرج الـ Decoder بوزن المجال الأصغر، أي 1 KB، لتخصص خرجين متتالين من الـ Decoder للدارة 8755 وخرج واحد لكل دارة 8185 تجنبا لاحداث أي فراغ في المجال العنوني للمعالج. ويمكننا من الدارة 8755 تأمين بوابتين على الأقل للدخل أو للخرج بحسب الطلب (بالطبع إذا طلب عدد أكثر من البوابات يفترض اضافة دارات أخرى).

٧-٥ أسئلة متعددة الاجابات

1- بالنسبة للمعالج 8085 يحجز العنوان 24H في الذاكرة لتخديم المقاطعة:

a- REST5.5 b- TRAP c- REST7.5 d- INTER

2- التوافقية S1S0 = 11 بالنسبة لإشارات الحالة لدى المعالج 8085 تعني

a- Halt b- Read c- Fetch d- Write

3- عدد المسجلات التي طول كل منها 8 bit لدى المعالج 8085 هو

a- 7 Registers b- 8 Registers

b- 14 Registers d-16 Registers

4- عدد المسجلات التي يمكن أن يكون طول كل منها 16 bit لدى المعالج 8085 هو

a- 2 Registers b- 8 Registers

c- 6 Registers d- None of all

5- عندما يكون ALE فعالا (يساوي الواحد) في المعالج 8085 فإن:

a – المعطيات على الخطوط A0--A7 تكون Address

b- المعطيات على A0--A7 تكون Data

c – المعطيات على الخطوط A0--A16 تكون Address

d – المعطيات على A0--A16 تكون data

6- يدعم المعالج 8085:

a- 2^8 I/O Ports b- 16 I/O Ports

c- 2^{16} I/O Ports d- 8 I/O Ports

7- يقلع المعالج 8085 بعد تطبيق إشارة RESET

a- من آخر عنوان للذاكرة FFFFH

b- من العنوان التالي للعنوان الذي طبق فيه RESET

c – من أول عنوان للذاكرة 0000H

d - من عنوان يختاره المعالج بحسب وضعه

8- يختلف المعالج 8085 عن المعالج 8080 بـ :

a- بوجود كتلة إضافية للنقل التسلسلي

b- بتركيب خطوط المعطيات على خطوط العنونة

c – بتطوير نظام المقاطعة

d - بكل ما سبق

9- المسجل بعرض 8 bit لدى المعالج 8085 يكون

a- SP b- HL c- PC d- None of all

10- بمعالجات Intel التي كلمتها بطول 8 bits (مثل المعالجين 8085 ,8080) اسم المسجل الذي طول كلمته 16 bits هو:

a. Stack pointer b. Accumulator

c. Program counter d. a&c

11- العنوان المخصص للمقاطعة RST6.5 لدى المعالج 8085 هو

a. 002CH b- 0024H c- 0034H d- 003CH

12- أي من المقاطعات التالية يمكن أن يكون غير قابل للتقنيع non-Maskable ?

a. INTER b. RST5.5 c.TRAP d. INT 21H

13- يمتاز النقل التسلسلي للمعطيات عن النقل التفرعي بـ:

a- النقل لمسافات بعيدة b- السرعة العالية

c- قلة الأخطاء في النقل d- كل ما سبق

14- أي من دارات الدعم لمعالجات Intel التالية هي من النوع المبرمج؟

a. 8755 b- 8155 c- 8355 d- 8212

15- الدارى 8155 عالية التكامل (من رتبة VLSI) تمثل دارة دعم للمعالج الصغري

a- 8080 only b- 8085 only

c- 8080 & 8085 d- None of all

16- الدارى 8251 عالية التكامل (من رتبة VLSI) تمثل

a- UART b- Programmable interrupt controller

c- USART d- Programmable interval timer/ counter

17- عندما تصفر الدارة 8255 تصبح كافة بواباتها للدخل والخرج مهيئة كا

a- Output port using mode 0 b- Input port using mode1

c- Output port using mode 1 d- Input port using mode 0

الفصل الثامن: معالجات 16 bits

ظهر بعد تأخير غير طويل من ظهور معالجات 8 bit المعالجات بطول 16 bits من شركات عدة كان من أهمها المعالج الصغري 8086 من شركة Intel والمعالج Z800 من شركة Zilog والمعالج MC68000 من شركة Motorola وغيرها، حيث إن الشركات الثلاثة المشار إليها أعلاه هي الأكثر شهرة في مجال تصنيع المعالجات، وتمتاز معالجات كل من هذه الشركات ببعض الميزات الخاصة بها (إذ إن كل شركة من الشركات المصنعة للمعالجات تركز على إحدى الجوانب الأساسية للمعالج من حيث التعليمات، التعامل مع الذاكرة، منظومة المقاطعة وغيرها). فمثلا معالج MC68000 له جملة التعليمات الأكثر فعالية. أما معالج Z800 فله القدرة على عنونة مجال ذاكرة أكبر يصل إلى 8MB. بينما معالج 8086I يعدّ الأكثر عمومية أو شعبية للتوافق بينه وبين المعالجات الأقدم 8080A و 8085A الواسعة الانتشار بتطبيقاتها المتنوعة.

٨-١ العائلة The 8086 µP Family

ظهر المعالج 8086 من شركة Intel في العام 1978 وبفترة قريبة من هذا التاريخ أنتجت إنتل معالج آخر بالاسم 8088 الذي يمكن تصنيفه مع كل من معالجات 8bit و16 bit لان بنيته الداخلية مماثلة لبنية المعالج 8086 بـ 16bit أما ممر المعطيات DB الخاص به فهو بعرض 8bit فقط.

وقد أنتجت شركة إنتل بالإضافة للمعالجين 8088 ,8086 المخصصين للاستخدامات العامة، معالج آخر بطول كلمة 16 bit لعمليات الدخل والخرج بالاسم 8089. ومن مجمل المعالجات الستة عشرية الأولى من شركة انتل نكتفي بالإشارة للمعالج 8086 كممثل عن المعالجات 8089 ,8088 ,8086 نظرا لكثرة المواصفات المشتركة بين هذه المعالجات الثلاثة. تشترك المعالجات (8086,8088,8089) Intel بالعديد من الأمور التي نذكر منها:

- سرعة أي من هذه المعالجات معطى بالتواتر الساعي 5MHZ (وهذا يقابل دور للساعة بمقدار 200ns بالطبع النسخ الأسرع للمعالج تعمل بتواتر 8MHZ).

– خطوط المعطيات الداخلية بكل من هذه المعالجات الثلاثة هي بعرض 16bit. أما التعامل مع الوسط الخارجي فيتم بممر معطيات بعرض 16bit أو 8bit وهي مركبة Multiplexed بالمعالجات الثلاثة.

– يمتلك كل معالج عشرين خطاً للعنونة، وبالتالي يسمح كل من هذه المعالجات بعنونة ذاكرة بسعة 1MB أو 64 KB للدخل /الخرج.

– تمثل كل من المعالجات الثلاثة دارة أحادية البلورة بتغليف DIL – 40.

تندرج المعالجات الثلاثة تحت اسم عائلة واحدة 8086 Family التي تضم كافة المعالجات الأحدث من إنتل، وقد سميت بهذا الاسم بالنظر للتوافقية Compatibility بينها. ومن أهم أفراد هذه العائلة:

المعالج 80186 الذي يعدّ نسخة محسنة عن المعالج 8086 و

المعالج 80188 الذي يعدّ نسخة محسنة عن المعالج 8088،

ويعدّ المعالج Intel 80286 معالجاً صغريا يتعامل مع 16 bits وهو نسخة متقدمة عن المعالج 8086 وقد صمم خصيصاً للاستخدام كوحدة معالجة مركزية بامكانيات أكبر بالحواسيب الصغرية المتعددة المستخدمين Multiuser أو المتعددة المهام Multitasking.

يعدّ المعالج Intel 80386 معالجاً صغريا يتعامل مع 32 bits ويتمتع بميزات بارزة مقارنة مع المعالج 80286، حيث يمكنه أن يعنون مباشرة حتى 4GB من الذاكرة. أما المعالج Intel 80486 فيعدّ خطوة متطورة عن المعالج 80386. وأخيراً المعالج 80586 الذي يقابل معالج بنتيوم Pentium هو الأحدث والأفضل من كل ما سبق وله إصدارات مختلفة تحت تسميات متعددة.

٨-٢ الأنظمة متعددة المعالجات

صمم المعالج 8086 للعمل بالأنظمة Multi processor System (الأنظمة متعددة المعالجات)، ولازال البعض يخلط بين مفهومي تعدد المعالجات Multiprocessor وتعدد المعالجة Multiprocessing، إذ إن مصطلح تعدد المعالجات يشير إلى البنية التي تسمح بوجود اثنين أو أكثر من وحدات المعالجة (معالجات متعددة) تتقاسم كل من الذاكرة

الرئيسية والأجهزة الطرفية، في حين أن تعدد المعالجة يعني قدرة نظام دعم أكثر من معالج واحد و/أو القدرة على توزيع المهام بينها. ويشير مفهوم تعدد المعالجة إلى تنفيذ عدة عمليات برمجة متزامنة. تمتاز الأنظمة متعددة المعالجات بالمقارنة مع الأنظمة التي تستخدم معالجاً واحداً بالعديد من الايجابيات التي يمكن إيجازها بالتالي:

التخصيص: حيث يمكن تقسيم المطاليب المترتبة على منظومة ما بشكل تخصيصي على المعالجات المستخدمة كل بحسب نوعه، لان بعض المعالجات قد تكون مناسبة لتطبيق محدد أكثر من غيرها من المعالجات الأخرى.

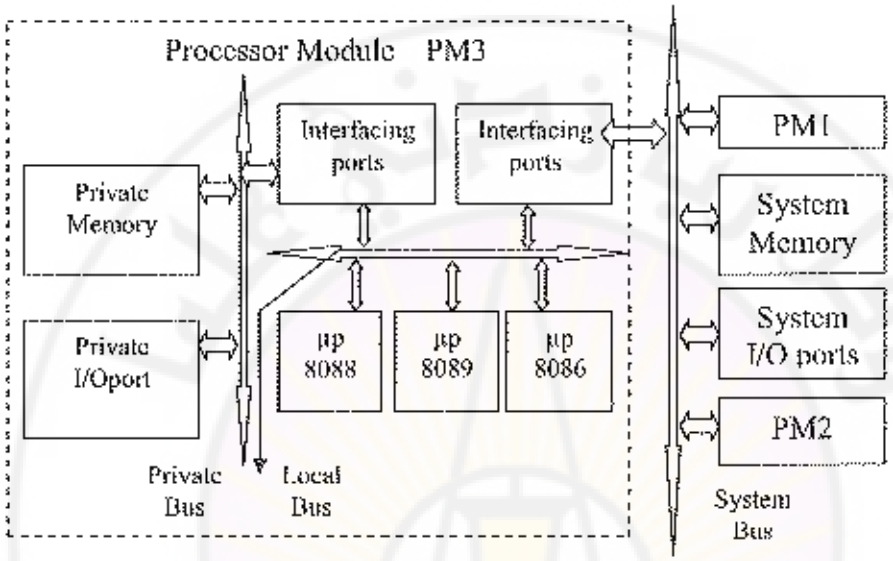
الفعالية: إن فعالية المنظومة متعددة المعالجات تكون أكبر لوجود أكثر من معالج في المنظومة الواحدة، ولأن المعالجات قد تعمل على التوازي في حل مسألة محددة بدلا من حلها بمعالج واحد.

الوثوقية: تكون وثوقية Reliability الأنظمة المتعددة المعالجات أعلى من وثوقية الأنظمة أحادية المعالج لان تعطل أي جزء منها سيكون له أثر محدود وبسيط على بقية أجزاء المنظومة بالنظر إلى إمكانية إنجاز ما يبقى من المسار Process المعالج بما هو متوفر من معالجات أخرى.

التطوير: تقسم عادة المنظومة الواحدة من أنظمة تعدد المعالجات إلى عدة منظومات جزئية كل منها بمعالج خاص وهذا يسهل عملية التطوير وعلى التفرع، أي يمكن بأي وقت إعادة تشكيل النظام أو إعادة تنظيمه بالتركيبة التي تناسب المستثمر.

ويمكن تقسيم المعالجات اعتمادا على مفهوم تعدد المعالجات، إلى نوعين أساسيين:
- المعالجات غير المرتبطة مثل المعالجات 8089 ,8088 ,8086 الذي يمكن لكل منها إنجاز تسلسل التعليمات وفق ورودها ببرنامج معين ذاتياً.
- المعالجات المرتبطة أو التشاركية مثل المعالج المساعد (الحسابي) 8087 الذي يملك تعليمات خاصة به ينفذها بالتنسيق مع المعالج القائد في المنظومة، حيث لايمكنه العمل ذاتيا بمعزل عن المعالج المرتبط معه (المعالج 8087).

يتم تعامل المعالجات في الأنظمة متعددة المعالجات من خلال ممر يدعى ممر متعدد الأسياد Multi-master Bus. تتشكل المنظومة متعددة المعالجات عادة من عدة نماذج معالجة Processor Module (اختصارا PM)، بحيث كل PM يمثل وحدة معالجة مستقلة تضم معالجاً أو أكثر كما هو مبين بالشكل 1-8.



شكل 1-8 المخطط الصندوقي لمنظومة متعددة المعالجات

يظهر من المخطط الصندوقي لمنظومة متعددة المعالجات، وجود عدة أنواع من الممرات (System Bus, Local Bus, Private Bus) ولكل منها استخدام محدد. ويضاف لذلك بالأنظمة متعددة المعالجات ممر غير ظاهر بالمخطط أعلاه، يسمى بممر التحكيم Arbitration Bus الذي يستخدم لتخصيص معالج عن غيره بالممر.

٨-٣ المعالج الصغري 8086/8088

يعدّ المعالج 8086 بالمقارنة مع المعالجات 8080A و 8085A بسبع وحتى عشر مرات أسرع وقد صمم ليستخدم كوحدة معالجة مركزية في الحواسيب الصغرية ويمكنه تشغيل كافة البرامج المعدّة للمعالج 8080A و 8085A بدون صعوبات. يملك المعالج 8086 ما يقارب 29000 ترانزستور في تركيبته الداخلية على الـ Chip الواحد ويصنع بالتقانة HMOS (High-performance MOS) التي تعدّ شكلا معدلا للتقانة NMOS إلا أنها أسرع منها بأربعة أضعاف).

يعمل المعالج بعدة أنماط وهنا نميز بين ثلاثة معانٍ لمصطلح " نمط عمل المعالج" الأول: يتعلق بأسلوب الولوج للمعطيات وهو مفهوم برمجي (عن طرق العنونة) والثاني: يتعلق بكيفية تعامل المعالج مع البرامج والمستخدمين (بيئة التشغيل). والثالث: يتعلق بالتركيبة التي يعمل بها المعالج وهو مفهوم تقني.

٨-٣-١ أشكال (صيغ) عمل المعالج الصغري 8086

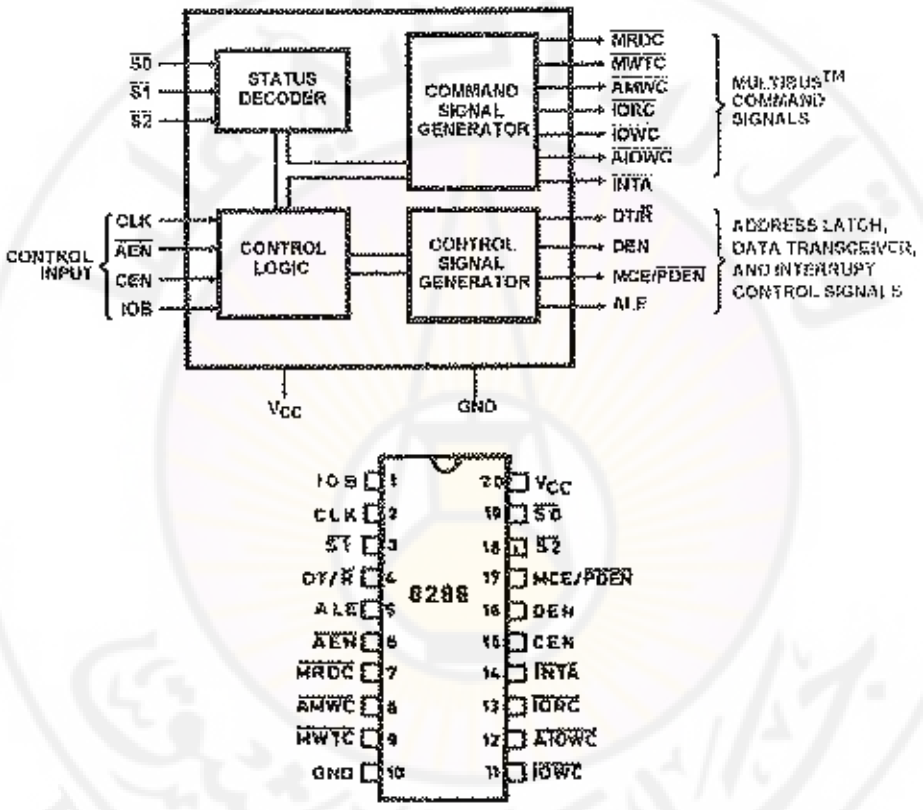
يعمل المعالج 8086 بصيغتين (نمطين وهو المفهوم الثالث لمصطلح نمط عمل المعالج) يحدد أي منهما بالقيمة الموجودة على إشارة 'MN/MX للمعالج، حيث إن:

1 = 'MN/MX تعني الصيغة الدنيا أو النموذج الاصغري Minimum Mode. وتصادف هذه الصيغة في الأنظمة التي تضم معالجاً صغرياً واحداً، وفيها يشرف المعالج بشكل مباشر على تشكيل إشارات التحكم اللازمة للتعامل مع الدارات المرتبطة معه من ذواكر وبوابات، كما كان الحال بالنسبة لمعالجات الـ bit 8.

0 = 'MN/MX تعني الصيغة العظمى أو النموذج الأعظمي Maximum Mode وتخصص هذه الصيغة للعمل بمنظومة من نوع Multiple Bus Master والتي فيها يخصص المعالج لأعمال المعالجة فقط، أما كل ما يخص عملية توليد إشارات التحكم والتزامن والولوج إلى ممر الـ Multi-master تتولاها بعض دارات التحكم الداعمة للمعالج 8086 مثل الدارة 8288 (Bus Controller) و 8289 (Bus Arbiter).

٨٠٣٠١٠١ متحكم الممر (الممر) 8288

نشير باختصار إلى مفهوم المتحكم 8288 الذي سنصادفه بكل منظومة متعددة المعالجات، وذلك بعرض التركيبة الداخلية له واسم كل من إشاراته دون الدخول بمزيد من التفاصيل حولها. يبين الشكل 2-8 التركيبة العامة للمتحكم 8288 وتوزع إشاراته. ويبين الجدول 1-8 اسم كل من إشارات المتحكم 8288 ونوعها إن كانت دخلا أم خرجاً.



الشكل2-8: المخطط الصندوقي لمتحكم الممر 8288 وتوزع إشاراته

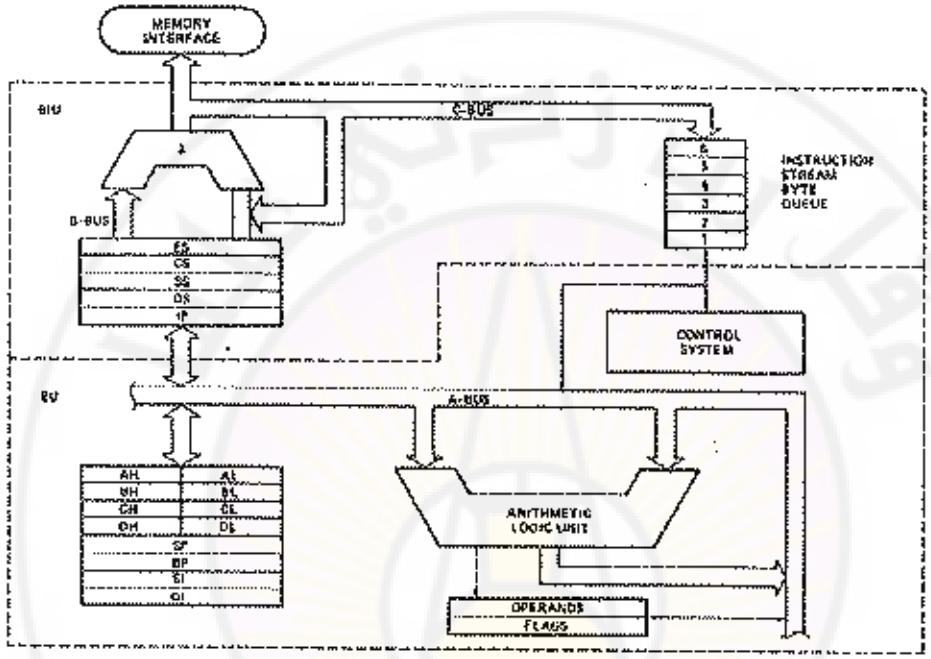
يستخدم متحكم الممر 8288 بـ 20 Pins في أنظمة المعالجات 8086/8088 المتوسطة ويقوم بعرض الأوامر المتعلقة بالقراءة والكتابة من الذواكر والبوابات، كما يقوم بتوليد التحكم الزمني لعمل المعالج. يعمل المتحكم بتغذية وحيدة، حيث إن Ucc = +5V على المأخذ 20 والأرضي على المأخذ 10.

الجدول1-8: مفهوم إشارات متحكم الممر 8288

pin	Name	type	pin	Name	Type
1	IOB- In/Out Bus Mode	In	11	IOWC'- In/out Write Command	Out
2	CLK- Clock	In	12	AIOWC'-Advanced I/O Write Command	Out
3	S1'- Status Input	In	13	IORC'- I/O read Command	Out
4	DT/R'- Data Transmit/Receive	Out	14	INTA'- Interrupt Acknowledge	Out
5	ALE-Address Latch Enable	Out	15	CEN- Command Enable	In
6	AEN'- Address Enable	In	16	DEN- Data Enable	Out
7	MRDC'- Memory Read Command	Out	17	MCE/PDEN'- IOB is tied Low/High	Out
8	AMWC'-Advanced Memory Read	Out	18	S2'- Status Input	In
9	MWTC'- Memory Write Command	Out	19	S0'- Status Input	In

٨-٣-٢ التركيبة الداخلية للمعالج 8086

يظهر الشكل 3-8 التركيبة الداخلية للمعالج 8086 التي تقسم إلى جزأين مستقلين وظيفياً هما EU (The Execution Unit) و BIU (The Bus Interface Unit)، حيث يمكن أن يعمل الجزآن معاً على التوازي.



شكل3-8: التركيبة الداخلية للمعالج الصغري I8086

تسمح الوحدتان EU و BIU للمعالج بالتداخل مابين عملية جلب التعليمة وعملية تنفيذها (وتسمى هذه الظاهرة Pipelining) مما يساعد باستخدام أفضل لممر النظام. تتمثل أهم الفوراق بين المعالجين 8086 و 8088 في أنه بالوحدة BIU طول رتل التعليمات للمعالج 8086 هو 6 Bytes بينما بالنسبة للمعالج 8088 طول الرتل هو 4 Bytes. وأن التجهيزات المحيطية التي طول كلمتها 8 bits ، تربط بممر المعالج 8088 مباشرة، اما بالنسبة للمعالج 8086 فتربط إما على القسم السفلي لممر المعطيات $D_0 - D_7$، وإما على القسم العلوي $D_8 - D_{15}$ (لا فرق بالاختيار).

٨-٣-٢-١ وحدة التنفيذ EU (The Execution Unit)

لا ترتبط وحدة التنفيذ EU مع ممرات النظام، بل هي تستقبل وتخرج كافة المعطيات عبر الوحدة BIU. وهي المسؤولة عن استقبال شفرات تعليمات البرنامج والمعطيات من BIU، بحيث إنها تأخذ التعليمات من رتل خاص والمعطيات من المسجلات العامة وتقوم بتنفيذ التعليمات وتخزن النتائج بالمسجلات. تتكون الوحدة EU من الأجزاء التالية:

1 – وحدة الحساب والمنطق ALU والدارات المرتبطة بها للتحكم بتنفيذ تعليمات البرنامج ومختلف أشكال المعالجة.

2 – مسجل الأعلام Flag Register : يرتبط هذا المسجل مباشرة مع ALU وهو يمثل مسجل بطول 16 bit مستخدم منه تسع بيتات فقط. المؤشر أو " العلم Flag " هو قلاب يستقر بواحدة من قيمتين، ليشير بحسب نوعه إلى حالة المعالج بعد تنفيذ تعليمة ما في وحدة ALU بحيث كل مؤشر يأخذ قيمتين هما: Reset – 0, Set –1. أعلام المعالج 8086 موزعة بمواقع محددة بالمسجل، كما هو مبين بالشكل4–8.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

شكل4–8: مسجل الأعلام لدى المعالج 8086

خمسة من المؤشرات المبينة بالشكل أعلاه مماثلة تماما من حيث التسمية والتوضع لمؤشرات المعالجات بطول كلمة 8 bit التي سبق وتعرفنا عليها وذلك للمحافظة على توافقية معينة بين معالجات إنتل. أضيف للمؤشرات الخمسة لدى المعالجين 8080A و 8085A أربعة مؤشرات جديدة إحداها حسابي وثلاثة تحكمية. والمؤشرات الأربعة الجديدة هي:

- مؤشر التتبع TF: Trap Flag
- مؤشر المقاطعة IF: Interrupt Flag
- مؤشر التوجيه DF: Direction Flag
- مؤشر الطفح OF: Overflow Flag

ترتبط ستة من المؤشرات التسعة للمعالج بالعمليات الحسابية ضمن ALU ، وتسمى بالمؤشرات (الأعلام) الحسابية أو الشرطية. وتسمى المؤشرات الثلاث المتبقية DF,IF,TF بمؤشرات التحكم Control Flags. ويشير الرمز U بالشكل أعلاه إلى إن البيت المقابل غير معَرف Undefined. تختلف أعلام التحكم عن الأعلام الشرطية في طريقة وضعها بواحد (Set) أو تصفيرها حيث إن الأعلام الشرطية الستة ترفع أو تصفر بوساطة EU بناء على نتائج بعض العمليات الحسابية أو المنطقية، بينما ترفع أعلام التحكم أو تصفر بتعليمات محددة يضعها المبرمج في البرنامج، بحيث إن

*- مؤشر التتبع (علم المصيدة) Trap Flag : يستخدم من أجل نمط الخطوات الأفرادية Single--Step في البرنامج (بمعنى أنه إذا كان في حالة Set فإن المعالج ينتقل إلى نمط الخطوة - خطوة لتتبع البرنامج والكشف عن الأخطاء.

*- مؤشر أو علم المقاطعة Interrupt Flag : ويستخدم للسماح بالمقاطعة (إذا كان بحالة الواحد) أو لمنعها (حالة IF = 0).

*- مؤشر أو علم الاتجاه Direction Flag : ويستخدم مع تعليمات السلاسل لتحديد اتجاه البحث من الأعلى إلى الأسفل أو من اليسار لليمين، بمعنى إذا كان: DF =1 فإن التعليمات التي تتعامل مع السلاسل تنقص بشكل آلي مسجلات التأشير وفي حال DF = 0 يتم زيادة محتوى مسجلات التأشير تلقائياً.

3- مسجلات الأغراض العامة General Purpose Register

يمتاز استخدام المسجلات بشكل عام بتوفر إمكانية الولوج السريع إلى محتوياتها بالمقارنة مع الولوج إلى المعطيات المخزنة في ذاكرة خارجية . تمتلك الوحدة EU أربع مسجلات للاستخدامات العامة كل منها بطول 16 bit هي المسجلات AX, BX, CX, DX ولتحقيق التوافقية مع المعالجات الأقدم ذات الكلمة 8 bit يمكن تقسيم كل من هذه المسجلات إلى مسجلين طول كل منهما 8 bit، بحيث يشير الدليل L (Low) إلى القسم السفلي والدليل H (High) إلى العلوي من المسجل. صممت مسجلات المعالج 8086 بهذه الطريقة لتأمين عملية نقل البرامج المكتوبة للمعالجين السابقين 8085A ,8080A.

يمتاز كل من هذه المسجلات العامة بالإضافة للتخزين المؤقت للمعطيات بوظيفة محددة تميزه عن غيره، بحيث إن

- المراكم AX, Accumulator (AX = AH + AL): هو مسجل خاص بطول 16 bit قابل للتقسيم إلى مسجلين كل منهما بطول 8 bit، يملك خصائص وميزات لا يملكها غيره من مسجلات الأغراض العامة الأخرى كاستخدامه في تعليمات الإدخال أو الإخراج.
- المسجل BX, Base Register (BX = BH+ BL): وهو مسجل مخصص للتعامل مع الذاكرة لحفظ عناوين المواقع التي سيتم منها جلب البيانات أو تخزينها.
- المسجل CX, Counter (CX = CH+ CL): وهو مسجل يخصص للاستخدام كعداد يقوم بحفظ القيمة التي سيتم العد إليها عند استخدام التكرارات والحلقات.
- المسجل DX, Data Register (DX = DH+ DL): ويستخدم في بعض الأعمال الخاصة مثل العمليات الحسابية من ضرب وقسمة لحفظ الناتج بالتزافق مع AX.

4 – مجموعة مسجلات المؤشرات Pointer Register:

وتضم مسجلين هما: SP – Stack Pointer و BP – Base Pointer ، حيث المسجل SP هو بطول 16 bit، ويشير محتواه إلى قمة المكدس Top of Stack. أما المسجل المؤشر BP فهو أيضاً بطول 16 bit غير قابل للتقسيم ويشير إلى المعطيات بمقطع المكدس، يستخدم المسجل BP عادة في أنماط العنونة: Register Indirect or Based, Based Indexed

5 – مجموعة مسجلات الدليل Index Register الخاصة بالتعليمات التي تتعامل مع السلاسل Strings، حيث إن EU تملك نوعين من مسجلات الدليل هما:

- مسجل فهرس المصدر SI–Source Index register
- ومسجل فهرس الوجهة DI--Destination Index register

المسجل المؤشر SI هو بطول 16 bit غير قابل للتقسيم ويستخدم من أجل العنونة Indexed, Based Indexed and Register Indirect كعنوان بيانات المصدر بالتعليمات التي تتعامل مع السلاسل. والمسجل DI هو بطول 16 bit غير قابل للتقسيم ويستخدم من أجل العنونة بنفس الأنماط للمسجل SI ولكن كعنوان لبيانات الهدف.

٨-٣-٢-٢ وحدة ربط الممر BIU (The Bus Interface Unit)

تستجيب وحدة ربط الممر BIU للعمليات على الممر الخارجية. إذ تقوم بكل الأعمال المتعلقة بجلب التعليمات والقراءة والكتابة بكل من الذاكرة ووحدات المدخل والخرج. تؤمن BIU ممر معطيات بعرض 16 bit وممر عناوين بعرض 20 bit (أي توليد العناوين الفيزيائية) وتأمين كل ما يلزم الوحدة EU. تتكون BIU من الأجزاء الوظيفية التالية :

1- الرتل The Queue : تجلب BIU من الذاكرة 6 bytes على الأكثر في وقت مسبق وذلك لتسريع تنفيذ البرنامج، حيث تحفظ البايتات المجلوبة من أجل الوحدة EU بالرتل queue الذي يعمل بطريقة FIFO . وتستطيع BIU جلب بايتات التعليمات في الوقت الذي تقوم فيه EU بتفكيك شفرة تعليمة أو تنفيذ تعليمة أخرى لايتطلب تنفيذها استخدام الممرات. عندما تكون EU جاهزة للعمل تقرأ من الرتل في BIU، وهذا الأسلوب في العمل أسرع بكثير من عملية إرسال عنوان إلى الذاكرة ثم الانتظار حتى ترسل الذاكرة بايتات التعليمة التالية. إذا كان الرتل فارغا فإن EU تنتظر البايت التالي ليتم جلبه وإزاحته إلى قمة أو مقدمة الرتل. وعندما تنفذ EU تفريع أو تعليمة القفز تنقل التحكم للمكان المقابل لمجموعة متسلسلة من التعليمات. بالطبع عند تفريع البرنامج تقوم BIU بتصفير الرتل آليا وتبدأ بجلب التعليمات من الموضع الجديد لتعيد تعبئة الرتل.

2- مسجلات القطع Segment Register: وتحتوي BIU على أربعة مسجلات خاصة بالقطع الذاكرية، كل منها بطول 16bit غير قابل للتجزئة (لا يقبل أي منها التقسيم إلى مسجلين كما هو الحال بالنسبة للمسجلات العامة) وهي:

- مسجل قطعة الشيفرة Code Segment Register اختصاراً CS
- مسجل المكدس Stack Segment Register اختصاراً SS
- مسجل قطعة المعطيات Data Segment Register اختصاراً DS
- مسجل القطعة الإضافية Extra Segment Register اختصاراً ES

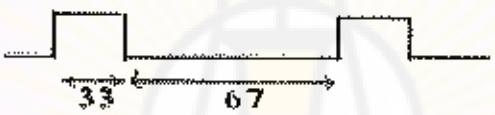
بالطبع كل من هذه المسجلات الأربعة يستخدم لحفظ البتات الستة عشر العليا لعناوين البداية للقطع الذاكرية التي يعمل عليها المعالج 8086 في أوقات محددة. يتكامل عمل مسجلات القطع مع عمل بعض مسجلات EU لحساب العنوان الفيزيائي للذاكرة بطول

20 bit وفق تراكيب محددة. فمثلا بالتركيبة (SI , DS) يستخدم المسجل SI لحفظ الإزاحة لكلمة معطيات موجودة في المقطع DS وهكذا سيكون الحال بالنسبة لبقية التراكيب، كما سنجد لاحقا.

3- مؤشر التعليمات Instruction Pointer – IP : يحتفظ IP بعنوان بايت الشفرة التالي ضمن قطعة الشفرة وهو بطول 16bit غير قابل للتقسيم.

٨-٣-٣ إشارات المعالج الصغري 8086

يبلغ العدد الإجمالي لإشارات المعالج 8086 ما يقارب 70 إشارة توزع على 40 Pins وهذا يعني إن بعض مآخذ المعالج تحمل بإشارتين أو أكثر . فإذا علمنا أن عشر إشارات موزعة على عشرة مآخذ، يتبقى 60 إشارة موزعة على 30 Pins. كافة مآخذ المعالج 8086 متوافقة مع TTL باستثناء مداخل النبضات الساعية، حيث للمعالج فلز واحد بالسويات الفولطية: 0.6 V ÷ 0.5- للمستوى Log 0 و +1 Vcc ÷ 3.9 للمستوى Log 1، ويتعاقب إشارة 33% وهذا يعني أن شكل إشارة الساعة هو كما في الشكل 5-8.



الشكل5-8: الشكل العام لاشارة الساعة لدى المعالج 8086

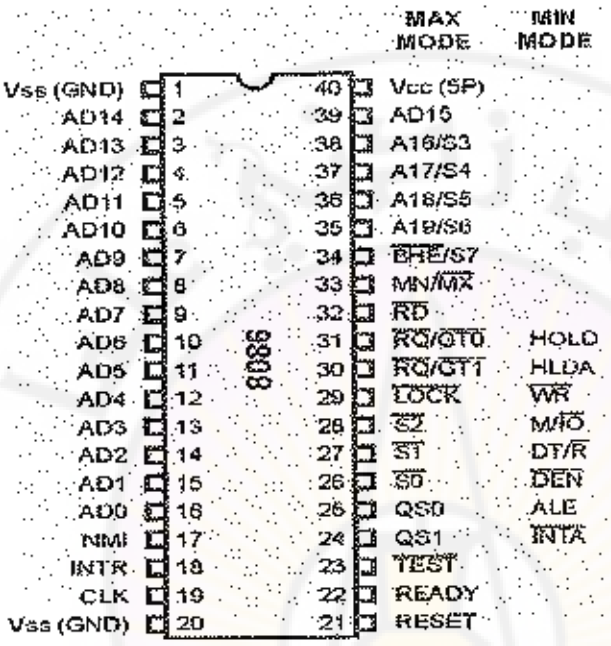
وبما أن المعالج يعمل بنمطين مختلفين يمكن تصنيف مجمل إشارات المعالج بثلاث مجموعات وفق التالي:

الأولى: مجموعة الإشارات المشتركة، وتمثل الإشارات غير المرتبطة بنمط عمل المعالج، بل هي ضرورية وموجودة في كل نمط مثل الإشارات المتعلقة بالتزامن والتغذية.

الثانية: مجموعة إشارات النمط الاصغري، وتمثل الإشارات المستخدمة عند عمل المعالج 8086 بالنمط الاصغري والذي يعني وجود معالج واحد بالمنظومة مسؤول عن تأمين الاشارات الأساسية اللازمه لعمله.

الثالثة: مجموعة إشارات النمط الأعظمي، وتمثل الإشارات المستخدمة عند عمل المعالج 8086 بالنمط الأعظمي الذي يعني وجود أكثر من معالج واحد بالمنظومة.

يظهر الشكل 6-8 توزع إشارات المعالج 8086 على الـ Pins، وقد تم وضع الإشارات التي تمثل النمط الاصغري بين قوسين للتمييز بينها وبين بقية أنواع الإشارات الأخرى، علما أن مجموعة الإشارات المشتركة تكون موجودة في كل نمط عمل للمعالج.



شكل6-8: توزع إشارات المعالج 8086

أولا- مجموعة الإشارات المشتركة

- التغذية الكهربائية ولها نفس المفهوم بكل معالج، لكن قد يختلف عدد مآخذ التغذية من معالج لآخر. فبالرغم من أن المعالج 8086 يعمل بتغذية وحيدة Ucc = +5V على الـ pin 40 إلا إنه يمتلك مأخذين للأرض (Pin 1, 20).

- إشارة 'MN/MX على الـ pin33 لتحديد نمط عمل المعالج.

- الساعة Clock على المأخذ 19 (سبق و أشرنا إلى أنه يلزم إشارات ساعية لعمل المعالج وهذه الإشارات هي الوحيدة غير المتوافقة مع TTL). تؤمن إشارة CLK اللازمة لعمل المعالج من مولد خارجي مقاد بكريستال لمزامنة العمليات الداخلية في المعالج ويقاد هذا المولد الخارجي بالدارة 8284 . يمكن أن يعمل المعالج 8086 على تواتر بين

5MHZ و 10MHZ حيث إن الدارة 8284 تقوم بتقسيم قيمة الكريستال الموصول بها على 3 وهي تولد فاز واحد يلزم لعمل المعالج بكل أنماط العمل التي يعمل بها بتناوب للإشارة قدره 33% .

– $AD_0 \div AD_{15}$ وهي تمثل ممر عناوين ومعطيات مركبة Multiplexed . أما البيتات الأربعة العلوية من ممر العناوين فهي مركبة مع بعض إشارات الحالة للمعالج وفق الشكل: A19/S6 , A18/S5 , A17/S4 , A16/S3 حيث إن القيم على هذه المآخذ تمثل في حال التعامل مع الذاكرة عناوين في الزمن T1 من كل دورة آلة، وتكون قيم هذه المآخذ في حال التعامل مع البوابات أصفاراً . أما في الأزمنة T2 وما فوق فتمثل القيم على هذه المآخذ إشارات الحالة $S_3 \div S_6$.

– المقاطعة Interrupt حيث للمعالج 8086 مأخذين للمقاطعة هما :

إشارة NMI – Non Maskable Interrupt على الـ pin17 أي مقاطعة غير مقنعة (غير قابلة للتحجيب).

و INTR على الـ Pin 18 ولهما نفس المفهوم الذي سبق ومر معنا حول طلب مقاطعة البرنامج لدى المعالج 8085A. ويشار لتخديم المقاطعة لدى 8086 بالاشارة 'INTA .

– RESET على الـ pin 21 للتصفير وهي فعالة بالواحد وبتفعيل هذه الإشارة تصفر المسجلات ES,SS,DS , IP, Flag Reg ويتم في المسجل CS وضع القيمة FFFF H وهكذا عندما ينتهي RESET سيجلب المعالج تعليمته التالية اعتباراً من العنوان الفيزيائي FFFF0H .

– Ready على الـ pin22 وهي تمثل إشارة الجاهزية، بمعنى أن هذه الإشارة هي التي تخبر المعالج فيما إن كانت المعطيات على الممر فعلية أو جاهزة وتستخدم عند التعامل مع الذواكر أو وحدات I/O البطيئة (وهي دوماً فعالة بحالة الواحد).

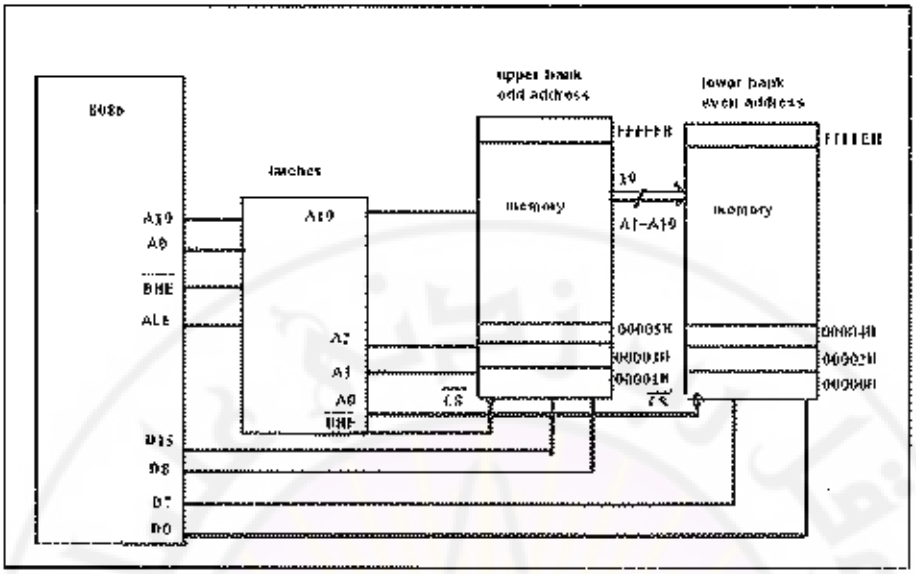
– BHE'/S7 الموجودة على الـ pin34 وهي تمثل إشارة قيادية للتحكم بنقل المعطيات على الممر كا Byte أو Word . تعمل هذه الإشارة بالتوافق مع خط العنونة A_0 لتحديد عنوان الموضع إن كان فرديا أم زوجيا. وهذا المأخذ محمل ببت الحالة S7. تعني الإشارة BHE' " تمكين الممر المرتفع High Bus Enable " تحديد العنوان بالزمن T1 من كل

دورة آلة. ولتوضيح عمل هذه الإشارة نشير إلى إن كتابة أي كلمة في الذاكرة سيؤدي إلى استخدام عنواني ذاكرة متعاقبين. ولتأمين قراءة كلمة أو كتابتها بدورة آلة واحدة، تركب الذاكرة على شكل بنكين ' 2 Banks ' من الذاكرة يتسع كل منها حتى 1/2 MB بحيث يحتوي البنك الذاكري الأول كل البايتات التي لها عناوين زوجية مثل 00000,00002,00004، ..... وتوصل خطوط المعطيات لهذا البنك إلى الخطوط $D_0 \div D_7$. أما البنك الذاكري الأخر فيحتوي كل البايتات التي لها عناوين فردية، مثل 00001,00003,00005،... وتوصل خطوط المعطيات لهذا البنك إلى $D_8 \div D_{15}$. تستخدم الإشارة 'BHE مع A_0 بحيث إن A_0 يستخدم جزءاً من عملية تمكين الذاكرة في البنك السفلي من أجل $A_0 = 0$ لتحديد العنوان الزوجي بالتالي يتم وصله وكأنه CS . المفهوم العام لكل من التراكيب الأربعة للإشارتين 'BHE و A_0 مبينة بالجدول 2-8.

جدول2-8: ترميز الاشارتين A_0 و 'BHE لتحديد نوع المعطيات

BHE'	A_0	Characteristic
0	0	Whole word
0	1	Upper byte from / to odd address
1	0	Lower byte from / to even address
1	1	None

أما عند التعامل مع كلمة تبدأ بعنوان زوجي، فإنها تقرأ من قبل المعالج بعملية واحدة بإرسال عنوان واحد، على عكس الكلمة المخزنة ابتداء بعنوان فردي والتي تقرأ من قبل المعالج على مرحلتين. يوضح الشكل 7-8 ذلك على شكل مخطط، حيث تستخدم بقية الخطوط $A_1 - A_{19}$ لتحديد الموقع المطلوب في البنك وتحديد البنك نفسه.



الشكل7-8: مخطط بنوك الذاكرة لدى المعالج 8086

عند قراءة كلمة (أو كتابتها) بطول 2 Bytes مخزنة بموقعين متتاليين في الذاكرة من أو على عنوان فردي، يلزم المعالج دورتي آلة لنسخ البايتين من الذاكرة، بحيث خلال دورة الآلة الأولى يخرج المعالج العنوان الفردي و يجعل $BHE' = 0$ عندما $A_0 = 1$ وبالتالي تتم قراءة البايت من العنوان الفردي على الخطوط $D_8 \div D_{15}$ ، أما خلال دورة الآلة الثانية، فيرسل المعالج العنوان الزوجي و يكون $A_0 = 0$ بينما $BHE' = 1$ وبالتالي تتم قراءة البايت الثاني على المآخذ $D_0 \div D_7$.

- TEST' على الـ pin 23 وهي مخصصة لفحص التعليمة Wait بحيث عندما تكون 0 يبقى المعالج في الزمن Tw وبالحالة المعاكسة يكمل البرنامج بالتعليمة اللاحقة.
- إشارة RD' على الـ pin33 وتعني أمراً لإدخال المعطيات من الممر إلى المعالج. أما بخصوص الإشارات المتبقية للمعالج فتعود إلى نمطين مختلفين لعمله، بحيث إن الإشارات: HOLD, HLDA, WR', M/IO', DT/R', DEN', ALE, INTA' تعود لعمل المعالج بالنمط الاصغري. بينما الإشارات: RQ'/GT0', RQ'/GT1', LOCK', S0', S1', S2', QS0, QS1 تعود لعمل المعالج بالنمط الأعظمي.

ثانياً: إشارات النمط الاصغري

عندما يعمل المعالج 8086 بالنمط الاصغري يكون هو المسؤول عن توليد كافة الإشارات التحكمية للمعالج للتعامل مع الدارات الأخرى، وبالتالي معظم إشارات هذا النمط أصبحت معروفة ومنها:

- إشارات التحكم 'M/IO' , 'WR حيث توجد الإشارة 'WR على الـ pin29 والإشارة 'M/IO على الـ Pin28.
- 'INTA ويوجد على الـ 24 pin وتأتي جواباً لإشارات المقاطعة.
- ALE على الـ pin35 و تعني ' Address Latch Enable ' ولهذه الإشارة نفس المفهوم لدى المعالج 8085A.
- إشارة 'DEN من Data Enable وتستخدم هذه الإشارة مع غيرها من الإشارات الأخرى للتحكم بدارات الدعم للمعالج بالنمط الاصغري، حيث يمكن وصلها إلى إشارة 'CS لبعض الدارات لتسمح بنقل المعطيات عبر ممر المعطيات، ولهذا فإن المعالج سيمكن عوازل ممر المعطيات فقط عندما تصبح 0 = 'DEN
- إشارة 'DT/R (Data Transmit/Receive) وهي الإشارة التي تحدد اتجاه نقل المعطيات من المعالج وإليه بمعنى إذا كان 1 = 'DT/R فسيكون اتجاه النقل من المعالج إلى الذاكرة أو البوابات أي إرسال، أما إذا كان 0 = 'DT/R فسيكون اتجاه النقل بالعكس أي استقبال.
- الإشارتان HOLD وجوابها HLDA للنقل بنمط الـ DMA تشبه وظيفيا إشارات RQ/GT من Request/Grant في صيغة العمل بالنمط الأعظمي للمعالج.

يضاف لما سبق الاشارات الاساية المشتركة في تعطي عمل المعالج من ساعة وتغذية كهربائية وغيرها. بالطبع كما تحسب الاشارات الاساسية مع اشارات النمط الأصغري عندما يعمل المعالج بهذا النمط تحسب أيضا مع اشارات النمط الأعظمي عندما يعمل المعالج بالنمط الأعظمي.

ثالثاً – إشارات النمط الأعظمي

وتستخدم إشارات هذه المجموعة عندما يعمل المعالج 8086 بالنمط الأعظمي وهي:

– QS0 و QS1 (Queue Status) تعطي معلومات عن حالة رتل التعليمات كما هو مبين في الجدول 3–8.

الجدول3–8: ترميز الاشارتين QS_0 و QS_1 لتحديد حالة الرتل

QS_1	QS_0	Characteristics (Queue Status)
0	0	No operation. During the last clock cycle,nothing was taken from the queue
0	1	First Byte. The byte taken from the queue was the first byte of the instruction
1	0	Queue Empty. The queue has been reinitialized as a result of the execution
1	1	Subsequent Byte. The byte taken from the queue was a subsequent byte of the instrct.

تحتوي معمارية المعالج 8086 رتل بـ 6 Bytes يسرع في تنفيذ التعليمات. عندما يحمل CS:IP بالعنوان المطلوب من المكان الذي سيبدأ منه التنفيذ، سيكون الرتل فارغاً والمعالج يبدأ بجلب أول عملية بطول 1 Byte (البايت الأول) من شيفرة التعليمة إذا كان العنوان في CS:IP فرديا أو 2 Bytes إذا كان العنوان زوجيا. بعد ذلك تتداخل عملية جلب التعليمة التالية مع تنفيذ التعليمة الحالية.

– LOCK إشارة قفل الممر وهي إشارة خرج من المعالج تشير بأن قائداً آخر لممر النظام سيمنع من كسب ممر النظام طالما أن إشارة 'LOCK فعّالة. تفعل 'LOCK ببادئة (Prefix) للتعليمة، وتبقى فعالة حتى انتهاء التعليمة التالية. عندما تنفذ CPU تعليمة تتطلب ممر النظام، فإن البادئة LOCK للتعليمة تؤمن بأن معالج آخر موصول بالنظام

سوف لن يحصل على التحكم بالممر. طالما المعالج ينفذ تعليمة ببادئة، يصرح بإشارة الخرج LOCK للممر التي يمكن أن تكون موصولة بمتحكم خارجي للممر.
- الإشارات S0',S1',S2' وهي جزء من إشارات الحالة للمعالج وترسل بصيغة مشفرة على الماخذ 26,27,28 لإنتاج إشارات ممر التحكم اللازمة المنظومة التي تملك معالجين أو أكثر. ويمكن إظهار ترميز هذه الإشارات الثلاثة بالجدول 4-8.

الجدول4-8 ترميز الاشارات S0',S1',S2' لتحديد كلمة الحالة للمعالج 8086

Status Inputs			CPU Cycles	8288 Command
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$		
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction Fetch	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

أي أن هذه الإشارات تعكس نوع العملية التي تنفذ بالمعالج، وهي تصبح فعالة بـ T4 من الحلقة السابقة وتفعل خلال T1 و T2 من حلقة الممر الحالية. أما S3, S4 فيمكنها تحديد نوع المقطع الذي يتم التعامل معه وفق ما هو مبين بالجدول 5-8.

الجدول5-8: ترميز الاشارتين S_3, S_4 لتحديد نوع المقطع

S_4	S_3	Characteristics
0	0	Alternate data (extra segment)
0	1	Stack Segment
1	0	Code or none
1	1	Data Segment

بيت الحالة S5 يعكس حالة مميزات داخلية أخرى للمعالج 8086 وهو المستوى المنطقي لمؤشر المساحية للمقاطعة. البت S6 غالبا ما يكون بالمستوى المنطقي صفر وهو يشكل مع البت S7 احتياطاً لكلمة الحالة.

- RQ'/GT1' , RQ'/GT0' (طلب/ منحة Request/Grant) وتستخدم هذه المآخذ من سيد Master آخر للممر المحلية لإجبار المعالج على التخلي عن الممر المحلية بنهاية حلقة الممر للمعالج الحالي. كل مأخذ RQ/GT هو ثنائي الاتجاه مع الأخذ بعين الاعتبار أن الإشارة بالدليل '0" هي بأفضلية أعلى.

٨-٣-٤ مقاطعات المعالج المصغري 8086

عندما يقاطع المعالج فإنه يوقف تنفيذ برنامجه الحالي ويستدعي CALL الإجراء الذي يقوم بخدمة المقاطعة وعند نهاية الإجراء يعود RETURN التنفيذ إلى البرنامج المقاطع. يمكن أن تأتي مقاطعة المعالج 8086 من ثلاثة مصادر مختلفة هي:

*- من إشارة خارجية مطبقة على مدخل المقاطعة NMI أو على المدخل INTR ويشار للمقاطعة الناجمة عن ذلك بمقاطعة المكونات الصلبة Hardware Interrupt.

*- تنفيذ تعليمة المقاطعة INT ويشار إلى هذا النوع على أنه Software Interrupt.

*- مقاطعة من بعض الشروط الحاصلة في المعالج كتنفيذ عملية التقسيم على صفر أو وجود طفح ويشار لهذا النوع بالمقاطعات الشرطية Conditional Interrupt.

وعند استجابة المعالج 8086 لمقاطعة ما يؤدي سلسلة من الأفعال الأساسية هي:

1 - ينقص SP بمقدار 2 ويدفع PUSH مسجل الأعلام على المكدس .

2 - يحجب تمكين مدخل المقاطعة INTR بتصفير مؤشر المقاطعة بمسجل الأعلام

3 - يصفر TF الموجود أيضاً في مسجل الأعلام .

4 - ينقص SP بمقدار 2 ويدفع محتويات CS الحالية على المكدس .

5 - ينقص SP من جديد بمقدار 2 ويدفع محتوى IP الحالي على المكدس .

6 - ينفذ قفزاً بعيدا غير مباشر إلى بداية الإجراء الذي نكتبه للاستجابة إلى المقاطعة.

يستخدم المكدس Stack لتخزين عناوين المقاطعة. وذاكرة المكدس مثل ذاكرة شيفرة

البرنامج أو ذاكرة المعطيات، يمكن أن تكون بأي مكان في الذاكرة الرئيسية. التعليمة IRET عند نهاية إجراء خدمة المقاطعة تعيد التنفيذ إلى البرنامج الرئيسي .

مقاطعات المكونات اللينة (الأنواع من 255 ÷ 0)

يمكن أن تستخدم التعليمة INT لتجبر المعالج 8086 على إنجاز أي نوع من أنواع المقاطعات الممكنة والبالغ عددها 256 مقاطعة، حيث كل رقم من 0 وحتى 255 يقابل نوعاً محدداً من المقاطعات.

يسمح المدخل INTR لبعض الإشارات الخارجية بمقاطعة تنفيذ البرنامج الحالي، حيث يمكن تقنيع INTR وبالتالي لا تحدث المقاطعة . فإذا كان IF = 0 تكون المقاطعة عندئذ محجوبة، ويمكن جعل IF صفراً في أي وقت وذلك باستخدام تعليمة تصغير المقاطعة Clear Interrupt. وإذا أصبح IF = 1 بتعليمة رفع المقاطعة، سيتم تمكين المدخل INTR . ويظهر الجدول 6-8 أولويات المقاطعة لدى المعالج 8086.

الجدول6-8: ترتيب أولويات المقاطعة لدى المعالج 8086

Interrupt	Priority
Divide Error	Highest
INTm ,INT0	
NMI	
INTR	Lowest

تتمتع مقاطعات المكونات اللينة بوساطة التعليمة INT باستخدامات عديدة. فمثلا بالنسبة للمقاطعات الخمسة الأولى:

التعليمة INT0 (المقاطعة نوع 0): مقاطعة التقسيم على صفر وهي تنفذ آلياً . والتعليمة INT1 (المقاطعة نوع 1): مقاطعة الخطوة الوحيدة حيث إذا أصبح TF = 1 ينفذ 8086 المقاطعة آلياً بعد تنفيذ كل تعليمة.

والتعليمة INT2: المقاطعة غير المقنعة وتنفذ آلياً عند ما يتم الانتقال من المستوى المنخفض إلى المستوى المرتفع للإشارة على المدخل NMI .

والتعليمة INT3: مقاطعة نقطة التوقف: ويكمن الاستخدام الأساسي لهذه المقاطعة في إنجاز وظيفة نقطة التوقف Break Point في منظوم ما.

والتعليمة INT4: مقاطعة الفيضان وتنفذ عندما يصبح 1 = OF .

٨-٤ تقطيع الذاكرة

لا يتعامل المعالج 8086 بلحظة ما مع كامل مجال الذاكرة التي يستطيع عنونتها (1 MB)، بل مع مقطع Segment واحد من الذاكرة ، بالتالي تقطع الذاكرة التي يتعامل معها المعالج 8086 إلى مقاطع بحجم 64 KB لكل مقطع، حيث إن كل برنامج مكتوب بلغة الأسمبلي، كما سنجد لاحقا يحتاج إلى ثلاث قطع ذاكريه هي: CS, SS, DS وقد يستخدم مقطع رابع ES ان كان البرنامج يتعامل مع معطيات طويلة من نوع السلاسل. وهنا نتساءل لماذا صممت شركة Intel أجهزة العائلة 8086 بشكل نحتاج فيه إلى استخدام مفهوم تقطيع الذاكرة Memory Segmentation. يوجد سببان لذلك هما: الأول: عند العمل مع قطع ذاكريه مؤلفة من 64KB يستطيع المعالج التعامل فقط مع عناوين فعّالة مؤلفة من 16 bits للولوج إلى أي موقع في الذاكرة بسبب أن مخطط التقطيع للمعالج 8086 يعالج ويخزن مركبات العناوين ذات الطول 16 bits فقط. الثاني: الحواسيب الصغرية التي تستخدم المعالج 8086 يمكنها المشاركة في الزمن، أي يتشارك عدة مستخدمين بوحدة معالجة مركزية واحدة. فإذا كانت الوحدة CPU قد خصصت لبرنامج المستخدم الأول 20 ms مثلا فإنها ستعمل مع المستخدم الثاني أيضاً لمدة 20 ms وبعد أن تنهي الوحدة CPU التعامل مع كل المستخدمين فإنها تعود ثانية إلى المستخدم الأول وهكذا. وفي كل مرة تنتقل فيها الوحدة CPU من برنامج مستخدم إلى برنامج آخر ينبغي عليها الولوج إلى مقطع جديد للشفرة ومقاطع جديدة للمعطيات وهذا ما يجعل عملية الانتقال أكثر سهولة باستخدام مفهوم التقطيع، بحيث يعين لكل برنامج مستخدم مجموعة مستقلة من القطع المنطقية لشفرته ومعطياته.

يحتوي برنامج المستخدم الإزاحات (الابتعادات) المحددة عن قواعد هذه القطع. ولذلك للانتقال من أحد برامج المستخدم إلى برنامج مستخدم ثان يكفي إعادة تحميل مسجلات القطع الأربعة بعناوين قاعدة القطع المعيّنة إلى برنامج المستخدم الثاني.

٨-٤-١ حساب العنوان الفيزيائي

يسمى العنوان الذي يرد عادة بالتعليمة بالعنوان الفعّال Effective Addresses وهو بطول 16 bit، ويسمى عنوان أي موضع في الذاكرة بالعنوان الفيزيائي (أو العنوان المطلق) Physical or Absolute Address وهو بطول 20 bit. وبما أن طول كل من مسجلات المعالج 8086 هو 16 bits وأن المجال العنوني للذاكرة التي يتعامل معها المعالج هو 1 MB أو ما يعادل 20 bit يطرح السؤال التالي:

كيف يمكن تعامل معالج بمسجلات طولها 16 bit مع ذاكرة عنوان كل موضع فيها مكون من 20 bit، بمعنى آخر كيف يمكن تشكيل عنوان فيزيائي بطول 20 bits باستخدام مسجلات المعالج التي طول كل منها 16 bits؟

الجواب: لتشكيل العنوان الفيزيائي لموضع في الذاكرة يلزم دوماً مسجلان كل منهما بطول 16 bit وفق التالي: بداية يوضع عنوان بداية كل مقطع بواحد من المسجلات الأربعة CS, SS, DS, ES بحسب نوع المقطع.

وتوضع الإزاحة Offset أو البعد ضمن كل من المقاطع الأربعة السابقة بواحد من المسجلات الأربعة التالية: IP, SP, SI, DI بحيث إن هذه المسجلات تتعامل مع بعضها البعض وفق ما هو مبين بالجدول 7-8.

الجدول7-8: مسجلات الدليل والقطع في حساب العنوان الفيزيائي

اسم المسجل الذي يحتوي على Offset	اسم مسجل القطعة Segment
IP	CS
SP	SS
SI	DS
DI	ES

بمعنى إذا كان التعامل مع مقطع شيفرة CS فإن الإزاحة ضمن هذا المقطع يجب أن تكون في المسجل IP وهكذا الحال بالنسبة لبقية المسجلات. عملية حساب العنوان الفيزيائي هي من مهام الوحدة BIU، حيث إن BIU تغرس دائماً أصفاراً في البيتات الأربع السفلى لعنوان بداية المقطع، ثم تعمل إزاحة نحو اليسار لتجمع الناتج مع محتوى المسجل الذي يحتوي على الإزاحة لإنتاج عنوان فيزيائي بطول bit 20. ولتوضيح ذلك نجري المثال التالي: بفرض أن عنوان بداية القطعة الخاصة بشفرة البرنامج هو 348AH مخزن بالمسجل CS وأن إزاحة التعليمة المراد حساب عنوانها الفيزيائي هو بمقدار B0E2H مخزنة بالمسجل IP. تعمل BIU على غرس صفر بأقصى يمين العنوان الموجود في CS ليصبح من الشكل 348A0H بعد ذلك تضيف له محتوى IP وفق الشكل:

```
  3 4 8 A 0
+ B 0 E 2
-----------
  3 F 9 8 2 H   وهو العنوان الفيزيائي المقابل.
```

وعند التعامل مع المعطيات يتم العمل بنفس الآلية استخدام المسجلين DS لحفظ عنوان بداية مقطع المعطيات و SI لحفظ عنوان الازاحة ضمن مقطع المعطيات. ومما يجدر ذكره هنا، هو أن مقاطع الذاكرة يمكن أن تكون مفصولة عن بعضها (كل قطعة في مكان من الذاكرة) ويمكن أن تتداخل مع بعضها من أجل البرامج الصغيرة. فعلى سبيل المثال يمكن أن تبدأ كل المقاطع عند العنوان 0000H وهذا ما يسمى بنظام أصغري (أي وكأنه يتم التعامل مع المعالج 8085A).

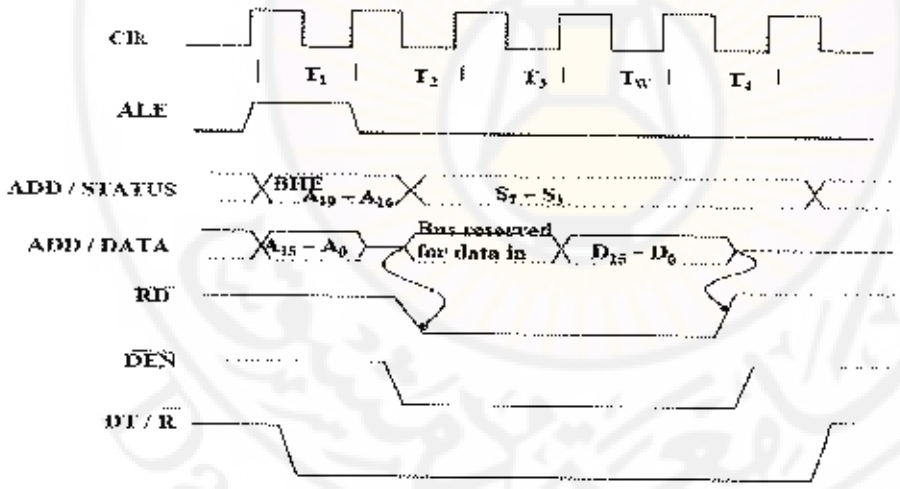
٨-٥ نظام التزامن لدى المعالج الصغري 8086

يمثل المخطط الزمني للمعالج الصغري شكلاً من أشكال توصيف فعالية المعالج على ممر النظام. يظهر خط الساعة CLK على شكل قطار من النبضات الساعية يمثل " موجة الساعة Clock Wave form " وهو عبارة عن إشارة الساعة المقادة بالكريستال من 8284 ، حيث يشار إلى الدورة الواحدة بالحالة State ويرمز لها بـ T . تتطلب كل عملية أساسية على الممر مثل قراءة Byte من الذاكرة أو كتابة Word على بوابة عدد من

الحالات . تسمى مجموعة الحالات اللازمة من أجل عملية أساسية على الممر دورة آلة " Machine Cycle " أو حلقة الممر، ويسمى الزمن الإجمالي الذي يأخذه المعالج لجلب تعليمة ما Fetch وتنفيذها Execute " بدورة التعليمة Instruction Cycle ". تتألف دورة التعليمة من واحدة أو أكثر من دورات الآلة. يزامن مولد الساعة أيضاً بعض الإشارات الخارجية مع ساعة النظام. ونكتفي هنا بالإشارة إلى عملية القراءة فقط. تتكون بشكل أساسي كل حلقات الممر للمعالج 8086 على الأقل من أربع حلقات ساعية يشار إليها بـ T1, T2, T3, T4. يرسل العنوان من المعالج خلال الزمن T1 وهو يوجد على الممر لحلقة واحدة. حلقتا جلب شيفرة التعليمة والقراءة متماثلة، لذلك تخطيط التوقيت يمكن تصنيفه بجزأين: الأول هو المخطط الزمني لحلقة القراءة.

والثاني المخطط الزمني لحلقة الكتابة.

يظهر الشكل 8-8 مخطط التزامن الأساسي لدى المعالج 8086 أثناء عمله بالنموذج الاصغري، وفق ما هو معلن من شركة Intel عن عملية قراءة بسيطة.



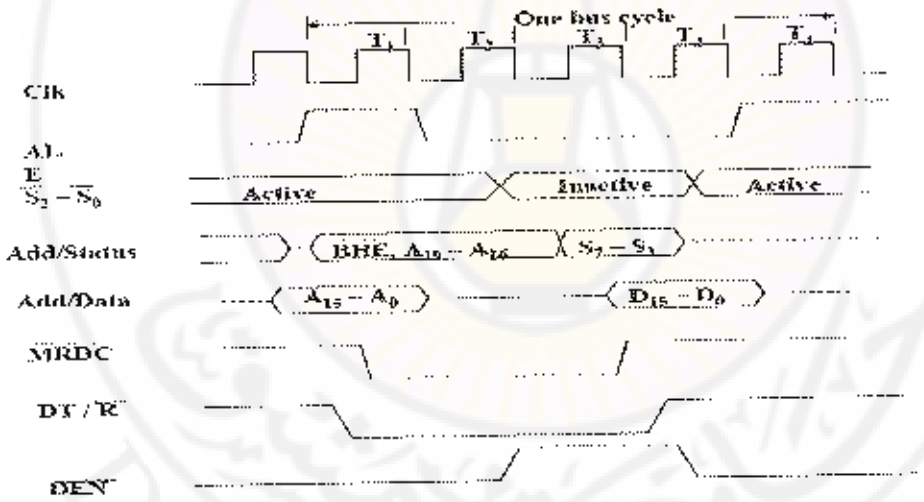
الشكل8-8: المخطط الزمني لحلقة القراءة للمعالج 8086 بالنموذج الاصغري

يمكن المعالج 8086 خلال الحالة الأولى T1 لدورة القراءة الإشارة M/IO' ويرسل بعد تمكينها قيمة مرتفعة على ALE، بحيث توصل هذه الإشارة إلى المدخل STB لدارات مسك العناوين لفصل المعطيات عن العناوين. ويوصل لهذه الدارات أيضاً الخطوط

$AD_0 - AD_{15}$ والخطوط من $A_{16} - A_{19}$ والإشارة BHE' ويبقى العنوان موجوداً على مخارج المواسك طالما إشارة ALE = 1. خلال T2 يصبح الممر بالحالة الثالثة Three State لتغير اتجاه الممر لأجل حلقة قراءة المعطيات.

تأخذ المعطيات المرسلة مكان خلال T3, T4 من المخطط الزمني وفي حال أن المعالج يتعامل مع جهاز محيطي بطيء، فان المعالج يدخل Tw (زمن انتظار واحد أو أكثر) بين T3 و T4. يزال بـ T2 العنوان من الممر المحلية ويرسل إلى الخرج، تفعّل إشارة التحكم بالقراءة RD' بالزمن T2 وبعد تفعيل RD' ستصبح المعطيات متاحة على DB.

أما بخصوص المخطط الزمني للمعالج 8086 بالنموذج الأعظمي، فان الفرق بينه وبين المخطط الزمني للمعالج بالنموذج الاصغري يكون بحالة الإشارات المستخدمة والتحكم المتاح وإشارات الأوامر المتقدمة (المحسنة Advanced). ويظهر الشكل 9-8 مخطط التزامن الأساسي للمعالج 8086 بالنموذج الأعظمي.



الشكل9-8: مخطط عملية القراءة من الذاكرة للمعالج 8086 بالنموذج الأعظمي

٦-٨ اسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

1- أي من المقاطعات التالية تمثل مقاطعة مادية (hardware interrupt)؟

a- TRAP b- INTR c- RST5.5 d- All the above

2- المكدس Stack الذي يستخدم لدى المعالج الصغري هو من النوع

a- FIFO b- LIFO c- FILO d- LILO

3- المكدس Stack في المعالج 8086 يكون

a. located in code memory

b. necessary for subroutine calls and interrupts.

c. able to be disabled only during interrupt routines

d.provides indexed offset addressing for subroutine paramet.

4- يصبح المؤشر Sign Flag = 1 في سجل المؤشرات لدى المعالج 8086 عندما

a. الناتج سالب b. الناتج موجب

c. الناتج واحد d. الناتج صفر

5- يمكن للمعالج 8086 أن يكون بال

a. Real address mode only (single task at time)

b. Protected virtual address mode only (multitasking)

c. Two mode of operation (real and protected)

d. all the above

6- الوحدتين الرئيسيتين للمعالج 8086 هما:

a. ALU and EU Unit b. control unit and ALU

c- EU and BIU d. ALU and BUS unit

7- المقاطعة ذات الأفضلية الأعلى من بين مقاطعات المعالج 8086 هي:

a- NMI b- DIV 0

c- Type 21H d- Overflow

8- المعالج الصغري بـ 20 address lines يمكنه أن يعنون ذاكرة بسعة

a. 1 KB = 1024 locations in memory

b. 1 MB = 1 048 576 locations

c. 64k = 65 536 locations in memory

d. none of the above

9- حجم مقطع الذاكرة لدى المعالج 8086 يحدد بـ

a. 256 kB b. 128 kB c. 64 kB d. 1024 kB

11- الوظيفة الرئيسية لمسجل المقطع لدى المعالج 8086 هي:

a. to hold the offset address b. to hold the starting address

c. to indicate the result d. all the above

10- عندما يعمل المعالج 8086 بالنمط الأعظمي (maximum mode)، فإن اشارات التحكم لممر التحكم تكون من

a- 8086 chip b- support circuit 8288

c- Support circuit d- All the above

11- تستخدم الاشارة BHE' لدى المعالج 8086 لربط

a- Even bank memory b- I/O

c- Odd bank memory d- DMA

12- المؤشر ZF لدى المعالج 8086 يصبح بحالة الواحد عندما

a. the result of arithmetic operation is one

b. the result of arithmetic operation is zero

c. the result of arithmetic operation is not zero

d. the result of arithmetic operation is negative number

13- يستخدم مؤشر الاتجاه (Direction flag) لدى المعالج 8086 مع

a- String instruction b- Branch instructions

c- Stack instructions d- Arithmetic instructions

14- يملك المعالج 8086 مجموعة من المسجلات للاستخدامات العامة التي يمكنها أن تضم المسجلات:

a. AX,BP,DS,DX
b. AX,BX,CX,DX
c. DS,CS,ES,SS
d. None of all

15- يمكن الحصول على العنوان الفيزيائي (The actual address) لأي تعليمة من تعليمات المعالج 8086 بجمع

a. CS register + BX register
b. SS register + SP register.
c. CS register + IP register
d. DS register + BX register

16- إذا كان المذبذب (the crystal oscillator) يعمل بـ 15 MHz فإن PCLK بالنسبة للدارة 8284 يكون:

a. 2.5 MHz b- 5 MHz c- 7.5 MHz d- 10 MHz

17- يختلف المعالج الصغري 8088 عن المعالج 8086 :

a- Data width on the output
b- Address capability
c- Support of coprocessor
a- Support of MAX/MIN Mode

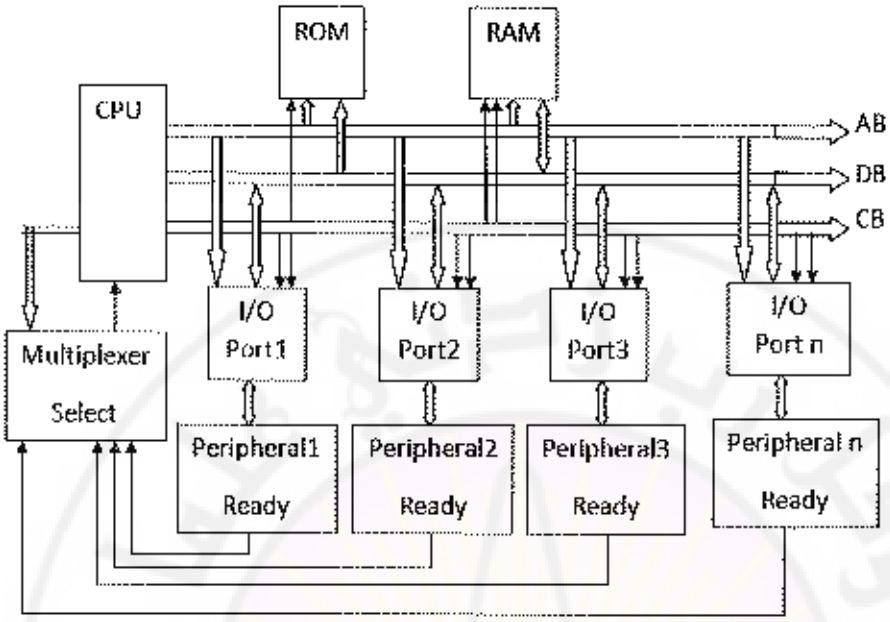
نصادف بعض المفاهيم المرتبطة بشكل مباشر مع معالجات العائلة 8086 Family، بعضها ذو مفهوم برمجي مثل أنماط عمل المعالج Processor Modes والبعض الآخر ذو مفهوم تقني مثل مفهوم الذاكرة المخبئية Cache Memory. نفرد هذا الفصل للتعرف بشكل مختصر على بعض المفاهيم المتفرقة التي تستخدم مع مختلف المعالجات بطول كلمة 16 bit وما فوق.

٩-١ المقاطعات Interrupts

ان التجهيزات المحيطية Peripherals في الانظمة الحاسوبية المبنية على المعالجات الصغرية تتطلب خدمة دورية من وحدة المعالجة المركزية وهذه الخدمة تتم عادة عبر المقاطعة Interrupt. والمقاطعة بالتعريف هي إشارة أو تعليمة تسبب ايقاف المسار الحالي (البرنامج أو المهمة) بشكل مؤقت إلى حين تنفيذ اجرائية الخدمة للجهاز الذي يطلب خدمة من المعالج (ارسال أو استقبال أية معطيات). وبشكل عام إن مختلف المحيطيات هي بطيئة بالمقارنة مع وحدة الـ CPU. بالتالي بين الازمنة التي تطلب فيها الوحدة CPU للخدمة يمكنها اجراء الكثير من العمليات (Processes). من هنا تأتي أهمية الطريقة التي تستخدم للتخديم، حيث يوجد ثلاثة اشكال لتنفيذ اجرائيات الخدمة هي

a) استطلاع المداخل/المخارج Polled I/O

بهذه الطريقة يجب على وحدة الـ CPU أن تختبر كل جهاز محيطي وبالتتابع خلال مجال زمني محدد لمعرفة إن كان بحاجة أو جاهز للتخديم. يظهر الشكل 1-9 المخطط الصندوقي للطريقة الاساسية للاستطلاع. حيث إن المعالج ينتقي بشكل تتابعي كل جهاز محيطي على حده عبر معاقب Multiplexer لمعرفة حالة خط الجاهزية للجهاز، لأن بعض الاجهزة قد تطلب خدمة في مجالات زمنية غير نظامية أو مجالات غير متوقعة. وبالتالي يجب على المعالج استطلاع الجهاز بأكبر سرعة ممكنة.



الشكل 1-9: المخطط الصندوقي لطريقة Polled I/O

المشكلة بالاستطلاع التتابعي تكّمن بالحالة التي يطلب بها الخدمة من جهازين أو أكثر بنفس الوقت. ووفق هذه الطريقة يتم تخديم أول جهاز تم استطلاعه، أما الجهاز الثاني فيجب أن ينتظر حتى ولو كان بحاجة للتخديم أكثر من الجهاز الذي تم استطلاعه أولا. تعتبر هذه الطريقة مناسبة لتخديم الاجهزة التي تحتاج للتخديم بمجالات زمنية منتظمة ومتوقعة وفي الحالات التي لاتؤخذ فيها الأفضلية بعين الاعتبار.

b) قيادة مقاطعة المداخل/ المخارج Interrupt driven I/O

تتجاوز هذه الطريقة سلبيات الطريقة السابقة، لأن الوحدة CPU تستجيب لحاجة الخدمة فقط عندما تطلب الخدمة من المحيطية، وبالتالي يمكن للمعالج أن يركّز على تنفيذ البرنامج الحالي (أن يكمل عمله) دون توقف ودون الحاجة لرؤية الجهاز المحيطي فيما ان كان يحتاج إلى تخديم. وعندما تستقبل الـ CPU اشارة مقاطعة من الوحدة I/O فانها تتوقف بشكل مؤقت عن البرنامج الحالي (البرنامج قيد التنفيذ) وتجاوب المقاطعة بجلب البرنامج

الخاص بالمحيطية (اجرائية التخديم) من الذاكرة. وعندما تنتهي CPU من التخديم تعود إلى المكان الذي كانت تعمل به قبل طلب المقاطعة.

الجهاز الذي يسمى PIC – Programmable Interrupt Controller يتعامل مع المقاطعات على اسس الأفضلية Priority. بالتالي إذا طلبت محيطيتين أو أكثر الخدمة بنفس الوقت، فان المحيطية بالأفضلية الأعلى هي التي تخدم أولاً، تليها المحيطية بالأفضلية الأقل. والمخطط الصندوقي لهذه الطريقة مماثل جدا للمخطط الوارد بالشكل–9 I بفارق أنه عوضا عن الـ Multiplexer يرسم الـ PIC حيث إن خرج الـ PIC للوحدة CPU سيكون الإشارة INTR للـ CPU وأن الـ CPU ترد على الـ PIC بالاشارة 'INTA.

المقاطعة البرمجية Software Interrupt

تمثل هذه الطريقة تعليمات برنامج يمكنه استدعاء invoke نفس اجراءات الخدمة الموصفة سابقا. الفرق هو أن الاستدعاء يتم من البرنامج عوضا عن العتاد الخارجي. عندما يتم استدعاء اجرائية الخدمة للمقاطعة تنفذ بالضبط كما يحدث بمقاطعة العتاد. المقاطعات الخمس الأولى معرفة من Intel وبقية المقاطعات معرفة بواسطة BIOS وبواسطة DOS لاجراء العديد من عمليات الادخال والاخراج، مثل قراءة وكتابة المعطيات على القرص الصلب، وكتابة المعطيات على شاشة الاظهار وقراءة المعطيات من لوحة المفاتيح.

٩–٢ أنماط عمل المعالج الصغري 8086

تشير أنماط (وسائط) المعالج إلى الطرق المختلفة التي يخلق المعالج بيئة تشغيل لها. نمط المعالجة على وجه التحديد يراقب (يتحكم) كيف المعالج يرى ويدير ذاكرة النظام والمهام التي تستخدمها. يوجد هناك ثلاثة أنماط مختلفة للعملية التي نتجت من تطور الحواسيب الشخصية وهي:

1– النمط الحقيقي Real Mode

يمكن لجهاز الكومبيوتر IBM الأصلي أن يعنون ذاكرة بسعة 1 MB فقط، وقد صممت نسخ لنظام التشغيل DOS (Disk Operating System) الأصلية للعمل آخذته ذلك بعين الاعتبار، حيث إن نظام DOS هو بطبيعته نظام تشغيل لمهمة واحدة. وهذا يعني

أنه يمكن أن يتعامل مع برنامج واحد فقط قيد التشغيل في وقت واحد. يوجد عند تشغيل المعالج في النمط الحقيقي محددات عدة، كتحديد مجال الذاكرة والولوج البطيء لها وعدم الاستفادة من استخدام كلمة بطول 32 bit للمعالجة في وحدات CPU الحديثة. يقلع الحاسوب العادي بالنمط الحقيقي وبالتالي كل المعالجات تملك هذا النمط. بالطبع النمط الحقيقي يستخدم بـ DOS وتطبيقاته القياسية. ولكن في الواقع أصبح عدد برامج DOS التي تستخدم ذاكرة قياسية بسعة 60 KB فقط قليلا جداً. يوجد ضمن DOS الآن برامج خاصة متاحة من شأنها توسيع نطاق DOS للسماح بالوصول إلى ذاكرة موسعة (فوق الـ 1 MB) وبولوج أسرع لـ 32 bit. وهذا ما يسمى بممددات DOS.

2- النمط المحمي Protected Mode

بدأ هذا النمط بالظهور مع ظهور المعالج 80286 في حواسيب IBM AT. وهو أكثر قوة بكثير بالعمل من النمط الحقيقي، ويستخدم في كافة أنظمة التشغيل متعددة المهام الحديثة Multitasking Systems. وإن من أهم مزايا (إيجابيات) النمط المحمي بالمقارنة مع النمط الحقيقي نذكر :

- الوصول الكامل إلى ذاكرة النظام، حيث لا يوجد تحديد بـ 1 MB بالنمط المحمي.
- إمكانية تعدد المهام وهذا يعني أن نظام التشغيل يملك القدرة على إدارة تنفيذ عدة برامج بنفس الوقت
- دعم الذاكرة الظاهرية Virtual Memory التي تسمح للنظام باستخدام القرص الصلب لمحاكاة ذاكرة إضافية للنظام عندما يكون ذلك ضروريا.
- ولوج أسرع للذاكرة بـ 32 bit وسواقات أسرع بـ 32 bit للعمل بالنقل I/O

يأتي اسم هذا الوضع (النمط) من استخدامه الأساسي من قبل أنظمة التشغيل متعددة المهام . كل برنامج يعمل له إسناد خاص من مواقع الذاكرة يتم حمايتها من الصراع مع البرامج الأخرى . إذا حاول أحد البرامج استخدام عنوان الذاكرة لا يسمح له باستخدامه، يولد 'خطأ حماية' . واستخدام النمط المحمي لم يصبح شعبياً حتى ظهور هيمنة نظام التشغيل مايكروسوفت ويندوز .

النمط المحمي الحالي هو الآن الأسلوب الذي يستخدمه معظم الناس في حواسيبهم الشخصية. كل أنظمة التشغيل الرئيسية في هذه الأيام تستخدم النمط المحمي بما في ذلك Windows 3.x, Window 9x, Windows NT, OS/2 and Linux حتى إن DOS التي تعمل عادة بالنمط الحقيقي يمكنها الولوج إلى النمط المحمي. كل المعالجات من 80286 وما بعد يمكنها استخدام النمط المحمي. ويمكن للمعالج 80386، وما أتى بعده من معالجات، التبديل مباشرة من النمط الحقيقي إلى النمط المحمي والعكس بالعكس، المعالج 80286 يمكنه فقط التبديل من النمط الحقيقي إلى النمط المحمي لمرة واحدة.

3- النمط الحقيقي الظاهري Virtual Real Mode

هذا النمط الثالث لعمل المعالج هو في الواقع قدرة إضافية وتعزيز للنمط المحمي وقد تم تعريفه بدءا من المعالجات 80386. يتم استخدام الوضع المحمي عادة لتشغيل رسومات أنظمة التشغيل متعددة المهام. غالبا ما يكون هناك رغبة لتكوين قدرة على تشغيل برامج DOS تحت ويندوز، ولكن برامج DOS تحتاج للتشغيل النمط الحقيقي وليس النمط المحمي. تم إنشاء الوضع الحقيقي الظاهري لحل هذه المشكلة. في الجوهر إن هذا النمط يحاكي الوضع الحقيقي من داخل الوضع المحمي، مما يسمح بتشغيل برامج DOS. يستطيع النمط المحمي لنظام التشغيل مثل ويندوز خلق آلات وضع حقيقي ظاهري متعدد، كل واحدة منها تظهر إلى البرامج التي تعمل عليها كما لو أنها البرنامج الوحيد قيد التشغيل على الجهاز. كل جهاز ظاهري يحصل على فراغ العنونة MB 1 الخاص فيه. الوضع الحقيقي الظاهري هو ما يستخدم عند استخدام صندوق DOS أو تشغيل لعبة DOS في نظام التشغيل Windows 95 . عند بدء تشغيل تطبيقات DOS يقوم Windows 95 بإنشاء جهاز DOS الظاهري من أجل أن يعمل تحته.

٩-٣ وحدة الفاصلة العشرية

من الصعوبة بمكان على أي معالج صغري أن يقوم بحساب أعداد الفاصلة العشرية لأنه في هذه الحالة سوف يستهلك الكثير من قوة المعالجة في حساب عملية واحدة. ووحدة الفاصلة العشرية هي وحدة موجودة داخل المعالج ومتخصصة في العمليات الحسابية

الخاصة بالفاصلة العشرية، وتؤدي دوراً رئيسياً في سرعة تشغيل البرامج التي تعتمد بشكل كبير على الأعداد العشرية وفي تسريع الألعاب الثلاثية الأبعاد وغيرها. وبالرغم من أن دور المعالج قد قل بفضل دخول البطاقات الرسومية المسرعة بقوتها الكبيرة، مما قلل من الاعتماد على المعالج المركزي في هذا المجال. توجد وحدة الفاصلة العشرية في المعالجات 80486 والمعالجات الأحدث (ما عدا المعالج 486SX) داخل المعالج، وقد كانت توضع في المعالجات 80386 وما قبله خارج المعالج وتسمى بالمعالج الحسابي أو المعالج المساعد Math co-processor .

وبالعودة إلى أول فرد بالعائلة 8086 Family أي المعالج الصغري 8086، يلاحظ أن البنية الداخلية له ومجموعة تعليماته صممت للتطبيقات العادية وهي غير مصممة لتعمل بكفاءة مع المعالجة الرياضية، لذلك فقد طورت لحل هذه المشكلة معالجات خاصة لها بنى داخلية ومجموعة تعليمات ذات أداء عالٍ جداً لحسابات الأعداد ومعالجة الرسوم البيانية وقد تجلى هذا الحل من شركة إنتل بإنتاج المعالج الحسابي 8087، الذي يستخدم بالتوازي مع المعالج الصغري الرئيسي في النظام بدلا من أن يقوم بالخدمة كمعالج رئيسي، لذلك يشار إليه بالمعالج المساعد. يزود المعالج المساعد 8087 بمجموعة من التعليمات الخاصة به، التي يمكن أن تكتب في البرنامج بحسب الحاجة (تبدأ تعليمات المعالج 8087 بالحرف F ليرمز للفاصلة العائمة Floating Point).

يمكن أن تكون تعليمات المعالج 8087 موزعة مع تعليمات المعالج 8086، حيث إن إضافة المعالج 8087 إلى النظام يضيف مجموعة جديدة من التعليمات الحسابية القوية لاستخدامها في كتابة البرامج. وكما ظهر المعالج 8087 مع المعالج 8086، ظهر المعالج الحسابي 80287 مع المعالج 80286 و 80387 مع المعالج 80386. توضع شفرات العمليات لتعليمات المعالج المساعد في الذاكرة مع شفرات تعليمات المعالج الأساسي، فمثلا بالنسبة للمعالج 8086، عندما يقوم بجلب بايتات التعليمات من الذاكرة يضعها في رتله الخاص وإن المعالج 8087 يقرأ أيضاً هذه البايتات ويضعها هو الآخر في رتله الداخلي، بحيث إن المعالج 8087 عندما يقوم بتفكيك الشفرة لتعليمه ما من رتله ويجد أنها تعليمة للمعالج 8086 فإنه يعالجها ببساطة وكأنها تعليمة NOP. وبنفس الشيء

بالنسبة للمعالج 8086 عندما يقوم بتفكيك شفرة تعليمه من رتله ويجد أنها لا تخصه يتعامل معها وكأنها NOP. يمكن تمييز تعليمات المعالج 8087 بكون جميع شفرات تعليماته لها القيمة 11011 في البتات الأكثر أهمية لبايت الشفرة الأولى من التعليمة. تعدّ عملية التزامن بين المعالجين 8086 و8087 مثالا لنظام مبني بمعالجات متعددة، إذ لنقل كلمات إضافية يحتاجها المعالج 8087 من الذاكرة وإليها، فإنه يتولى أمر الممرات من المعالج 8086. يستجيب المعالج 8086 للإشارة RQ_2 / GT_0 بإرسال نبضة للمعالج 8087 ثم يعوم المعالج 8086 ممراته، وعندما ينهي المعالج 8087 استخدام الممرات يرسل نبضة منخفضة أخرى على مآخذه RQ_2 / GT_0. صمم المعالج 8087 للتعامل مع صيغ الأنواع المختلفة للأعداد وهذه الأنواع العامة الثلاثة التي يتعامل معها، هي:

- الثنائي الصحيح Binary Integer .
- العشري المجمع Packed Decimal .
- الحقيقي Real .

حيث إن لجميع الأعداد الثنائية الصحيحة نفس الصيغة الأساسية التي تستخدم لتمثيل الأعداد الثنائية المؤشرة حتى ولو كانت بأطوال مختلفة. أما العدد العشري المجمع فيمثل كسلسلة مؤلفة من 18 رقما بصيغة الـ BCD مجمّعة كل رقمين في بايت واحد، وبالنسبة للأعداد الحقيقية التي يتكون كل منها من جزء صحيح، وجزء كسري فلها صيغ عدة للتمثيل بالصيغة الثنائية. والمبدأ الأساسي لكل هذه الصيغ هو استخدام مجموعة واحدة من البتات لتمثيل أرقام العدد ومجموعة أخرى من البتات لتمثيل موضع الفاصلة بعلاقتها مع هذه الأرقام. وهنا نشير إلى أن المعالج 8087 يصنف الأعداد الحقيقية عند تمثيلها بالصيغة الثنائية إلى ثلاثة أصناف هي:

- صيغة العدد الحقيقي القصير Short-real Format التي تستخدم 32 bit لتمثيل العدد ويشار لها بتمثيل ذي دقة وحيدة "Single-precision" ويستخدم من الـ 32 bit بيت واحد للدلالة على إشارة العدد و8 bit لتمثيل مقدار الأس والبتات المتبقية (23 bit) لتمثيل قيمة العدد.

والمعطيات في وقت معين.

يتضح من الشكل أعلاه، أنه عندما يخرج المعالج الصغري عنوانا ما يقوم متحكم الكاش بفحص محتويات ذلك العنوان ليرى ما إذا كانت هذه المحتويات قد نقلت إلى الكاش مسبقاً. فإذا كانت الشفرة المعنونة أو كلمة المعطيات موجودة في الكاش، فإن متحكم الكاش يمكن Enable ذاكرة الكاش لإخراج الكلمة المعنونة، على ممر المعطيات. وبما أن هذا الولوج يتم لذاكرة استاتيكية سريعة SRAM فلا حاجة لحالات الانتظار هنا. أما إذا لم تكن الكلمة المعنونة موجودة في الكاش فإن متحكم الكاش يقوم بتمكين متحكم الذاكرة الديناميكية DRAM، وبعد ذلك يرسل متحكم الذاكرة العنوان إلى الذاكرة الرئيسية للحصول على كلمة المعطيات. وبما أن الذاكرة الرئيسية DRAM أبطأ من ذاكرة كاش فإن هذا الولوج يلزمه حالات انتظار.

لكن عندما تقرأ كلمة معينة من الذاكرة الرئيسية فإنها لا تذهب فقط إلى المعالج بل تكتب أيضاً في ذاكرة الكاش، وإذا احتاج المعالج للولوج إلى كلمة المعطيات هذه مرة أخرى فإنه يستطيع عندئذ قراءة المعطيات مباشرة من الكاش بدون حالات انتظار. تسمى نسبة الولوجات التي يجد المعالج من خلالها الشفرة أو كلمة المعطيات التي يحتاجها في ذاكرة كاش " بمعدل الإصابة أو الوصول hit rate" حيث إن معظم الأنظمة الحالية تملك معدلات إصابة تفوق الـ 90%.

ولكي يتم حفظ تسلسل الأحداث عن المواقع الذاكرية الأساسية الموجودة حاليا في ذاكرة الكاش، يستخدم متحكم الكاش ما يسمى "دليل ذاكرة الكاش Cache Directory" وهذا الدليل يكون محتوى ضمن المتحكم، حيث يمثل كل موقع في الكاش بوساطة مدخل Entry في الدليل. وتعتمد الصيغة الدقيقة لمدخل الدليل على التنظيم الخاص المستخدم للكاش، حيث إن التنظيمات المختبئة الأساسية الثلاثة هي:

- التنظيم المخطط مباشرة Direct-mapped
- التنظيم الترابطي عن طريق مجموعتين Two-way set associative
- التنظيم الترابطي بشكل كامل Fully associative

لا نجد ضرورة هنا لمناقشة هذه التنظيمات.

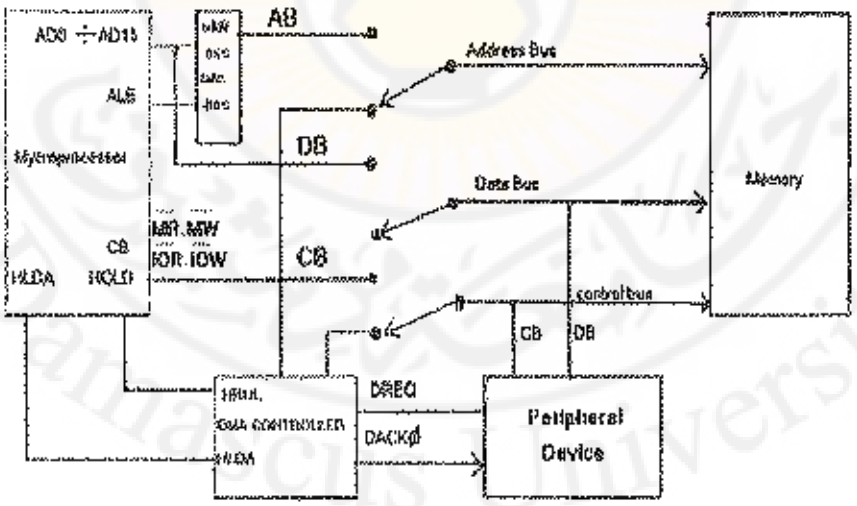
تكون ذاكرة الكاش قريبة من المعالج، ويوجد لها مستويات Levels مختلفة هي:
- الذاكرة كاش من المستوى الأول L1 Cache وهي تكون فيزيائيا بجانب نواة معالجة، لذلك تكون سريعة وثابتة عند تشغيلها. تأتي الذاكرة L1 عادة ضمن شريحة المعالج وأن حجمها النموذجي يتراوح بين 8KB و64KB. لا يتطلب هذا المستوى من ذواكر الكاش دورات تجديد (إنعاش) بحكم أنها من نوع SRAM وهي عموما تقسم إلى نصفين، بمعدل نصف للأوامر ونصف للمعطيات. فمثلا معالج Intel Celeron يستخدم ذاكرتي L1 كل منهما بسعة 16KB واحدة للتعليمات والثانية للمعطيات.
- ذاكرة كاش من المستوى L2 Cache للتخزين المؤقت، وتكون فيزيائيا قريبة من النواة ولكن يتم تنفيذها باستخدام ذاكرة الوصول العشوائي الديناميكية DRAM بالتالي يذهب العديد من دورات الساعة خلال دورات التحديث للاحتفاظ بمحتوياتها . الذاكرة L2 ليست بسرعة L1 كما أنه لا يمكن الوصول إليها خلال عملية الإنعاش. وهي تأتي بين L1 و RAM (بالترتيب processor–L1–L2–RAM) وهي أكبر من ذاكرة كاش الأولية (typically 64KB to 4MB).
- الذاكرة كاش من المستوى L3 Cache هي أيضاً ذاكرة للتخزين المؤقت وقد راج استخدامها مع ظهور وحدات المعالجة المركزية متعددة النوى. يمكن إيجاد L3 على اللوحة الأم بدلا من وجودها ضمن المعالج وهي توجد بين الـ RAM و L2.

بالتالي في منظومة تضم ذواكر كاش بالمستويات L1, L2, L3 ستكون بالترتيب: L1 -> L2-> L3-> RAM وهذا يعني إذا لم يجد المعالج المعطيات في L1 يقوم بالبحث عنها في L2 بعد ذلك في L3 وأخيرا في الذاكرة الرئيسية RAM. ذاكرة كاش L1 هي للمعالج وهي صغيرة الحجم وتعمل بنفس سرعة المعالج، لذلك هي أول ذاكرة يراها المعالج من بين الذواكر الأخرى. أما الذواكر L2 و L3 فهي امتداد لـ L1.

٩-٥ وحدة نقل المعطيات بالولوج الذاكري المباشر

تستخدم تعليمات البرنامج لنقل المعطيات من البوابات إلى الذاكرة وبالعكس. أما من أجل نقل بايتات المعطيات إلى الذاكرة في بعض التطبيقات كالقراءة من قرص ضوئي أو مغناطيسي فإن بايتات المعطيات المرسلة من القرص يكون أسرع من تلك المقروءة بوساطة تعليمات البرنامج وفي هذه الحالة يستخدم جهاز مكونات صلبة مكرس لهذا الغرض يدعى متحكم الولوج المباشر إلى الذاكرة Direct Memory Access اختصارا DMA. وبما أن نقل المعطيات عبر الـ DMA يعالج كليا بمكونات صلبة فإنه يحصل بشكل أسرع بكثير مما لو فعلنا ذلك بتعليمات البرنامج.

يوجد مشاركة زمنية للمعالج الصغري ومتحكم العملية DMA لاستخدام ممرات العناوين والمعطيات والتحكم، حيث يستعير متحكم الـ DMA بشكل مؤقت كافة ممرات المعالج AB, CB, DB لينقل بايتات المعطيات مباشرة من البوابة إلى سلسلة من المواقع الذاكرية ويستطيع أيضاً أن ينقل معطيات من الذاكرة إلى بوابة ما وتستطيع بعض أجهزة DMA أيضاً أن تنفذ التنقلات من ذاكرة إلى ذاكرة. يظهر الشكل 3-9 المخطط الصندوقي الذي يبين كيفية عمل المتحكم DMA في منظومة حاسوب صغري



شكل3-9: مخطط صندوقي يبين كيفية عمل المتحكم DMA في الحاسوب

يبين الشكل أعلاه وجود ثلاثة مفاتيح للوصل مابين المعالج والـ DMA من طرف والذاكرة من طرف آخر . عند توصيل التغذية أولاً إلى النظام تكون المفاتيح في الوضع الذي تصبح فيه الممرات كلها موصولة من المعالج إلى ذاكرة النظام والأجهزة المحيطية تهيأ جميع الأجهزة القابلة للبرمجة في النظام، وبعد ذلك يتم الذهاب لتنفيذ البرنامج ويستمر العمل حتى تتم الحاجة إلى قراءة ملف مثلا من القرص ، ترسل سلسلة الأوامر إلى جهاز متحكم القرص لتخبره بأن يذهب ليجد ويقرأ كتلة المعطيات المطلوبة من القرص وعندما يجد متحكم القرص البايت الأول للمعطيات من كتلة القرص الجاهزة، فإنه يرسل الإشارة DREQ التي تعد بمثابة طلب إلى المتحكم DMA ، إذا كان ذلك المدخل أو القناة للمتحكم DMA غير مقنع فإن DMA سيرسل طلب الاحتجاز HRQ إلى مدخل المعالج HOLD وسيستجيب المعالج لهذا المدخل بتعويم ممراته وإرسال إشارة اعتراف بالحجز أو المسك HLDA إلى المتحكم DMA.

عندما يستقبل هذا المتحكم الإشارة HLDA فإنه يرسل إشارة تحكم تحول مفاتيح الممرات الثلاثة إلى الموضع DMA . وهذا ما يفصل المعالج عن ممراته وبعدئذ يخرج المتحكم DMA على الممرات ثم يرسل إشارة اعتراف بعملية DMA هي الإشارة DACKO (DMA-Acknowledge) إلى جهاز متحكم القرص ليخبره بأن يصبح جاهزا لإخراج البايت ، وأخيرا يحفز DMA كلاً من MW و IOR على ممر التحكم ، إذا إن تحفيز MW يمكّن الذاكرة المعنونة لكتابة المعطيات فيها ويمكن تحفيز IOR متحكم القرص لإخراج بايت المعطيات من القرص على DB وبعدئذ سيكتب بايت المعطيات في الموقع الذاكري المعنون. عندما يتم نقل المعطيات، فإن DMA يلغي إشارته (طلب المسك) إلى المعالج ويدع المعالج يأخذ الممرات ثانية إلى أن يكون هناك ضرورة لعملية نقل DMA أخرى، وبعدئذ يتابع المعالج التنفيذ من حيث غادره. ويمكن لعمليات النقل DMA أن تحدث لبايت أو لكتلة في وقت معين.

٩-٦ أسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

1- أي من تراكيب المعالج التالية هي pipelining ؟

a- All x80 Processors b- All x85 Processors

c- All x86 Processors d- All the above

2- أي من مآخذ المعالج تستخدم لطلب النقل بالـ DMA والاجابة؟

a- Resel and Ready b- Ready and Wait

c- HOLD and HLDA d- None of these

3- حدّد بالنسبة للمعالج 8086 أي من العبارات التالية غير صحيحة؟

a- Coprocessor is interfaced in MAX Mode

b- Coprocessor is interfaced in MIN Mode

c- I/O can be interfaced in MAX/MIN Mode

d- Support pipelining

4- يستخدم 80 bit لتمثيل العدد إذا كان بصيغة

a- Temporary-real b- Short-real Format

c- Double-precision d- None of all

5- إن الذاكرة الخبئية (ذاكرة كاش) التي تتواجد على اللوحة الأم بدلا من وجودها ضمن المعالج هي:

a. L1 Cache b. L2 Cache

c. L3 Cache d. None of all

21- نمط العمل 2 (Mod2) لدارة الدعم 8255 يعني:

a. Port A bidirectional, and port B simple or handshaking

b. Ports A, and B are handshaking

c. Ports A, and B are simple

d. Port A and port B are bidirectional,

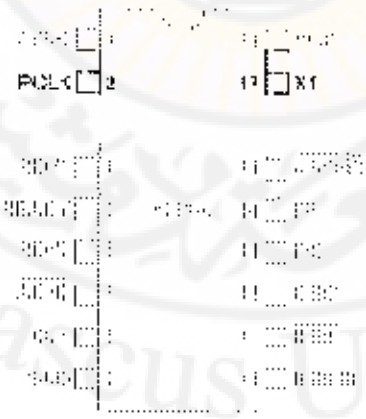
الفصل العاشر: دارات الدعم للمعالج 8086

نعرض بهذا الفصل من مجمل دارات الدعم لمعالجات العائلة 8086 تعريف مختصر بخمس دارات فقط من أكثر دارات الدعم انتشارا هي: دارة لتوليد النبضات الساعية للمعالج والتي لابد من استخدامها في كل تصميم يستخدم به المعالج 8086، ودارة مبرمجة للتحكم بأولوية المقاطعات والتي يمكن استخدامها مع كل المعالجات من إنتل، ودارتان صممتا لتحقيق اثنين من المفاهيم الواردة بالفصل السابق حول مفهوم الولوج المباشر للذاكرة ومفهوم المعالج المساعد، ودارة خاصة بالحكم على ممر النظام عند عمل المعالج 8086 بمنظومة متعددة المعالجات.

١٠-١ مولد النبضات الساعية 8284A

تمثل الدارة 8284 إحدى دارات الدعم الأساسية للمعالج 8086/8088 وهي مخصصة لتوليد الساعة المناسبة له (Clock Generator). بالطبع لا يمكن تشغيل المعالج الصغري بدون هذه الدارة، كما أن الدارة لا تصلح للعمل مع غيره وهي تماثل بالنسبة للمعالج 8086 الدارة 8224 بالنسبة للمعالج 8080A .

تقدم 8284 ثلاث وظائف أساسية هي: توليد الساعة Clock Generation، وتزامن الإشارتين Reset و Ready. تصنع الدارة بالتغليف DIL-18 ويبين الشكل 1-10 أسماء إشاراتها وتوزعها، حيث:



شكل1-10 توزع إشارات الدارة 8284

- X1 and X2 (Crystal Inputs) تمثل مآخذ لوصل كريستال خارجي. ووفقا لنشرة معطيات الدارة 8284 فان تردد الكريستال الموصول معها يقسم على 3 لإنتاج إشارة الساعة المناسبة. يبين الشكل 2-10 توصيل الكريستال في حال استخدامه.



الشكل2-10: الشكل القياسي لتوصيل الكريستال للمعالج 8086 عبر الدارة 8224

أما في حال عدم استخدام مداخل الكريستال، يجب أن تكون X1 مربوطة بـ Ucc أو بـ GND و X2 يجب أن تبقى متروكة.

- EFI (External Frequency Input) عندما يربط هذا المأخذ بالمستوى High فإن إشارة الساعة CLK للمعالج تولد من التردد الظاهر على هذا المأخذ.

- F/C' (Frequency / Crystal Select) وهو مدخل يستخدم لخيار الربط، بمعنى: عندما يكون F/C' مربوط بالمستوى L يسمح لساعة المعالج أن تكون مولدة من الكريستال. وعندما يربط بالمستوى H فإن ساعة المعالج تولد لأجل الدخل EFI.

- CLK (Processor Clock) وهي إشارة خرج للساعة تستخدم من قبل المعالج وكافة التجهيزات (الدارات) المربوطة على الممر المحلية للمعالج،يكون تردد إشارة الساعة CLK ثلث قيمة الكريستال، أو ثلث قيمة تردد المدخل EFI.

- PCLK (Peripheral Clock) هي إشارة ساعة لأغراض عامة (تستخدم للدارات المحيطية) بحيث إن تردد هذه الإشارة يساوي نصف تردد الساعة وبتناوب 50%.

- OSC (Oscilloscope Clock) وهي إشارة خرج من دارات الذبذبة الداخلية، وإن ترددها يساوي تردد الكريستال الخارجي.

- RES' (Reset Input) وهي إشارة دخل فعّالة بالصفر تستخدم لتوليد إشارة RESET للمعالج. يبين الشكل3-10 دارة تولد إشارة RES' لتكون عملية تصفير آلية عند كل إقلاع

للمعالج أو بكبسة حسب الطلب.



الشكل3-10: توصيل إشارة RESET للمعالج 8086 عبر الدارة 8224

- AEN1' , AEN2' (Address Enable) وهي فعّالة بالصفر وتستخدم لتأهيل إشارات ممر الجاهزية (RDY1 or RDY2)، بحيث:

'AEN1 تصادق RDY1 (أو RDY1 يؤهل بـ 'AEN1) ،

بينما 'AEN2 تصادق RDY2 (أو RDY2 يؤهل بـ 'AEN2)

- RDY1 and RDY2 (Bus Ready – Transfer Complete) وتستخدم للمزامنة مع إشارة الساعة قبل إرسالهما إلى المعالج، بمعنى أن كلا من الإشارتين تشير من الجهاز المتوضع (المربوط) على ممر المعطيات للمنظومة بأن المعطيات متاحة.

- READY وهي إشارة خرج من الدارة ودخل للمعالج تزامن بإشارة الدخل RDY.

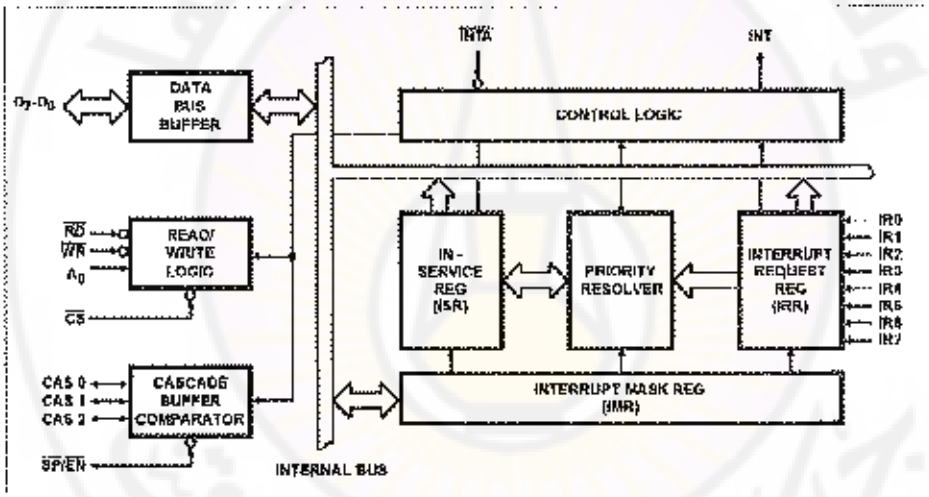
- ASYNC' (Ready Synchronization Select) وهي إشارة دخل فعّالة بالصفر تعرف نمط التزامن لمنطق READY، بحيث إذا كان 'ASYNC فعّالا يتم تجهيز مرحلتين من تزامن READY وإذا كان ASYNC' = 1 فإن مرحلة واحدة فقط من تزامن READY تجهز.

- CSYNC (Clock Synchronization) وهي إشارة دخل فعّالة بالواحد تسمح لأكثر من دارة 8284 أن تكون متزامنة لإعطاء ساعات بنفس الطور.

٤-١٠ المتحكم بأولوية المقاطعة 8259A

قد يلزم استخدام مقاطعات متعددة لمجموعة متنوعة من التطبيقات. وبما أن المعالج 8086 يملك مدخلي مقاطعة فقط هما NMI (مقاطعة غير مقنعة تخصص عادة لمقاطعة تعطل التغذية) يبقى مدخل وحيد للمقاطعة INTR وهو غير كاف، لذلك يستخدم من أجل التطبيقات التي تأتي فيها المقاطعات من مصادر متعددة دارة خاصة تدعى " متحكم المقاطعة بالأولوية PIC " (من Priority Interrupt Controller) لتحرير إشارات المقاطعة المختلفة إلى مدخل المقاطعة للمعالج.

وكمثال عن المتحكمات PIC نعرض الدارة 8259A وهي من دارات الدعم لمعالجات إنتل مصنعة بالتغليف DIL-28 والشكل 4-10 يظهر المخطط الصندوقي للدارة.



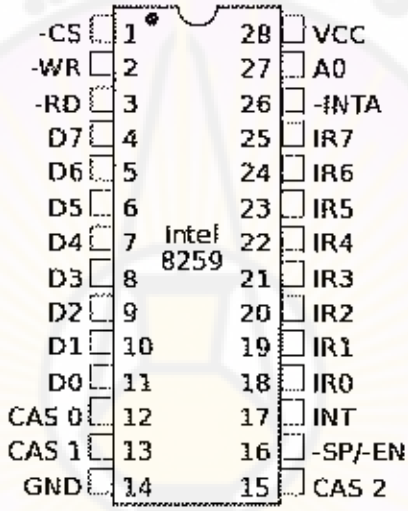
الشكل4-10: المخطط الصندوقي للتركيبة الداخلية للدارة 8259A

نكتفي هنا بعرض التركيبة العامة للدارة بالشكل الذي يسمح لنا بتوصيلها بشكل صحيح عند استخدامها، خاصة وأن متحكم المقاطعة 8259A هو من الدارات المبرمجة. يسمح ممر المعطيات للدارة بعرض 8bits بإرسال كلمات التحكم وبقراءة كلمة الأحول للدارة 8259A. تمتلك الدارة ثمانية مداخل للمقاطعة IR7 ÷ IR0 من مصادر مختلفة ومخرج الدارة INT يوصل مع INTR المعالج.

يظهر المخطط الصندوقي أيضاً الكتل الأربعة الأساسية في الدارة وهي :

- مسجل طلب المقاطعة INT REQ REG
- مسجل قناع المقاطعة INT MASK REG
- مسجل الخدمة الداخلة IN SERVICE REG
- محلل الأولوية PRIOP RESOLVER REG

وبخصوص أولويات مداخل المقاطعة، يعدّ المدخل IR0 ذا الأولوية العليا و R7 ذا الأولوية الأقل . يظهر الشكل 5-10 توزع إشارات الدارة على الـ Pins.



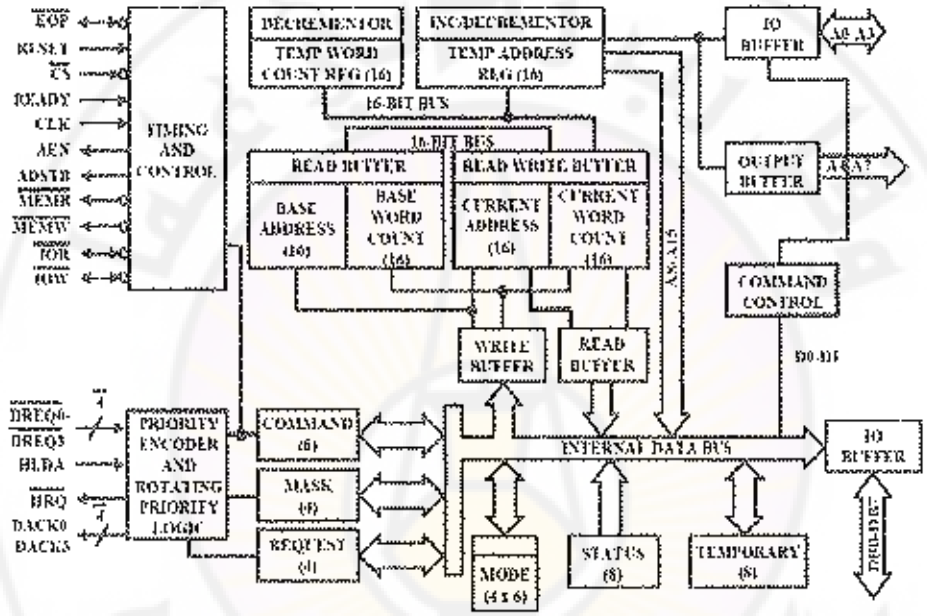
الشكل5-10: توزع إشارات الدارة 8259A

يستخدم مسجل قناع المقاطعة IMR لحجب أو تمكين مداخل المقاطعة إذ إن كل بت في هذا المسجل يتقابل مع مدخل مقاطعة بنفس العدد .

ملاحظة: إشارة ناقص (-) قبل بعض الإشارات في بعض المخططات تعني أن الإشارة المقابلة فعّالة بالصفر (حيث في الشكل أعلاه CS- تعني 'CS).

٣-١٠ المتحكم بالـ DMA الدارة Intel 8237 Controller

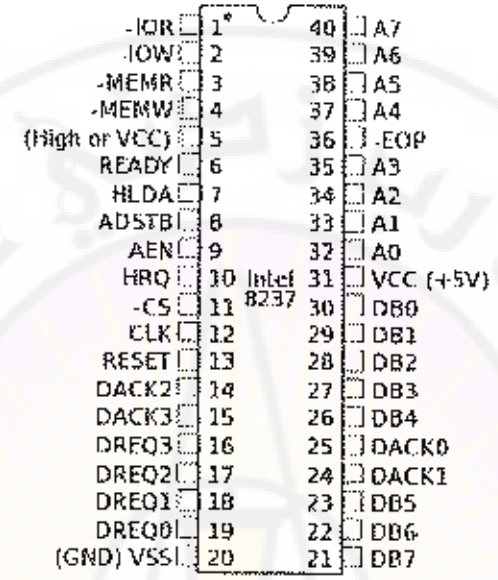
المتحكم بالولوج المباشر للذاكرة DMA (Direct Memory Access) هو كدارة متكاملة معقد بدرجة تقارب درجة تعقيد المعالج نفسه، ولذلك سنكتفي هنا بعرض التركيبة العامة للمتحكم 8237 شائع الاستخدام والذي يمثل دارة مبرمجة. ويبين الشكل 6-10 المخطط الصندوقي للمتحكم 8237



شكل6-10: المخطط الصندوقي للمتحكم 8237

يملك المتحكم 8237 أربعة مداخل أو قنوات Channels، ولكي نهيئ المتحكم 8237 يجب أن نرسل له كلمة الأمر التي تحدد العملية العامة ، كما ونحتاج أيضاً لإرسال كلمات النمط وبداية عناوين النقل وعدد البايتات الواجب نقلها من أجل كل قناة تستخدم في عملية النقل. يظهر من الشكل أعلاه أسماء الأنواع المختلفة المسجلات المستخدمة للاحتفاظ بهذه المعطيات في المتحكم 8237 وعدد البايتات في كل نوع من المسجلات وكم عدد المسجلات لذلك النوع التي يمتلكها الجهاز، أي ليس من الصعب تهيئة المتحكم. يوصل المتحكم 8237 في النظام لجهاز منفذ (بوابة port) ولهذا نكتب كلمات التهيئة حيث تقوم

بثبات العنوان الأربع السفلى $A_3 \div A_0$ معاً مع MW و IOR بتحديد المسجل الداخلي الذي سنكتب إليه أو نقرأ منه. تصنع الدارة بالتغليف DIL--40. يبين الشكل 7–10 توزع إشارات المتحكم.



شكل7–10: توزع إشارات المتحكم 8237 على مآخذه

ومن إشارات المتحكم الجديدة:

– DMA Request (DREQ0 – DREQ3) وهي خطوط فردية، تمثل مداخل الطلب غير المتزامنة للقنوات، وتستخدم من الدارات المحيطية للحصول على خدمة DMA. المدخل DREQ0 هو ذو الأفضلية الأعلى و DREQ3 ذو الأفضلية الأدنى.

– DMA Acknowledge موجودة على المآخذ DACK0 – DAC3 وتمثل إشارات خرج كجواب من DMA على طلب الخدمة.
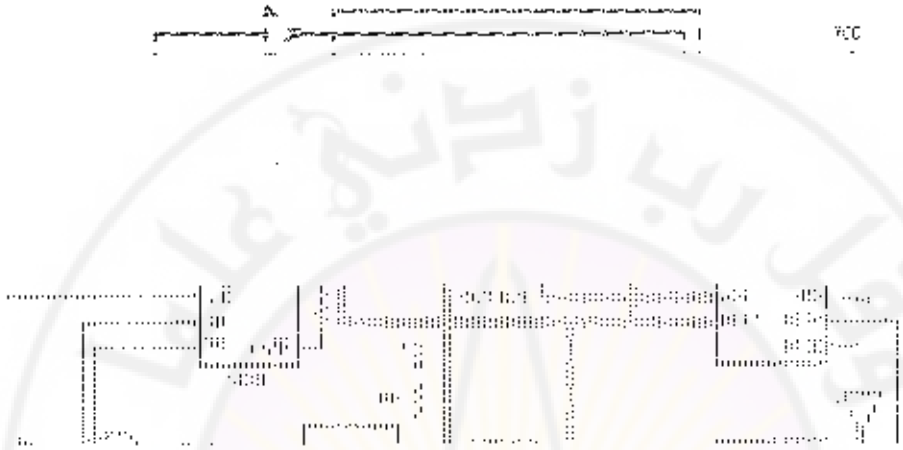
– END OF PROCESS بالرمز 'EOP' وتشير إلى المعلومات المتعلقة بانتهاء خدمة DMA التي تكون متاحة على المآخذ ثنائي الاتجاه EOP.

– ADDRESS STROBE بالرمز ADSTB وهي إشارة خرج

– ADDRESS ENABLE بالرمز AEN وهي إشارة خرج

- HOLD REQUEST بالرمز HRQ وهي إشارة خرج

ويمكن بالشكل 8-10 إظهار إحدى الأشكال الممكنة لوصل متحكم DMA بالتقنية CMOS (أي الدارة 82C37A) مع المعالج 80C88 ضمن منظومة متكاملة.

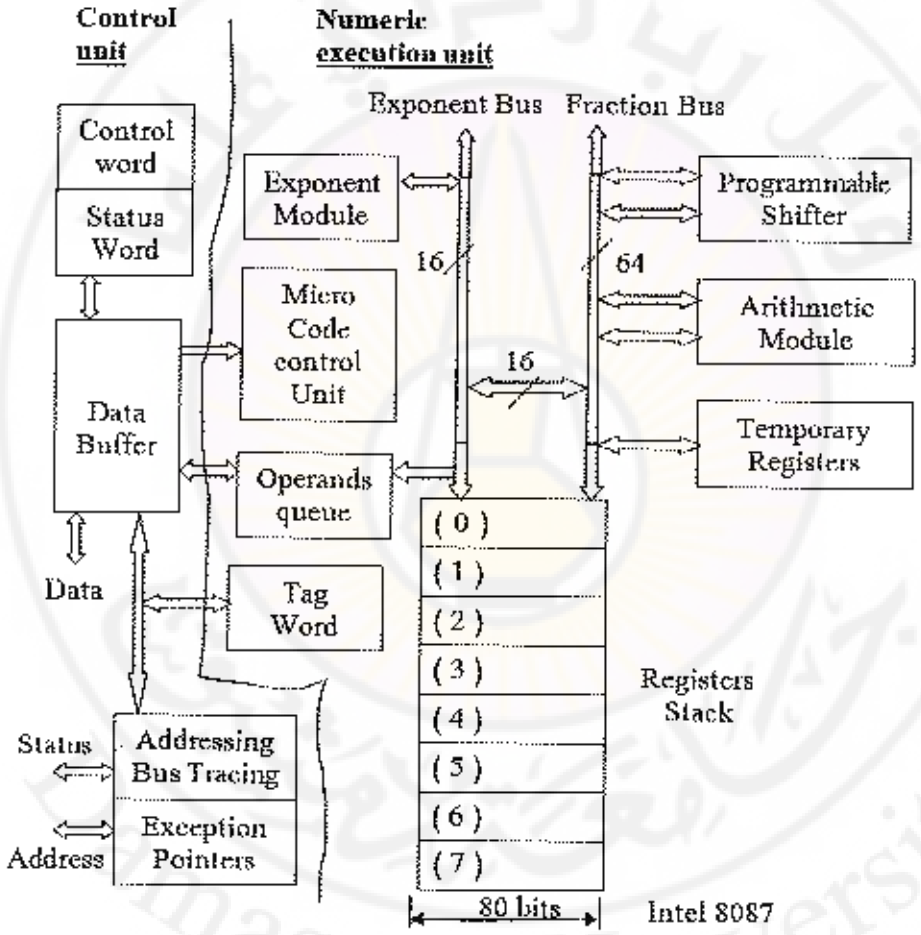


شكل8-10: نظام مايكرو كومبيوتر باستخدام المعالج 8088

وبوجود متحكم DMA

٤-١٠ المعالج الحسابي 8087

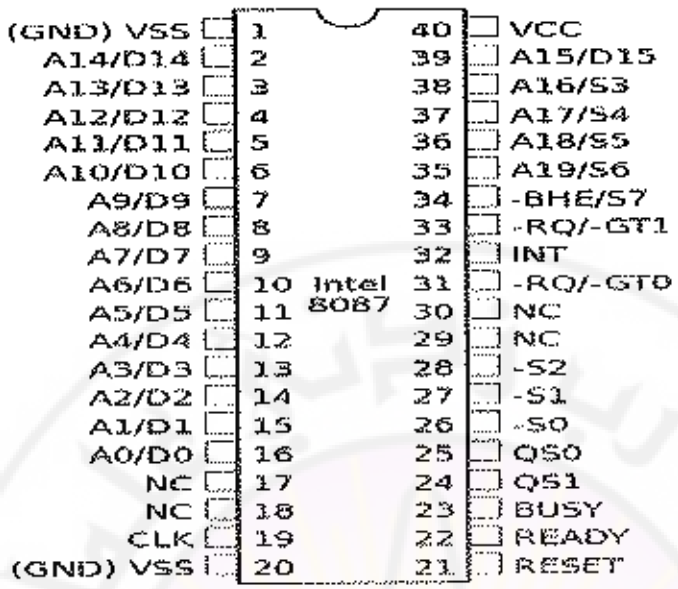
صمم المعالج المساعد 8087 ليعمل بالتوازي مع المعالج 8086 بنفس المنظومة، بحيث إن المعالج الصغري 8086 يعمل كقائد Master في تنفيذ البرنامج العام، بينما يؤدي المعالج الحسابي 8087 دوره كمقاد Slave للحسابات الرياضية الخاصة. تقارب التركيبة الداخلية للمعالج 8087 من حيث درجة التعقيد التركيبة الداخلية للمعالج الرئيسي 8086. يبين الشكل 9-10 التركيبة الداخلية للمعالج المساعد 8087.



الشكل9-10: التركيبة الداخلية للمعالج المساعد 8087

يوصل المعالج 8087 مباشرة إلى خطوط العناوين والمعطيات والأحوال للمعالج 8086، بالتالي فإن المعالج 8087 يستطيع أن يتعقب ويفكك شيفرة التعليمات المجلوبة بوساطة المعالج المضيف 8086. يملك المعالج 8087 مسجل لكلمة التحكم ومسجل للأحوال، حيث ترسل كلمات التحكم إلى المعالج 8087 بكتابتها في موقع ذاكري وعلى المعالج 8087 تنفيذ التعليمة التي تقرأ كلمة التحكم من الذاكرة. وبالمثل تقرأ كلمة الأحوال من المعالج 8087 بتنفيذه للتعليمة التي تكتب كلمة الأحوال في الذاكرة. يعمل المعالج 8087 داخليا مع جميع الأعداد بصيغة الحقيقي الموقت المؤلفة من 80 bits. ولحفظ الأعداد المعمول بها يمتلك المعالج 8087 مكدس مسجلات مكون من ثمانية مسجلات كل منها مكون من 80 bits مرقمة من 0 إلى 7 كما هو مبين بالشكل أعلاه. ويملك المعالج 8087 مؤشراً للمكدس بطول 3 bit لحفظ رقم المسجل الذي يكون حاليا في قمة المكدس. عندما يهيأ المعالج سيحمّل مؤشر المكدس بالقيمة 000، وبالتالي يكون المسجل 0 عند قمة المكدس. عندما يقرأ المعالج 8087 العدد الأول (الذي سيجري العمل عليه) من الذاكرة، فان المعالج يحول العدد إلى صيغة الحقيقي الموقت بطول 80 bit وينقص بعدئذ مؤشر المكدس إلى 111 ويكتب التمثيل الحقيقي الموقت للعدد في المسجل رقم 7. ويمكن تخيل المكدس أنه يلتف دائريا في حلقة. وإذا تم دفع Push أكثر من ثمانية أعداد على المكدس فإنها ستلتف وتكتب فوق الأعداد السابقة. يصنع المعالج 8087 بالتغليف DIL-40. والشكل 10-10 يبين توزع الإشارات على مأخذه.

ولتوضيح مفهوم إشارات المعالج 8087 نفترض بأننا سنقوم بتوصيل هذا المعالج مع المعالج 8086 بنفس المنظومة، فعندئذ يجب توصل خطوط ممر العناوين- المعطيات المضاعفة بشكل قياسي (بمعنى كل إشارة من المعالج الأول مع ما يقابلها من المعالج الثاني)، وكذلك خطوط الأحوال 'S0, S1', S2، وخطوط أحوال الزبائن QS0, QS1. توصل الإشارات طلب/ موافقة 'RQ'/GT0 من المعالج 8087 إلى المأخذ 'RQ'/GT1 للمعالج 8086. كما وتوصل الإشارة BUSY من المعالج 8087 إلى المدخل 'TEST للمعالج 8086.



شكل10-10: توزع إشارات المعالج المساعد 8087

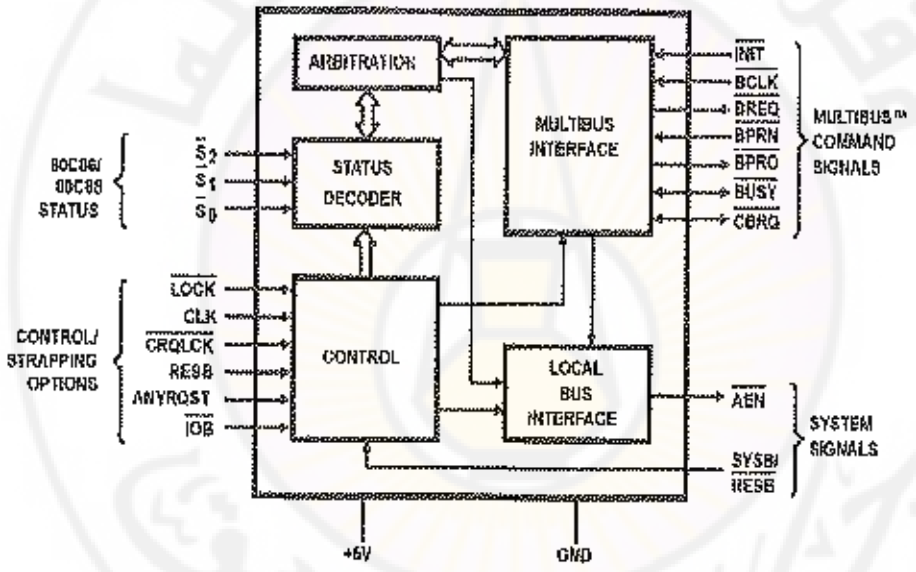
إذا توجب على المعالج 8086 أن يملك نتائج بعض الحسابات التي ينفذها المعالج 8087 قبل أن يكون باستطاعته المتابعة مع تعليماته، يجب إخبار المعالج 8086 بتعليمة WAIT كي يبقى مراقبا حالة المأخذ 'TEST حتى يجدها قد أصبحت منخفضة.

ويوصل مخرج المقاطعة INT من المعالج 8087 إلى مدخل المقاطعة غير المقنعة NMI للمعالج 8086. ويتم هكذا التوصيل بحيث إن أي حالة خطأ في المعالج 8087 تستطيع مقاطعة المعالج 8086 وتجعله يدرك شرط الخطأ هذا.

وأخيرا بخصوص المأخذ BHE والمأخذ RQ/GT1 (طلب/موافقة Request/Grant) توصل BHE مع خط النظام BHE لتمكين البنك العلوي من الذاكرة، ويكون المدخل RQ/GT1 متاحا بحيث يمكن وصل معالج مساعد آخر مثل معالج الدخل/الخرج 8089 I/O Processor الذي يعمل على التوازي مع المعالج 8087.

١٠-٥ محكم الممر (الممر) Bipolar Bus Arbiter 8289

تمثل هذه الدارة بـ 20 pins إحدى دارات الدعم للمعالج 8086 للاستخدام في الأنظمة متعددة المعالجات Multi-master/Multiprocessing المتوسطة وحتى الكبيرة. تعرض الدارة 8289 تحكيم لممر النظام في الأنظمة متعددة الممر مثل معالجات 8086/8088 مع معالج الدخل خرج IOP 8089 بنمطه البعيد Remote Mode. يعمل محكم الممر 8289 MBL بالارتباط مع متحكم الممر 8288 MBL لربط المعالجات 8086/186/188 بممر النظام متعدد الأسياد Multi-master. يبين الشكل 11-10 المخطط الصندوقي للمحكم 8289.



الشكل11-10: المخطط الصندوقي للمحكم 8289

يخدم المحكم 8289 لانتخاب المعالج (أو سيد الممر) في ممر النظام متعدد الأسياد ويتفادى مشاكل النزعم بين أسياد الممر . يبين الشكل 12-10 توزع إشارات المحكم 8289 التي نستعرض أسماؤها مع نوع كل منها فيما إن كانت إشارة دخل أم خرج بالجدول أدناه دون الدخول بالتفاصيل حول مفهوم هذه الإشارات.

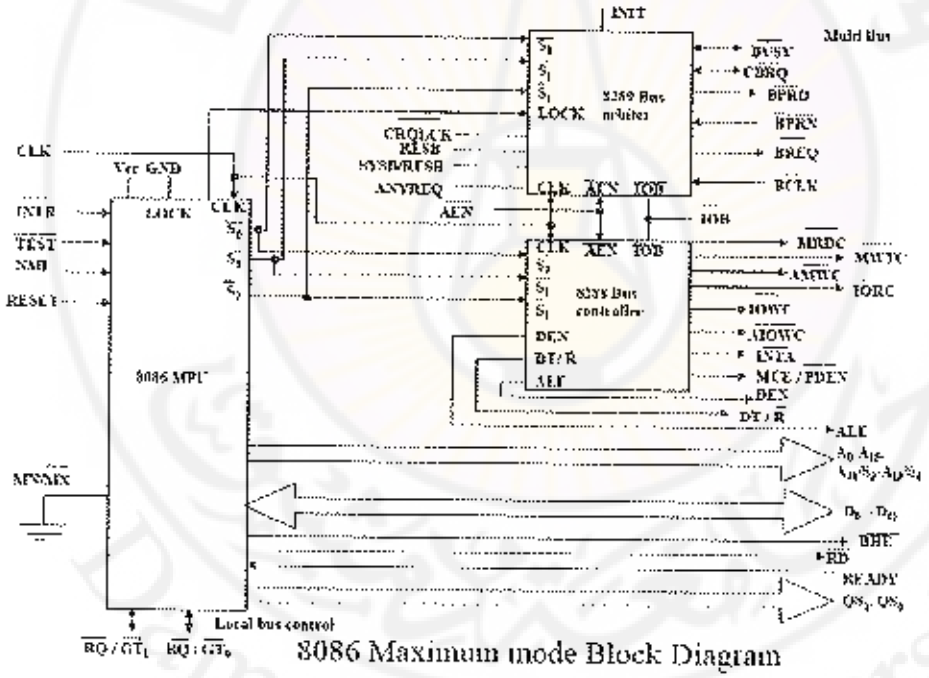
شكل12-10: توزع إشارات محكم الممر 8259

الجدول1-10: مفهوم إشارات محكم الممر 8289

pin	Name	type	pin	Name	Type
1	S2'-Status Input	In	11	BUSY'- Busy	I/O
2	IOB'- IO bus	In	12	CBRQ'- Common Bus Request	I/O
3	SYSB/RESB'- System/Resident	In	13	AEN'- Address Enable	Out
4	RESB- Resident Bus	In	14	ANYRQST- Any Request	In
5	BCLK'-BusClock	In	15	CRQLCK'- Common Request Lock	In
6	INIT'- Initialize	In	16	LOCK'- Lock	In
7	BREQ'-BusReq.	Out	17	CLK- Clock	In
8	BPRO'- Bus Priority Out	Out	18	S0'- Status Input	In
9	BPRN'- B.Priority	In	19	S1'- Status Input	In
10	GND = 0V	-	20	Ucc = 5V	

إن تغافل المعالج عن وجود المحكم وأمر بالقيام ببعض الأمور على ممر النظام في الوقت الذي لا يملك استعمال ممر النظام متعدد الأسياد فإن المحكم arbiter يمنع متحكم الممر 8288 من الاستقبال من ممر النظام المواجة (أي الممر التي يتم عليها العبور) وهذا يعني أن مخارج مواقة الممر تجبر للدخول بحالة الممانعة العالية. وطالما سلسلة الأوامر لم تصدر من قبل 8288 سيبدو ممر النظام وكأنه غير جاهز والمعالج سيدخل بحالة الانتظار. ويبقى المعالج بحالة الانتظار حتى يكتسب محكم الممر استخدام ممر النظام متعدد الأسياد، حيث المحكم سيسمح لجهاز سيطرة الممر استقبال البيانات وعنوان المزلاج للوصول أو الولوج للنظام.

ونعرض فيما يلي بشكل صندوقي استعمال المتحكم بالممر Bus Controller 8288 ومحكم الممر Bus Arbiter 8289 مع المعالج 8086 بالشكل 13-10.



8086 Maximum mode Block Diagram

الشكل13-10: المخطط الصندوقي للمعالج الصغري 8086 بالنموذج الأعظمي بوجود متحكم ومحكم الممر 8288 و 8289

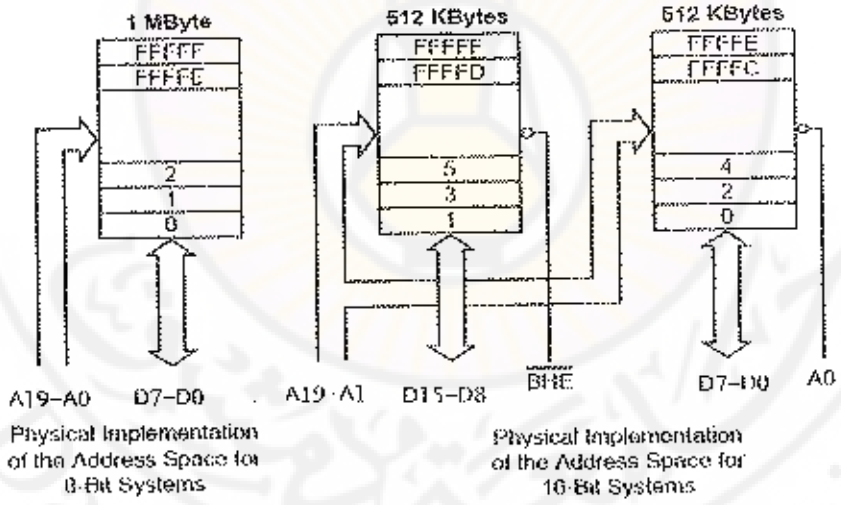
بعد أن تم توصيف المعالج الصغري 8086 تقنيا وتعرف بعض الدارات المتممة له، يمكننا المباشرة بتصميم منظومة حاسوب صغري باعتماد هذا المعالج أسوة بالتصاميم التي سبق وأشرنا إليها باستخدام معالجات الـ 8 bit وإظهار آلية توليد إشارات التحكم وتوصيل المعالج لربط ما يطلب من ذواكر ROM و RAM وربط ما يطلب من بوابات.

تتضمن عملية تصميم أي منظومة حاسوبية، تشكيل الجزأين الأساسين للمنظومة وهما:

الأول: المتعلق برسم دارة المعالج لتشكيل الممرات الأساسية (Buses).

والثاني: الذي يتعلق بربط مكونات المنظومة المطلوبة من ذواكر وبوابات بالممرات المشكلة بالجزء الأول.

يمكن تحقيق فضاء عناوين الذاكرة (المجال العنوني المعالج) بسعة 1 MB كما هو مبين بالشكل 1-11، بحيث إن ممر المعطيات بعرض 16 bit يقسم إلى بنكين من الذاكرة كل منهما بسعة 512 KB.



الشكل 1-11: النماذج الفيزيائية لممر المعطيات

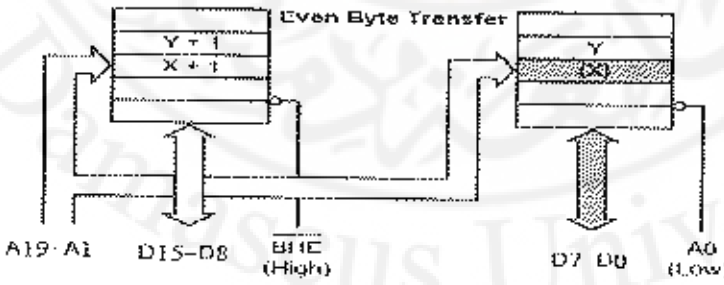
يتصل البنك الأول بالنصف السفلي لممر المعطيات D_0 حتى D_7 ويحتوي على كافة المواضع بالعنوان الزوجي (أي عندما تكون $A_0 = 0$). والبنك الآخر يتصل بالنصف

العلوي لممر المعطيات ويحتوي على كافة المواضع بالعنوان الفردي ($A_0 = 1$). يفعّل البنك الأول عندما تكون $A_0 = 0$ ويفعّل البنك الثاني عندما BHE' = 0 . وبالتالي A_0 و BHE' هي التي تخبّرنا كيف يتم نقل المعطيات عبر الممر بوجود أربع توافقيات مختلفة. يظهر الجدول 1-11 التوافقيات المختلفة لكلٍ من A_0 و BHE' في تحديد طول البيانات التي يتعامل معها المعالج 8086 مع الذاكرة إن كانت بلحظة ما بايت واحد أو كامل الكلمة (2 Bytes).

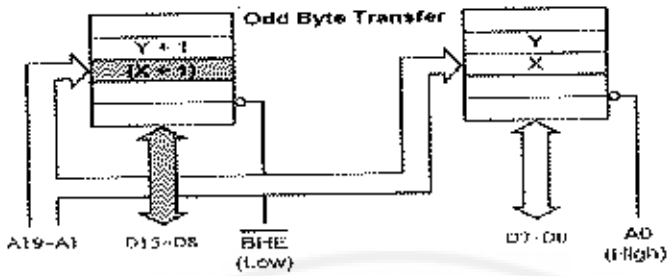
الجدول1-11: ترميز A0 و BHE' لتحديد طول البيانات التي يتعامل معها المعالج 8086 مع الذاكرة

Address	Data Type	BHE'	A_0	Bus Cycle	Data Lines Used
0 0 0 0	Word	0	0	One	$D_0 - D_{15}$
0 0 0 0	Byte	1	0	One	$D_0 - D_7$
0 0 0 1	Byte	0	1	One	$D_7 - D_{15}$
0 0 0 1	Word	0	1	First	$D_0 - D_7$
		1	0	Second	$D_7 - D_{15}$

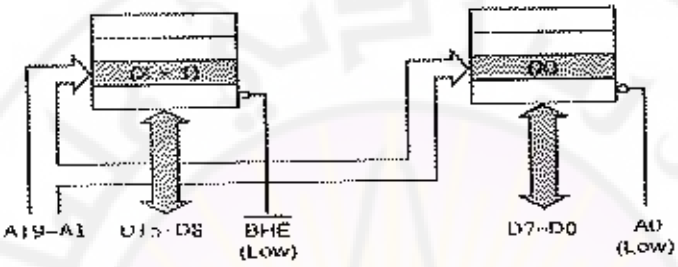
وأن خطوط العناوين من A_1 وحتى A_{19} تنتقي البايت المحدّد ضمن كل بنك. يظهر الشكل 2-11 بأجزائه الأربعة مختلف أشكال تعامل المعالج مع الذاكرة أثناء نقل Byte واحد أو كلمة (من 2 Bytes)



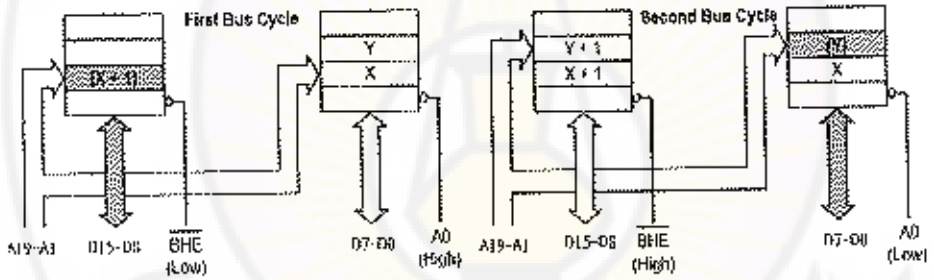
الشكل 2a-11: نقل بايت (زوجي) على 16 bit لممر المعطيات



الشكل 2b-11: نقل بايت (فردي) على 16 bit لممر المعطيات



الشكل 2c-11: نقل كلمة (زوجية) على 16 bit لممر المعطيات



الشكل 2d-11: نقل كلمة (فردية) على 16 bit لممر المعطيات

١١-١ تقنيات ترميز عناوين الذاكرة

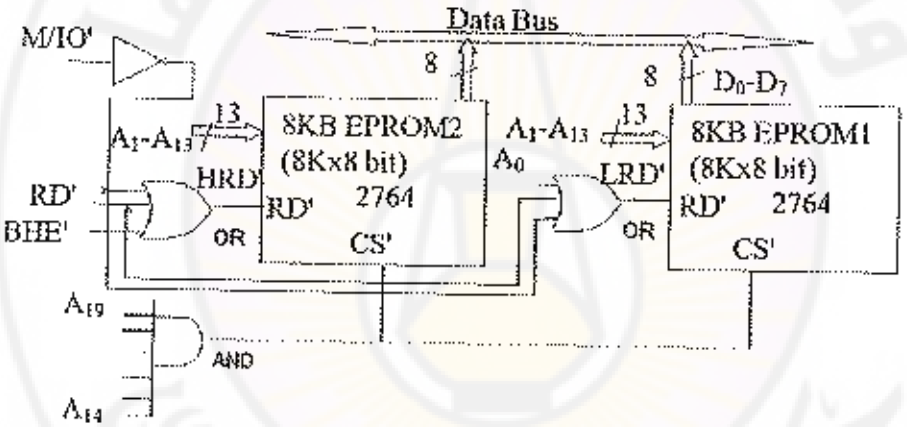
تحتوي معظم أنظمة الحاسوب الصغري على النوعين الأساسيين من الذواكر (ROM و RAM) وبالتالي خريطة توضع عناوين الذاكرة يجب أن تغطي RESET المعالج، علما أن المعالج 8086 مصمم بحيث يجلب التعليمة التالية بعد التصفير من الموقع FFFF0H وهذا يعني بأنه على هذا العنوان يجب أن تكون ذاكرة من نوع غير متطاير. تقسم الذاكرة لكتل ذاكريه، ولكي لا تغطي كتلة ذاكريه كتلة أخرى لابد من ترميز عناوين الكتل للولوج

إلى كل منها بشكل غير مرتبط. يوجد ثلاث تقنيات أساسية لترميز العناوين هي:

- Absolute Decoding الترميز المطلق
- Linear Decoding الترميز الخطي
- Block Decoding الترميز الكتلي

١- الترميز المطلق

يتم بهذه التقنية انتقاء كل شريحة ذاكرة (IC) فقط عند مستوى منطقي محدّد على خطوط العنونة. تستخدم هذه التقنية بالأنظمة الكبيرة، لكن سنحاول إظهار الشكل العام لهذه التقنية من خلال مثال بسيط، وليكن ربط ذاكرة بسعة 16 KB من نوع EPROM مثلا مع المعالج 8086 وفق ما هو مبين بالشكل 3-11.



الشكل3-11 الترميز المطلق للذواكر

نستخدم بالمثال شريحتين واحدة للبنك الزوجي والثانية للبنك الفردي، بحيث إن إشارتي التحكم A_0 و BHE' تستخدمان لتأهيل خرج البنكين وراء بعضهما البعض، على أن يوصل للبنك الأول A_0 وإلى البنك الثاني BHE' مع بقية إشارات التحكم اللازمة. يلزم كل دارة 2764 بسعة 8KB العنونة ضمن الـ Chip ثلاثة عشر خط عنوان، وبالتالي كافة خطوط العنونة المتبقية من A_{14} وحتى A_{19} تستخدم لتوليد إشارة انتقاء الدارة CS'. وانطلاقا من خارطة العناوين المبينة بالجدول 2-11.

الجدول2-11: خارطة عنونة ذاكرة EPROM بسعة 16 KB مع المعالج 8086

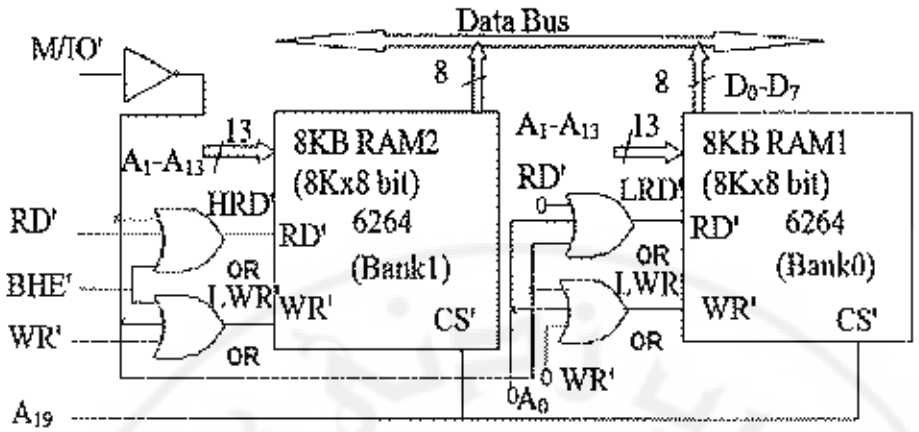
$A_{19}A_{18}A_{17}A_{16}$	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$
1 1 1 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0
.....				.....
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
1 1 1 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 1
.....				.....
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

يلاحظ أن كافة العناوين الزوجية، أي من العنوان FC000H وحتى العنوان FFFFEH تكون في البنك الأول، وتكون كافة العناوين الفردية، أي من العنوان FC001H وحتى العنوان FFFFFH في البنك الفردي. وهذه الطريقة بالعنونة تغطي عنوان ما بعد RESET (أي العنوان FFFF0H) الذي يقع ضمن البنك الأول.

إشارة الانتقاء 'CS تنتخب الداريتين معا، لكن عملية القراءة من البنك الزوجي لمعطيات بعرض 8 bit لا تتم إلا عندما يكون $A_0 = 0$ وكذلك عملية القراءة من البنك الفردي لا تتم إلا عندما تكون BHE' = 0. وفي حال أن $A_0 = BHE' = 0$ بنفس الوقت تتم قراءة كلمة كاملة من البنكين، بحيث يكون البايت الأول من البنك الأول (الزوجي Bank0) والبايت الثاني من البنك الثاني (الفردي Bank1).

٢- الترميز الخطي

يمكن بهذه التقنية للترميز، التي تستخدم في الأنظمة الصغيرة، تحديد المكونات المادية لمنطق ترميز العناوين باستخدام العدد المطلوب من خطوط العنونة (وليس كلها) وتلغى بقية الخطوط. وتسمى هذه التقنية للترميز بالترميز الجزئي Partial Decoding. يظهر الشكل 4-11 آلية عنونة 16 KB من ذواكر RAM باستخدام هذه التقنية.



الشكل 4-11: تقنية الترميز الخطي للذواكر

يستخدم بالشكل دارتي 6264 وتستخدم إشارات التحكم A_{19} و BHE' لتأهيل البنكين الذاكرين الزوجي والفردي على التوالي. يستخدم فقط خط العنونة A_{19} لاختيار شريحة واحدة من RAM عندما يكون $A_{19} = 0$، وبعكس ذلك يكون الاختيار غير متاح. إن جعل $A_{19} = 0$ لانتقاء الذاكرة RAM يترك المجال لانتقاء ذواكر EPROM في الجزء الأخير من المجال العنوني ليغطي وجود الذاكرة EPROM بعد RESET. لا تؤثر حالة الخطوط الزائدة (من A_{14} وحتى A_{18}) على منطق اختيار الشريحة وهذا يعطي عناوين متعددة (عناوين ظل Shadow Addresses)، حيث إن عدم وجود خطوط العنونة $A_{18}A_{17}A_{16}A_{15}A_{14}$ في المخطط يساهم في تشكيل تعدّد في العناوين يتراوح مابين القيمة 00000 و 11111 لهذه الخطوط. تقلل هذه التقنية تكلفة دارات الترميز ولكن تملك عائق تعدد العناوين.

٣- الترميز الكتلي

حقل الذاكرة في أنظمة المعالجات الصغرية غالبا ما يتكون من عدة كتل ذاكريه. تتطلب كل كتلة دارات ترميز لتفادي الترميز المنفصل للكتل الذاكرية.

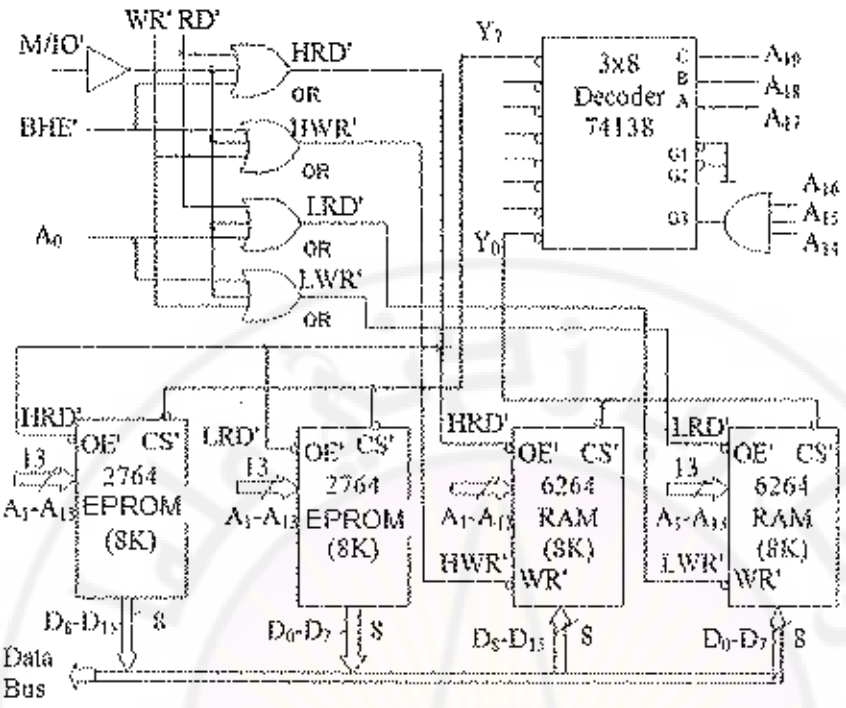
يستخدم مفكك ترميز خاص لتوليد إشارات الانتقاء لكل كتلة. فمثلا من أجل ذاكرة من نوع EPROM بسعة 16 KB، وذاكرة من نوع RAM بسعة 16 KB أيضاً، يظهر الشكل 5-11 تقنية الترميز الكتلي للذاكرة باستخدام مفكك الترميز 74138.

وبما أن أعلى مدخل لمفكك الترميز هو A_{19} فان وزن كل خرج من مخارجه الثمانية هو 128 K. يبين الشكل أيضاً المنطق المناسب لتوليد إشارات التحكم بكل من بنكي الذاكرة لكل من نوعيها، وهي Low Read و High Read للذاكرة من نوع EPROM، و'LRD', 'HRD', 'LWR', 'HWR' للذاكرة من نوع RAM. وتكون خارطة العناوين المقابلة لكل من الدارات وفق ما هو مبين بالجدول 3-11.

الجدول3-11: ترميز 16KB ذاكرة EPROM و 16 KB ذاكرة RAM مع المعالج 8086

$A_{19}A_{18}A_{17}A_{16}$	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$
0 0 0 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
0 0 0 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 1
0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
.....	.....	.....	.....	.....
1 1 1 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
1 1 1 1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

يخصص المجال العنوني من 1C000H وحتى 1FFFFH لبنكي الذاكرة من نوع RAM ، بحيث العناوين من 1C000H وحتى 1CFFEH تخصص للبنك الزوجي و 1C001H وحتى 1FFFFH تخصص للبنك الفردي، يستخدم منها 16K بمعدل 8K لكل بنك. وبأسلوب مماثل يلاحظ أن المجال العنوني من FC000H وحتى FFFFFH يخصص لبنكي الذاكرة من نوع EPROM ، بحيث إن العناوين من FC000H وحتى FFFFEH تخصص للبنك الزوجي والعناوين من FC001H، وحتى FFFFFH.
تخصص للبنك الفردي، يستخدم منها 16K بمعدل 8K لكل بنك.



الشكل 5-11: تقنية الترميز الكلي للذواكر

تستخدم إحدى 6264 كبنك زوجي لـ RAM ، والثانية كبنك فردي من أول 128 K التي تقابل خرج مفكك الترميز Y_0. ونفس الشيء بالنسبة لـ EPROM التي تربط بـ Y_7.

١١-٢ ارتباط تصميم الحاسوب الصغري بنمط عمل المعالج

سبق وأشرنا إلى أن عملية تصميم أي منظومة حاسوب صغري تقسم إلى جزأين أساسين. بخصوص الجزء الأول من التصميم حول توصيل دارة المعالج الصغري 8086 يجب أن نميز بين شكلين للتصميم هما:

- التصميم عندما يعمل المعالج 8086 بالنموذج الاصغري.
- التصميم عندما يعمل المعالج 8086 بالنموذج الأعظمي.

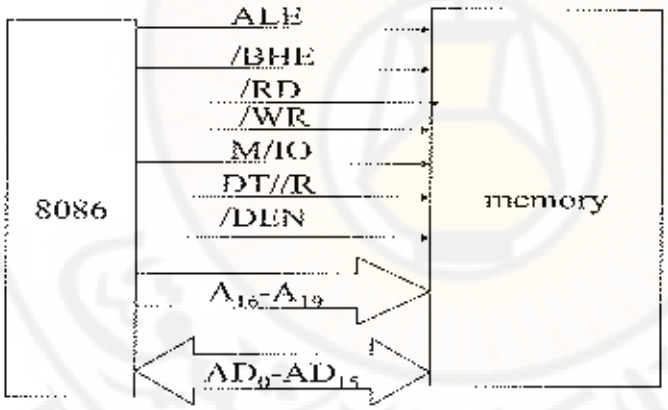
يتم التمييز بين النوعين من التصميم بالقيمة التي تسند لإشارة المعالج 'MN/MX ، حيث يتم توليد إشارات المعالج الأساسية (توليد Clock من الدارة 8284 وربط المقاطعات

بوجود منظم للمقاطعة وغيرها) في كل من نمطي العمل للمعالج وربطها بالشكل القياسي. يتميز النموذج الاصغري بوجود معالج صغري واحد في المنظومة وبالتالي يقوم المعالج بتوليد مختلف إشارات المنظومة مباشرة دون الحاجة لأي دارات دعم إضافية، أما في النموذج الأعظمي الذي يعني وجود أكثر من معالج في المنظومة الواحدة، يلزم استخدام بعض دارات الدعم الخاصة بالمعالج 8086 لتوليد إشارات التحكم.

١١-٢-١ تصميم منظومة حاسوب صغري باستخدام المعالج 8086 بالنموذج الاصغري

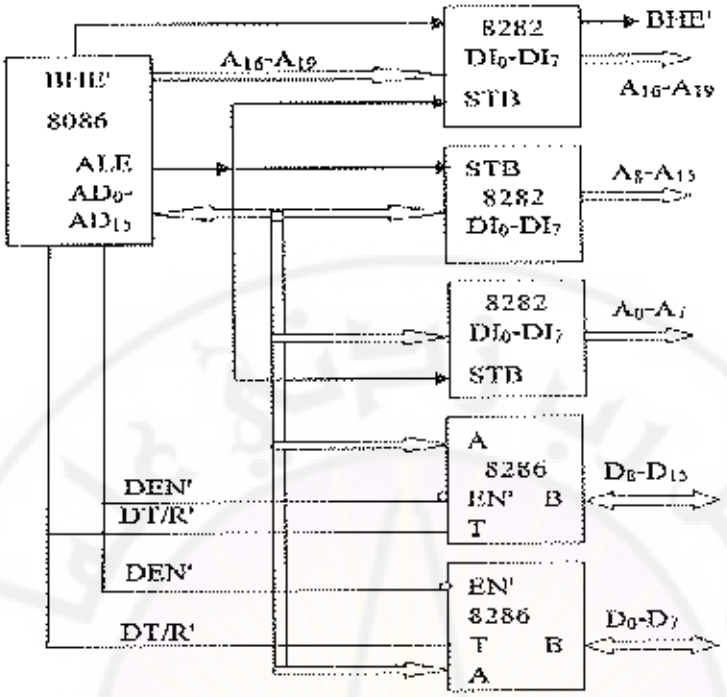
يتولى المعالج 8086 خلال توصيله بالنموذج الاصغري توليد كافة الاشارات الاساسية مثل أي معالج بطول كلمة 8 bit. يظهر الشكل 11-6 توليد المعالج للممرات الاساسية (ممرات العناوين AB بعرض 20 bit وباتجاه واحد، وممرات المعطيات DB بعرض 16 bit وبالاتجاهين بالتوصيل المباشر من المعالج) إضافة إلى ممر التحكم بالاشارات الأساسية التي يولدها للتعامل مع الذواكر والبوابات بهذا النموذج (النمط) وهي:

ALE, BHE', RD, WR, M/IO, DT/R', DEN, ...



الشكل11-6: إشارات المعالج 8086 المستخدمة بالنموذج الاصغري

ويظهر الشكل 11-7 التوصيل المباشر من المعالج إلى بعض دارات التقوية مثل الدارات 8282, 8286 وغيرها.



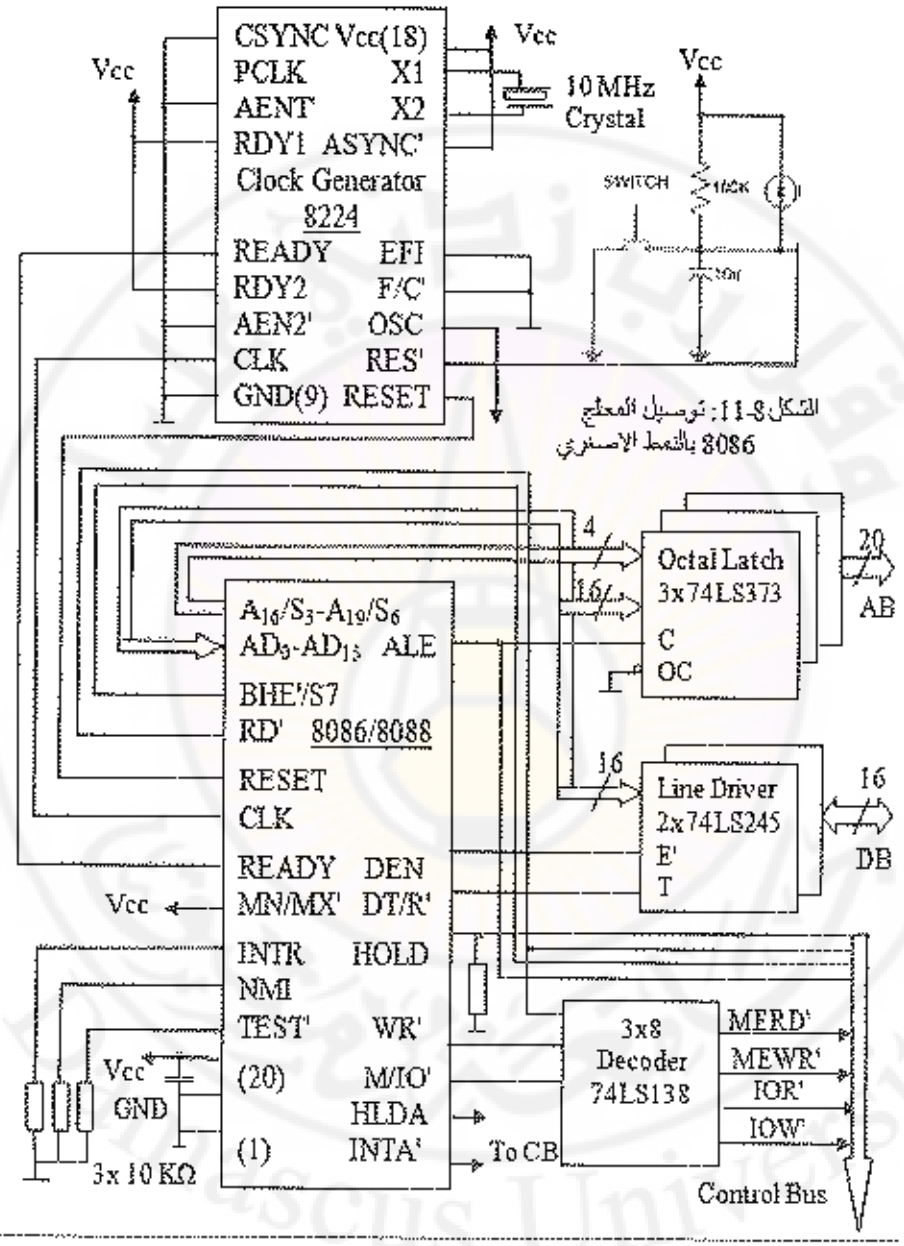
الشكل7-11: الشكل العام لتوليد ممرات AB و الـ DB للمعالج 8086

التوصيل بالشكل أعلاه مشترك بالنسبة للمعالج 8086 سواء أكان عمل المعالج بالنمط الاصغري أم بالنمط الأعظمي، وهذا يعني أن الفرق في توصيل المعالج بالنمطين يكمن في توليد إشارات ممر التحكم CB. نبين فيما يلي مجمل إشارات ممر التحكم للمعالج 8086 عند عمله بالنموذج الاصغري من خلال بعض الأمثلة.

مثال1-11: صمم منظومة حاسوب صغري باستخدام المعالج 8086 يضم 64KB ذاكرة من نوع EPROM ، و 64KB ذاكرة من نوع RAM ، وبوابة دخل وأخرى للخرج كل منهما بعرض 8 bit.

الجواب: سبق وأشرنا إلى أن للمعالج 8086 مجموعة من الإشارات التي تم تقسيمها إلى ثلاث مجموعات هي: مجموعة الإشارات المشتركة وهي الإشارات الضرورية لعمل المعالج بغض النظر عن نمط العمل. ومجموعة الإشارات الخاصة بعمل المعالج بالنمط الاصغري

ومجموعة الإشارات الخاصة بعمل المعالج بالنمط الأعظمي. يظهر الشكل 8-11 القسم الأول من التصميم.



الشكل 8-11: توصيل المعالج 8086 بالنمط الاصغري

يتمثل القسم الأول للتصميم بتوصيل المعالج 8086 لتشكيل الممرات الأساسية للحاسوب بغض النظر عن حجم الحاسوب الصغري (أي حجم الذواكر من نوع EPROM أو RAM المطلوب وعدد البوابات أو نوعها).

يمكن استخدام هذا الشكل بهذا التوصيل القياسي للمعالج في معظم الأمثلة، وقد حاولنا بالمخطط رسم الإشارات المشتركة بالطرف الأيسر من المعالج والإشارات الخاصة بنمط العمل الاصغري للمعالج بالطرف الأيمن ولم تتم الإشارة إلى مجموعة إشارات العمل الخاصة بالنمط الأعظمي.

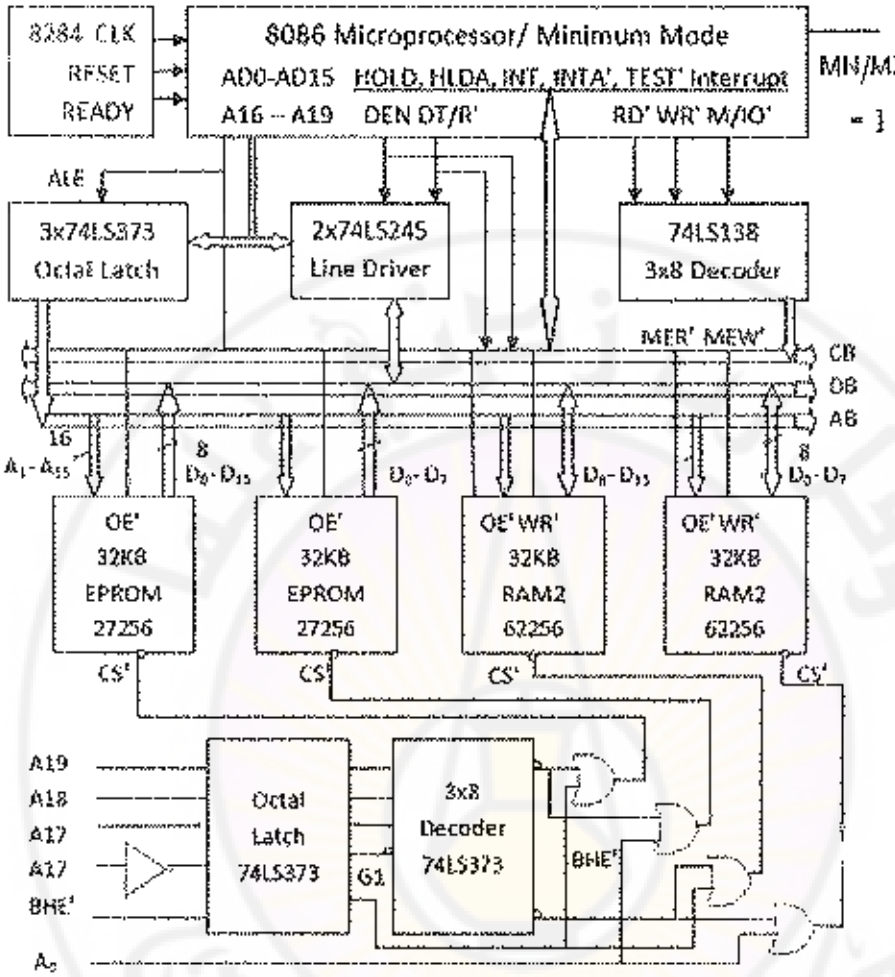
يظهر المخطط أعلاه آلية توصيل دارة الدعم 8224 لتوليد النبضات الساعية اللازمة لعمل المعالج، وكيفية توصيل كل من إشارتي READY و RESET عبر هذه الدارة.

وبخصوص القسم الثاني من التصميم المتمثل بتوصيل الذواكر والبوابات، فإن الأمر يختلف من مثال لآخر بحسب ما هو مطلوب.

يظهر الشكل 9-11 القسم الثاني المقابل للمثال 1-11 حول توصيل 64KB ذاكرة من نوع EPROM و 64KB ذاكرة من نوع RAM.

وبقصد تبسيط التصميم، تم استخدام دارتان 27256 لتشكيل الحجم المطلوب من ذاكرة EPROM بسعة 32KB لكل دارة، وكذلك دارتان 62256 لتشكيل الحجم المطلوب من ذاكرة RAM بسعة 32KB لكل دارة.

يقسم الحجم الكلي المطلوب من كلا نوعي الذاكرة إلى قسمين متساويين، أولهما البنك الزوجي (دارة واحدة بسعة 32KB) والآخر للبنك الفردي (دارة واحدة بسعة 32 KB).



الشكل9-11: توصيل 64KB من ذاكرة EPROM و 64 KB من ذاكرة RAM مع المعالج 8086 بالنمط الاصغري

بدأ التصميم برسم الممرات الأساسية للمعالج التي تم تأمينها من القسم الأول وفيه يلاحظ أن خطوط العنونة من A_1 وحتى A_{15} تستخدم لعنونة 32 KB ضمن الدارات (أو 64 KB في حال احتساب A_0 بالعناوين من 0000H وحتى FFFFH). وتستخدم بقية خطوط العنونة من A_{16} وحتى A_{19} باستخدام مفكك ترميز 3x8 لانتقاء البنك المطلوب بالارتباط مع قيمة A_0 و BHE' وفق ما هو مبين بالجدول 4-11.

الجدول4-11: ترميز 64KB ذاكرة EPROM و 64 KB ذاكرة RAM مع المعالج 8086

$A_{19}A_{18}A_{17}A_{16}$ ... A_0					Memory
d	d	d	0	d	Not select
0	0	0	1	0	Bank0(Even) – RAM (First 32 KB)
0	0	0	1	1	Bank1(Odd) – RAM (Second 32 KB)
.....					.....
1	1	1	1	0	Bank0(Even) – EPROM (First 32 KB)
1	1	1	1	1	Bank1(Odd) – EPROM (Second 32 KB)

أي أن المجال العنوني من 10000H وحتى 1FFFFH يخصص للذاكرة RAM بسعة 64 KB منها 32 KB للبنك الزوجي، و 32 KB للبنك الفردي. والمجال العنوني من F0000H وحتى FFFFFH يخصص للذاكرة EPROM بسعة 64 KB منها 32 KB للبنك الزوجي، و 32 KB للبنك الفردي.

تم استخدام خط العنونة A_{16} لتفعيل مفكك الترميز، بمعنى أنه طالما $A_{16} = 0$ فان مفكك الترميز لا يعمل، لان بوابته G1 فعّالة بالصفر. أما بخصوص توصيل البوابات الذي لم يظهر بالمخطط، نشير إليه بالمثال التالي عند توصيل المعالج 8086 بالنمط الأعظمي.

١١-٢-٢ تصميم منظومة حاسوب صغري باستخدام المعالج 8086 بالنموذج الأعظمي

يكون في النمط الأعظمي لعمل المعالج 8086 أكثر من معالج في المنظومة الواحدة (كوجود المعالج 8086 والمعالج الحسابي 8087 مثلا) ولكن بلحظة ما يكون على الممر معالج واحد، وإن كل معالج ينفذ برنامجه الخاص. تختلف المنظومة أحادية المعالج عن المنظومة متعددة المعالجات في أنه يوجد في الثانية مصادر مشتركة، تسمى بالمصادر العامة Global Resources لكل المعالجات، كما يوجد بعض المصادر المنسوبة لمعالجات تعرف باسم المصادر الخاصة Local or Private Resources. ولإظهار

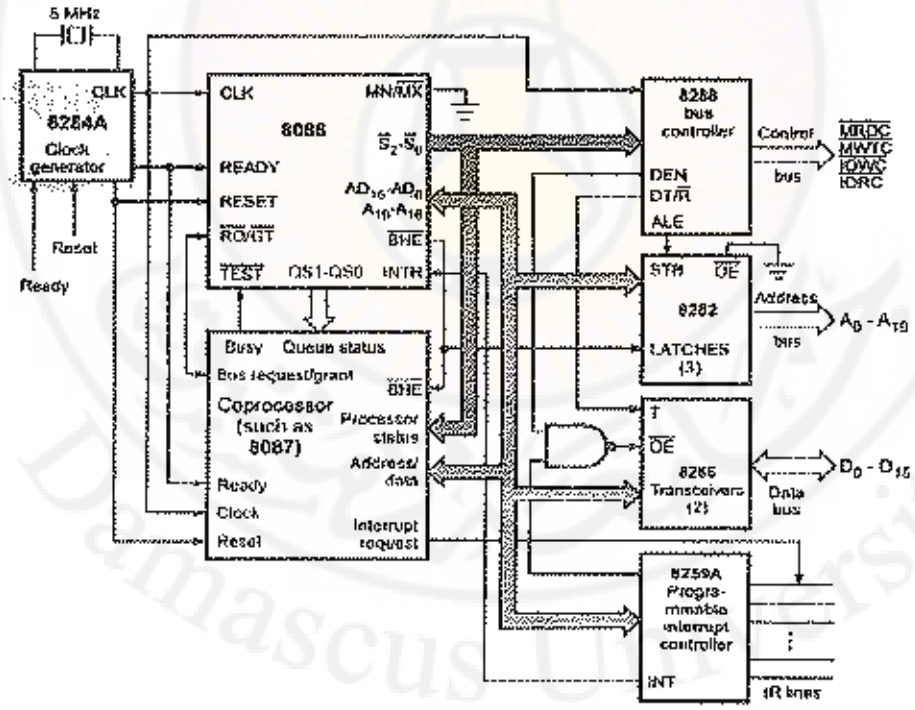
آلية تصميم منظومة متعددة المعالجات، ننطلق من مثال يعتمد على استخدام المعالج 8086 (بالنمط الأعظمي) وليكن التالي:

مثال2-11: صمم منظومة مايكروية مبنية على استخدام المعالج 8086 بالمحددات (المواصفات) التالية:

- يستخدم المعالج المساعد 8087 للحسابات العددية
- تحتوي المنظومة على 64 KB ذاكرة EPROM باستخدام الشرائح 2764
- تحتوي المنظومة على 128 KB ذواكر RAM باستخدام شرائح 62256
- تحتوي المنظومة على بوابتي دخل وبوابتي خرج كل منهما بـ 16 bit تعمل بنمط المصافحة باستخدام دارات 8255

اشرح التصميم وارسم خريطة توضع الذواكر والبوابات مستخدما تقنية الترميز المطلق.

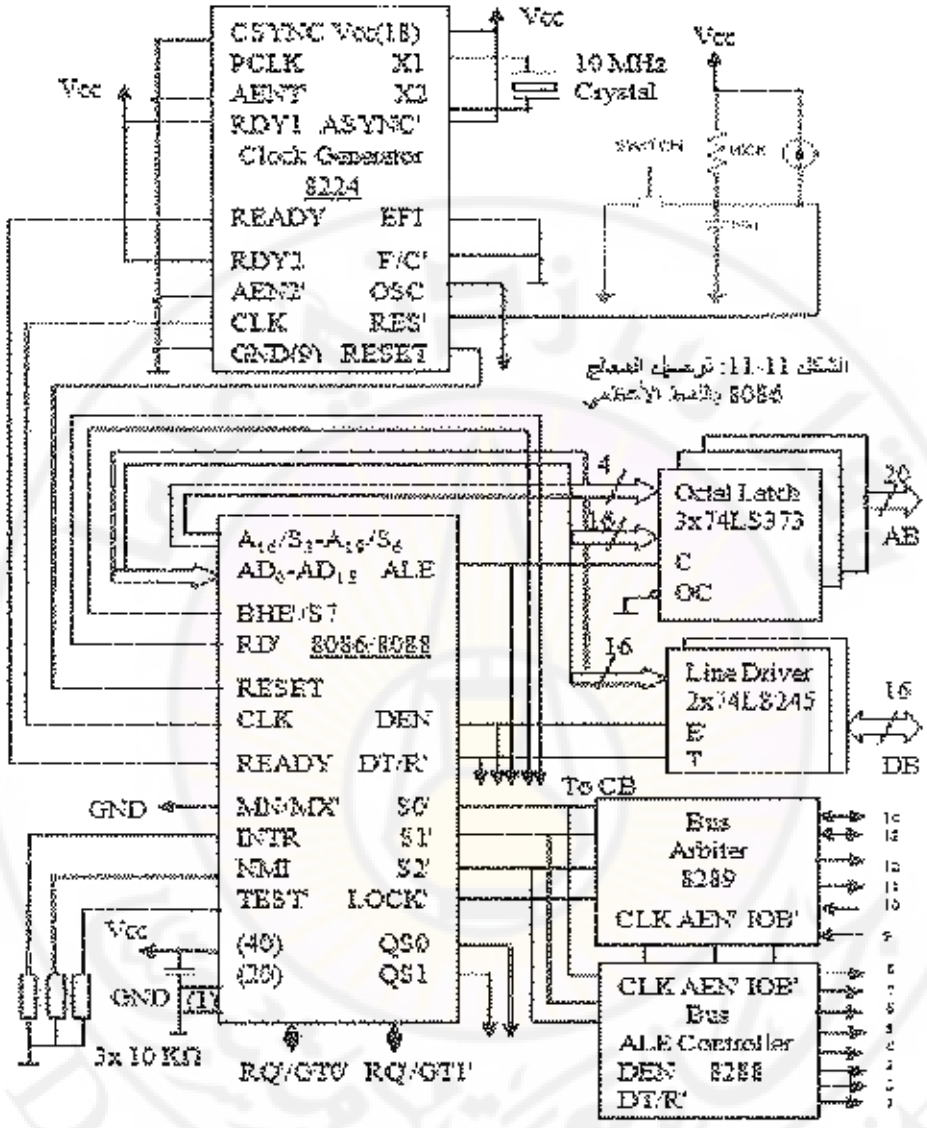
الجواب: يظهر الشكل 10-11 آلية ربط المعالج المساعد 8087 بالمعالج 8086.



الشكل10-11: ربط المعالج المساعد 8087 مع المعالج 8086

يظهر المخطط الصندوقي أعلاه آلية ربط المعالج الرئيسي 8086 مع المعالج المساعد 8087 بمنظومة واحدة تعتبر منظومة متعددة المعالجات. وتظهر آلية تشكيل كل من ممر العناوين $A_{19} - A_0$، وممر المعطيات $D_{15} - D_0$ ، وتشكيل بعض الإشارات التحكمية باستخدام متحكم الممر 8288. كما يظهر شكل ربط منظم المقاطعة 8259 بالمنظومة في حال الحاجة لاستخدامه.

ويمكن ربط ما هو مطلوب من ذواكر EPROM و RAM وبوابات دخل وخرج بشكل مباشر مع الممرات المشكلة. لا يولد المعالج في هذا النمط إشارات التحكم بل يرسل إشارات الحالة 'S0', S1', S2' (للمتحكم بالممر 8288) وإشارات الحالة للرتل QS0, QS1 (المعالج 8087) لتوليد إشارات التحكم دون أن يولدها بشكل مباشر. توصل خطوط العنونة العليا $A_{19} - A_{16}$ من المعالج 8086 إلى المعالج 8087 بشكل مباشر وان 8087 يستقبل نفس إشارات الـ Clock والـ Reset للمعالج 8086. كما أن إشارة Bus request/grant من المعالج 8087 توصل بإشارة 'RQ0'/GT0' أو 'RQ0'/GT0' للمعالج 8086. تربط إشارة Busy من المعالج 8087 بالمدخل 'TEST' للمعالج 8086، حيث إن المستوى 0 على المأخذ 'TEST' يعني أن الخرج Busy يشير إلى أن المعالج 8087 أنهى عملية الحساب. نرسم دارة مستقلة للمعالج بكل إشاراته الأساسية (الممرات الثلاثة) كجزء أول ونرسم بقية المتطلبات بالمثال كجزء ثان للتصميم. يظهر الشكل 11-11 الجزء الأول من التصميم بشكل متكامل حول توصيل دارة المعالج 8086 بالنمط الأعظمي، وبشكل يصلح للاستخدام في أي مثال آخر مماثل. وتماشيا مع المثال السابق (الشكل 8-11) حاولنا بالتصميم المحافظة على نفس الترتيب لإظهار الفرق بالتوصيل بين نمطي عمل المعالج، أي تم في الطرف الأيسر للمعالج رسم الإشارات المشتركة، وفي الطرف الأيمن رسم الإشارات الخاصة بنمط العمل الأعظمي. وبالنظر لعدم اتساع المخطط لتسمية كافة الإشارات وبخاصة إشارات متحكم الممر 8288 ومحكم (مخصص) الممر 8289 تم ترقيم إشارات الخرج للدارتين، بحيث إن الإشارات من 1 وحتى 8 تعود للدارة 8288 وهي تمثل الرموز والمفاهيم التالية:



الشكل 11-11: توصيل المعالج 8086 بالنمط الأعظمي

1- MRDC' -- Memory Read Command

2- MWRT' - Memory Write Command

3- AMWC' -- Advanced I/O Write Command

4- IORC' -- I/O Read Command

5- IOWC' -- I/O Write Command

6- AIOWC' - Advanced Memory Write Command

7- INTA' - Interrupt Acknowledge

8- MCE/PDEN'

تولد إشارات التحكم عبر 8288 بوجود الإشارات ALE , DT/R' ,DEN وهذه الإشارات لها نفس الوظيفة التي وصفت بها بالنمط الاصغري.

أما محكم الممر 8289 الذي يثبت المعالج على الممر وفق بروتوكول تخصيص الممر في الأنظمة المبنية على المعالج 8086 يولد الإشارات المرقمة من 9 وحتى 14 في المخطط وهي تمثل الرموز والمفاهيم التالية:

9- BUSY' - Bus Busy

10- CBRQ' - Common Bus Request

11- BPRO' - Bus Priority Out

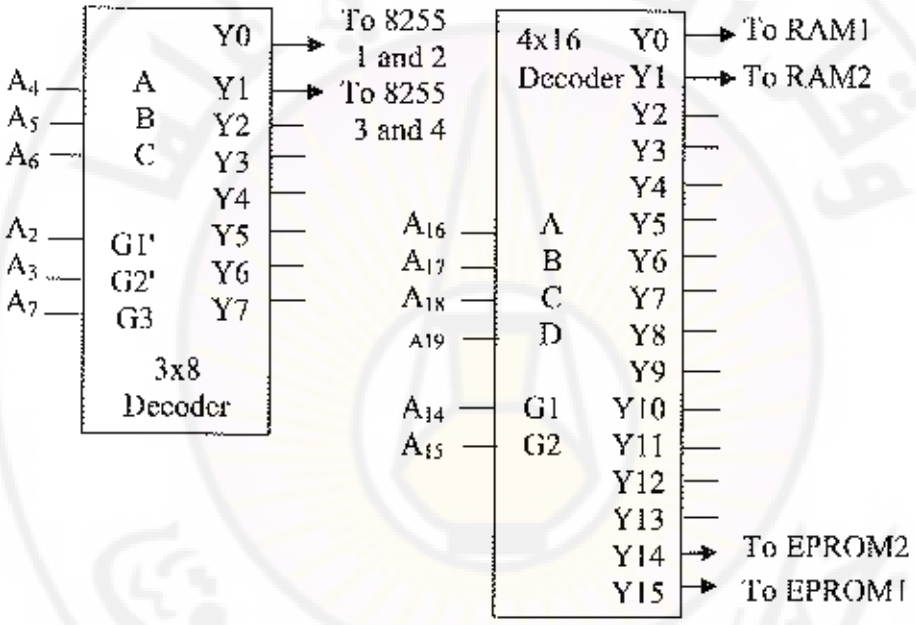
12- BREQ' - Bus Request

13- BPRN' - Bus Priority In

14- BCLK' - Bus Clock

تستخدم إشارة LOCK' كدخل مع S0', S1', S2' لمحكم الممر. بإضافة الدارتين 8288 و 8289 يحرر المعالج بعض المآخذ لاستخدامها في دعم تعدد المعالجات. ولربط الذواكر المطلوبة مع المعالج 8086 من الضروري تقسيمها إلى بنك زوجي وبنك فردي. ولتحقيق الحجم المطلوب وبالنوع المحدد من الدارات، يلزم على سبيل المثال استخدام 8 دارات من EPROM (كل منها بسعة 8 KB) و4 دارات من RAM (كل منها بسعة 32 KB). ولتشكيل بنوك الذاكرة في مثل هذه الحالة، نجعل 4 دارات EPROM ودارتان RAM للبنك الزوجي والباقي للبنك الفردي. الدارة 62256 بسعة 32 KB تحتاج للعنونة إلى 15 خط عنوان من A_1 وحتى A_{15}، بينما الدارة 2764 بسعة 8 KB تحتاج إلى 13 خط عنوان من A_1 وحتى A_{13}. الإشارتين A_{14} و BHE' تستخدمان لانتقاء البنوك الذاكرية.

وبما أن المطلوب في المثال المعطى هو بوابتان للدخل وبوابتان للخرج كل منها بعرض 16 bit، يعني أن مجمل عدد خطوط الدخل المطلوبة هو 32 مدخل وعدد خطوط الخرج المطلوبة هو أيضاً 32 مخرج. وهذا العدد من المآخذ يتطلب للعمل بنمط المصافحة 4 دارات 8255. تعطي البوابة الأولى PA من دارتي 8255 بوابة دخل بـ 16 bit (والـ 16 bit المتبقية للدخل تؤخذ من البوابة PA بالدارتين المتبقيتين). ونفس الشيء بالنسبة لبوابات الخرج، إذ إن البوابة PB من داراتي 8255 تعطي بوابة خرج بـ 16 bit. يظهر الشكل 12-11 مخطط الترميز المستخدم للذواكر والبوابات.

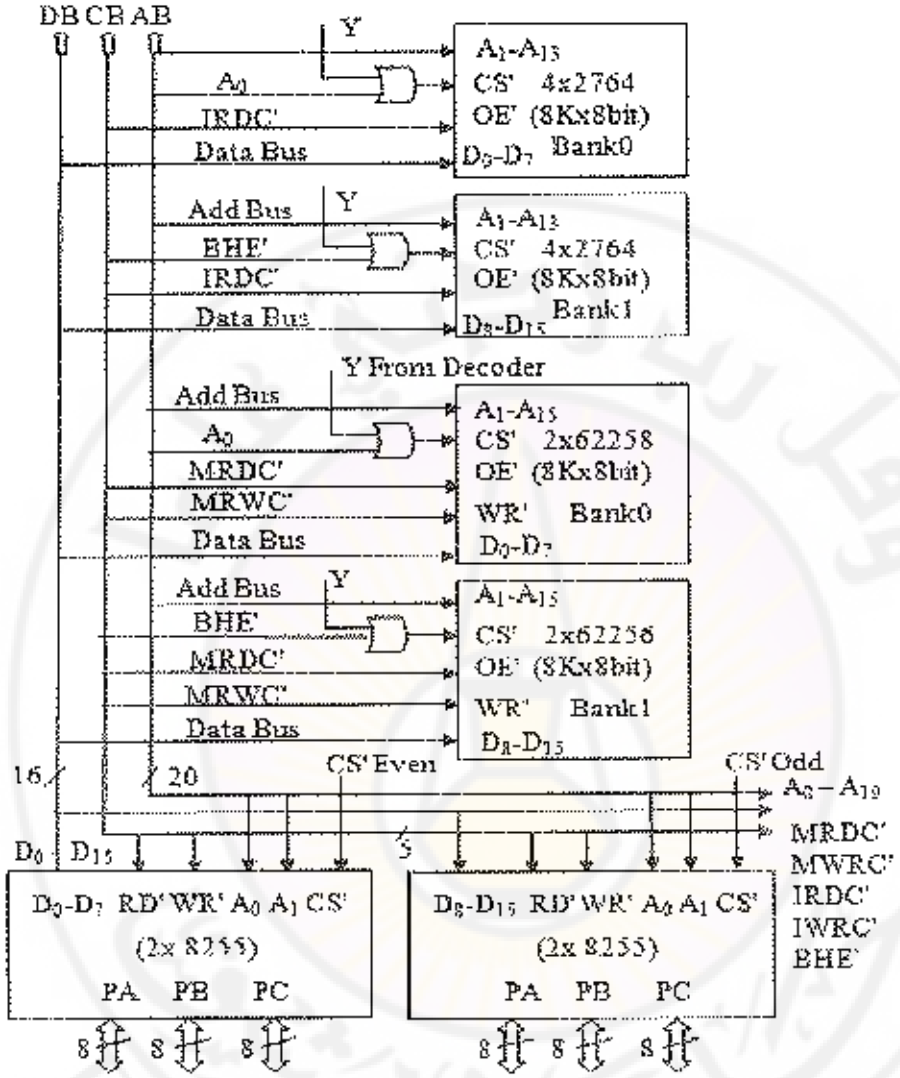


الشكل12-11: مخطط الترميز لوصل ذواكر وبوابات مع المعالج 8086

يظهر الشكل 13-11 شكل التوصيل.

حيث إن خارطة المداخل/المخارج تكون وفق ما هو مبين أدناه:

$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	Address	Address
1 0 0 0	0 0 1 1	83H	8255 -1 and 2 (CR)
1 0 0 1	0 0 1 1	93H	8255 -3 and 4 (CR)



الشكل13-11: توصيل الذواكر والبوابات بالمعالج 8086

وخارطة الذاكرة مبينة بالجدول 5-11.

الجدول5-11: ترميز 64KB ذاكرة EPROM و 128 KB ذاكرة RAM

BHE'	$A_{19}A_{18}A_{17}A_{16}$	$A_{15}A_{14}A_{13}A_{12}$	..	$A_3A_2A_1A_0$	Address	Address
1	1 1 1 1	1 1 0 0	..	0 0 0 0	FC000H	Even
1	1 1 1 1	1 1 1 1		1 1 1 0	FFFFEH	EPROM1
0	1 1 1 1	1 1 0 0	..	0 0 0 1	FC001H	Odd
0	1 1 1 1	1 1 1 1		1 1 1 1	FFFFFH	EPROM1
1	1 1 1 0	1 1 0 0	..	0 0 0 0	EC000H	Even
1	1 1 1 0	1 1 1 1		1 1 1 0	EFFFEH	EPROM2
0	1 1 1 0	1 1 0 0	..	0 0 0 1	EC001H	Odd
0	1 1 1 0	1 1 1 1		1 1 1 1	EFFFFH	EPROM2
1	1 1 0 1	1 1 0 0	..	0 0 0 0	DC000H	Even
1	1 1 0 1	1 1 1 1		1 1 1 0	DFFFE	EPROM3
0	1 1 0 1	1 1 0 0		0 0 0 1	DC001H	Odd
0	1 1 0 1	1 1 1 1		1 1 1 1	DFFFFH	EPROM3
1	1 1 0 0	1 1 0 0	..	0 0 0 0	CC000H	Even
1	1 1 0 0	1 1 1 1		1 1 1 0	CFFFE	EPROM4
0	1 1 0 0	1 1 0 0	..	0 0 0 1	CC001H	Odd
0	1 1 0 0	1 1 1 1		1 1 1 1	CFFFFH	EPROM4
...	...	...	..	...	...	...
1	0 0 0 0	0 0 0 0	..	0 0 0 0	00000H	Even
1	0 0 0 0	1 1 1 1		1 1 1 0	0FFFEH	RAM1
0	0 0 0 0	0 0 0 0	..	0 0 0 1	00001H	Odd
0	0 0 0 0	1 1 1 1		1 1 1 1	0FFFFH	RAM1
1	0 0 0 1	0 0 0 0	..	0 0 0 0	10000H	Even
1	0 0 0 1	1 1 1 1		1 1 1 0	1FFFEH	RAM2
0	0 0 0 1	0 0 0 0	..	0 0 0 1	10001H	Odd
0	0 0 0 1	1 1 1 1		1 1 1 1	1FFFFH	RAM2

١١-٣ أسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

١- بعد كل استجابة للمقاطعة Single step من قبل المعالج، فإن المؤشر Flag الذي يصفر هو

a. IF b.TF c. OF d. none of the mentioned

٢- خلال تنفيذ المقاطعة، إن المعطيات التي تدفع Pushed إلى المكدس تكون محتواة

a. IP b.CS c. PSW d. all of the mentioned

٣- كمثال عن المقاطعة الخارجية، يمكن أن يكون

a. Divide by zero interrupt b.Keyboard interrupt

c. Overflow interrupt c. Type interrupt

٤- أول معالج من عائلة Intel تضمن الذاكرة الافتراضية Virtual Memory كان

a. 80286 b.80386 c. 80486 c. pentium

٥- يستخدم في الأنظمة الصغيرة الترميز

a. Absolute Decoding b. Linear Decoding

c. Block Decoding c. All the above

٦- يستلزم ربط المعالج المساعد 8087 مع المعالج 8086 في منظومة واحدة أن يعمل المعالج 8086 بالنمط

a. Maximum Mode b. Minimum Mode

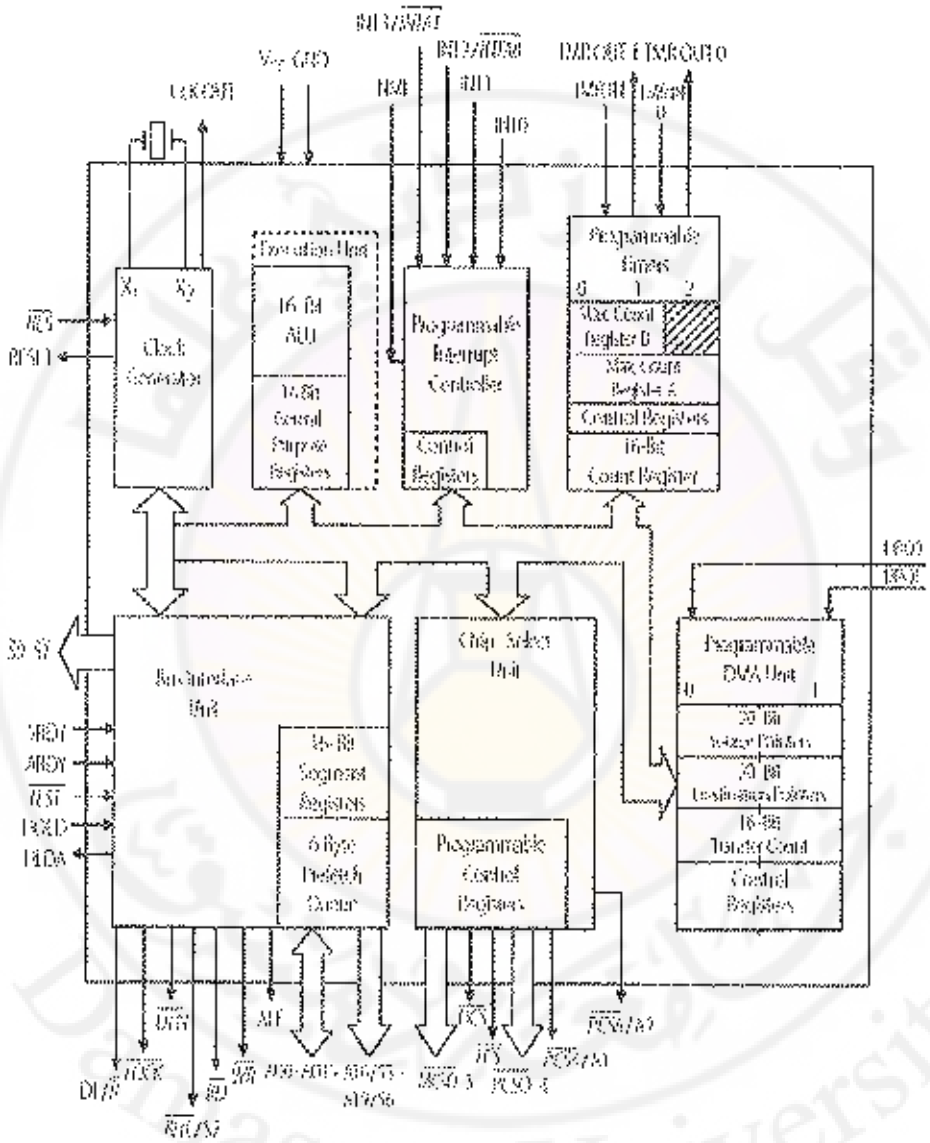
c. Maximum& Minimum Mode d. none of all

نكمل التعريف ببقية معالجات العائلة Intel 8086 Family، لكن بتفاصيل أقل، نظرا لعدم تطرقنا بهذا الكتاب إلى تصميم حواسيب صغرية باستخدام مثل هذه المعالجات الأحدث من المعالج 8086.

١٢-١ المعالج الصغري 80186/80188

سبق وأشرنا عند دراستنا لمعالجات إنتل على مستوى bit 8 أن المعالج 8085A نتج عن المعالج 8080A بإضافة بعض التحسينات التي كان من أهمها تكامل مولد النبضات الساعية 8224، ومنظم الممر 8228 مع المعالج 8080A على نفس الشريحة. وعلى مستوى المعالجات بطول bit 16 ينتج المعالجان 80186/80188 من المعالجين 8086/8088 بإضافة بعض التحسينات كتكامل بعض دارات الدعم للمعالج 8086 مع المعالج نفسه، ولذا يسمى 80186/80188 معالج بمحيطيات متكاملة 80186/80188 processor with integrated peripherals، علما إن البنية الداخلية ومجموعة التعليمات للمعالج 80186 هي نفسها للمعالج 80188 ، ماعدا أن المعالج 80188 له ممر معطيات خارجي بعرض bit 8 فقط. وبالتالي سنكتفي بالحديث عن المعالج 80186 الذي غالبا ما يدعى بمتحكم مدمج Embedded Controller بالنظر لما تم تخفيضه من عدد العناصر المحيطية اللازمة لتنفيذ منظومة متكاملة عند استخدامه أساساً في بناء الحاسوب الصغري. علاوة على أن المعالج 80186 بالمقارنة مع المعالج 8086 يمتلك العديد من المقاطعات الإضافية ولا يوجد هنا مجال لعرض هذه المقاطعات. يمكن عند استخدام المعالج 80186 العودة إلى نشرة المعطيات الخاصة به من شركة Intel . يوجد المعالج 80186 بأربع نسخ مختلفة لها نفس التركيبة الأساسية وجميعها تمثل CMOS Microprocessor، وقد تختلف نسخة عن الأخرى بوجود بعض التحسينات سوف لن ندخل هنا بتفاصيل هذه النسخ والتحسينات الحاصلة على كل منها بالنظر لقلة الأنظمة المبنية على المعالج 80186 بسبب ظهور المعالج الأحدث منه (80286). بنفس فترة ظهوره.

يبين الشكل 1-12 المخطط الصندوقي للبنية الداخلية لمختلف نسخ المعالج 80186 وما تتضمنه من محيطيات متكاملة على الشريحة.

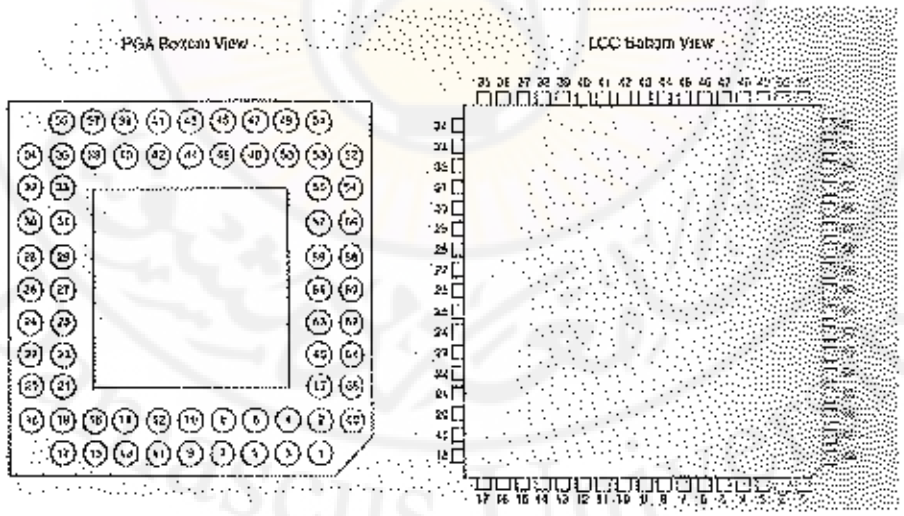


شكل1-12: التركيبة العامة للمعالج 80188/80186

وبالنظر لتكامل بعض المحيطيات مع الوحدة الأساسية CPU أصبح للمعالج 80186 بعض الميزات الأساسية المهمة ومنها وجود:

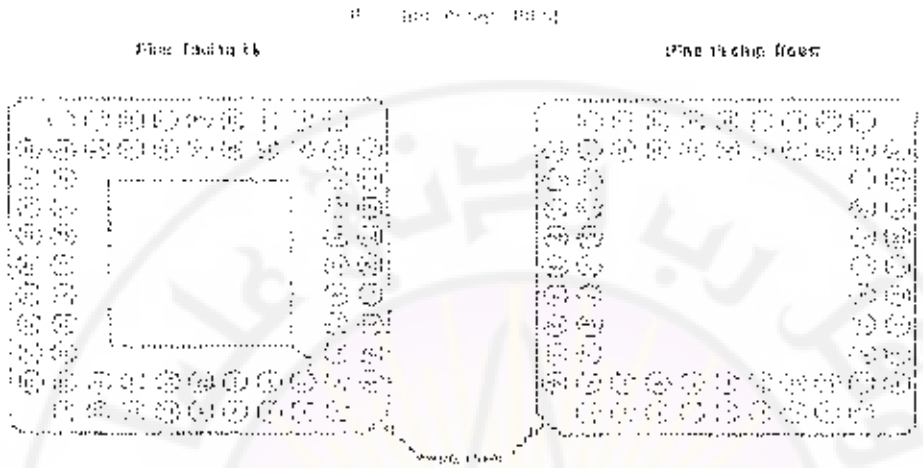
- Clock Generator
- Programmable Interrupt Controller
- Timers
- Programmable Chip Selection Unit
- Programmable DMA Unit
- Power Save/Power Down Feature
- Refresh Control Unit

يصنع المعالج 80186 على شكل دارة متكاملة IC مربعة الشكل بـ 68 Pins موزعة على الأطراف الأربعة بمعدل 17 pins على كل طرف. إن توزع المآخذ pin-out للمعالج 80186 قد يكون بواحد من تغليفين هما: إما LCC – leaded chip carrier وإما PGA – Pin Grid Array، يظهر الشكل 2-12 الوجه السفلي لمجمل مآخذ المعالج بكل من التغليفين.



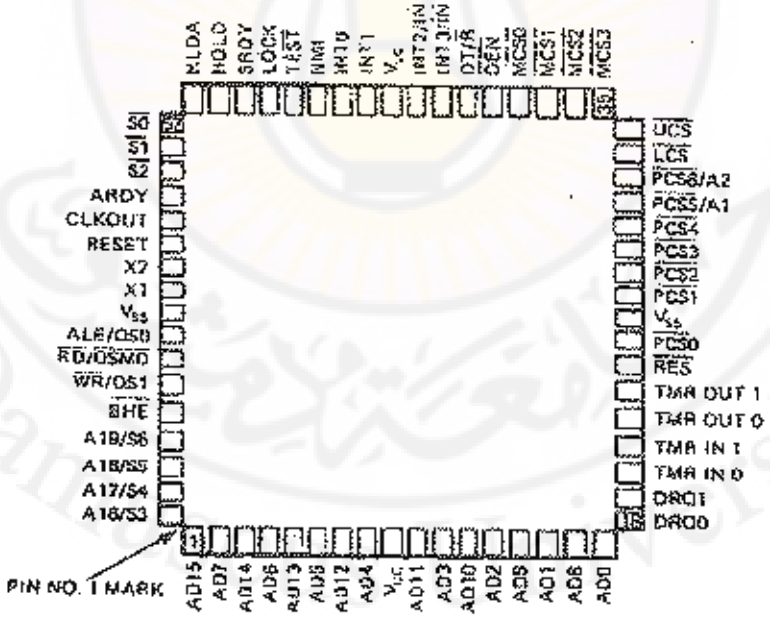
الشكل2-12: توزع مآخذ المعالج 80186 بكل من التغليفين LCC و PGA

ويظهر الشكل3-12 توزع المآخذ بكل من الوجه العلوي والوجه السفلي للمعالج 80186 بالتغليف PGA.



الشكل3-12: الوجه العلوي والسفلي لمأخذ المعالج 80186 بالتغليف PGA

ويظهر الشكل4-12 توزع إشارات المعالج 80186 على مآخذه بالتغليف LCC



الشكل4-12: توزع إشارات المعالج 80186 على مآخذه بالتغليف LCC

يملك المعالج 80186 نفس الوحدتين BIU وEU للمعالج 8086 ومع ذلك فإن المعالج 80186 بخلاف 8086 يملك مولد ساعة مبني داخليا، حيث يكفي وصل كريستال خارجي ليعمل المعالج. كما أن مفهوم معظم إشارات المعالج 80186 قد سبق ومرت معنا عند توصيفنا للمعالج 8086 وبالتالي سنكتفي بالإشارة إلى مفهوم الإشارات الجديدة فقط المرتبطة بالتحسينات التي طرأت على التركيبة الداخلية لـ 8086 ومنها:

الإشارتان: TMRIN0, TMRIN1 وتستخدمان منبعاً ساعياً خارجياً للمؤقتين 0, 1.

الإشارتان: TMROUT0 and TMROUT1 وهي تعرض إشارات الخرج من المؤقتين 0, 1 الذي يمكن برمجتها لعرض نبضات بموجات مربعة.

الإشارتان: DRQ0 and DRQ1 وتقوم بالمستوى High بقدح الطلبات على خطوط DMA للقناتين 0, 1.

الإشارات: INT0, INT1, INT2//INTA0, and INT3//INTA1 وتمثل مداخل مقاطعة مقنعة، تبرمج للتفعيل بالقدح أو بالمستوى، وتشكل أربعة مداخل للمقاطعة ان لم تكن الدارة 8259 موجودة.

الإشارة: ARDY- Asynchronous READY وهي إشارة دخل تخبر المعالج 80186 بأن الذاكرة أو البوابات جاهزة لقراءة أو كتابة المعطيات.

الإشارة: SRDY- Synchronous READY وهذا المدخل يتزامن مع ساعة النظام ليعرض التزامن مع المدخل Ready.

الإشارة: /UCS- The upper-memory chip select ويختار هذا الخرج الذاكرة على القسم العلوي من خريطة الذاكرة.

الإشارة: /LCS- The lower-memory chip select وتمكّن هذه الإشارة الذاكرة الابتداء من العنوان 00000H.

الإشارات: /MICS0-/MICS3- The middle-memory chip select وتؤهل هذه المآخذ أربعة أجهزة ذاكريه وسطية.

الإشارات: /PCS0-/PCS4 وتبرمج كخطوط انتقاء للمحيطات أو كبتات عنونة داخلية A1, A2.

الإشارتان: /PCS5/A1 and /PCS6/A2 لانتقاء الشريحة المحيطية ويمكن أن تبرمج لتقدم خط الانتقاء الخامس أو السادس.

ويلاحظ أيضاً من المخطط الصندوقي للمعالج 80186 وجود أربعة صناديق محيطية تمثل دارات دعم تم تكاملها على الـ chip . يوجد المتحكم بأولوية المقاطعة الذي يملك أربعة مداخل مقاطعة هي:

INT0 , INT1 , INT2/INTA0' , INT3/INTA1 بالإضافة إلى مدخل NM1. وإذا كانت المداخل الأربعة INT مبرمجة بنمطها الداخلي فان تطبيق إشارة على أحدها يسبب للمعالج 80186 بدفع عنوان العودة على المكدس، والتوجه مباشرة إلى بداية إجراء خدمة المقاطعة من أجل تلك المقاطعة.

تحوي التركيبة الداخلية للمعالج 80186 أيضاً على مفكك شيفرة العناوين المبني داخلياً والذي يشار له باسم " وحدة اختيار الشريحة Chip-select unit" الذي يمكن أن تبرمج لإنتاج إشارة " اختيار شريحة " حيث إن إشارات العنوان الذاكري الست المتوفرة لوحدة اختيار الشريحة هي : MCS0' ÷ MCS3' , UCS' , LCS' .

ستمكن إشارة " اختيار الشريحة السفلى LCS' بوساطة العناوين الواقعة بين 00000H وأحد العناوين الذي تحدده في كلمة التحكم، ويمكن أن يكون عنوان النهاية المحدد في أي مكان بين 1K و 256K ، ويكون العنوان الأعلى الذي سيمكن إشارة اختيار الشريحة العليا UCS' ثابتاً عند العنوان FFFFFH .

وأخيراً يوجد أربعة خطوط اختيار شريحة أوسطية Middle-chip-selected lines هي MCS0' ÷ MCS3' بحيث يمكن كل خط من هذه الخطوط الأربعة بعنوان في كتلة ذاكريه في المجال الأوسط للذاكرة، ويمكن أن يكون الحجم المحدد للكتل في أي مكان من 2K إلى 128K . لا يمكن أن تتراكب الـ Overlap أو تتقاطع المساحات الذاكرية المعينة لاختيارات شرائح مختلفة مع بعضها ، أو لا يمكن أن يكون مخرجي وحدة " اختيار الشريحة" فعّالين في نفس الوقت ، حيث إن النقطة الرئيسية لهذا المفكك المبيّت Built-in هي اختيار الكتل الرئيسية للذاكرة .

يمكن للمعالج 80186 بالإضافة إلى إنتاج إشارات "اختيار شريحة " ذاكرية أن يبرمج لإنتاج حوالي سبع إشارات " اختيار شريحة " محيطية على مآخذه 'PCS4 ÷ 'PCS0 و 'PCS5/A1 و 'PCS6/A2 ، ويمكن برمجة عنوان قاعدي من أجل عناوين الدخل والخرج I/O هذه في كلمة التحكم، وستتمكن الإشارة 'PCS0 عندما يتم إخراج هذا العنوان القاعدي أثناء تعليمة IN أو OUT.

يحتوي المعالج 80186 بداخله أيضاً على وحدة DMA قابلة للبرمجة ، حيث يوجد لهذه الوحدة مدخلان لطلب العملية DMA هما DRQ0 و DRQ1 حيث تسمح هذه المداخل لأجهزة خارجية مثل متحكمات الأقراص أو متحكمات الشاشات CRT أن تطلب استخدام أحد قنوات العملية DMA .

يمتلك المعالج 80186 من أجل كل قناة للعملية DMA مسجل كامل بطول 20 bit لحفظ عنوان المصدر لنقل معطيات DMA ومسجل بطول 20 bit لحفظ عنوان الوجهة وعداد بطول 16 bit لمعرفة عدد الكلمات أو البايتات التي تم نقلها.

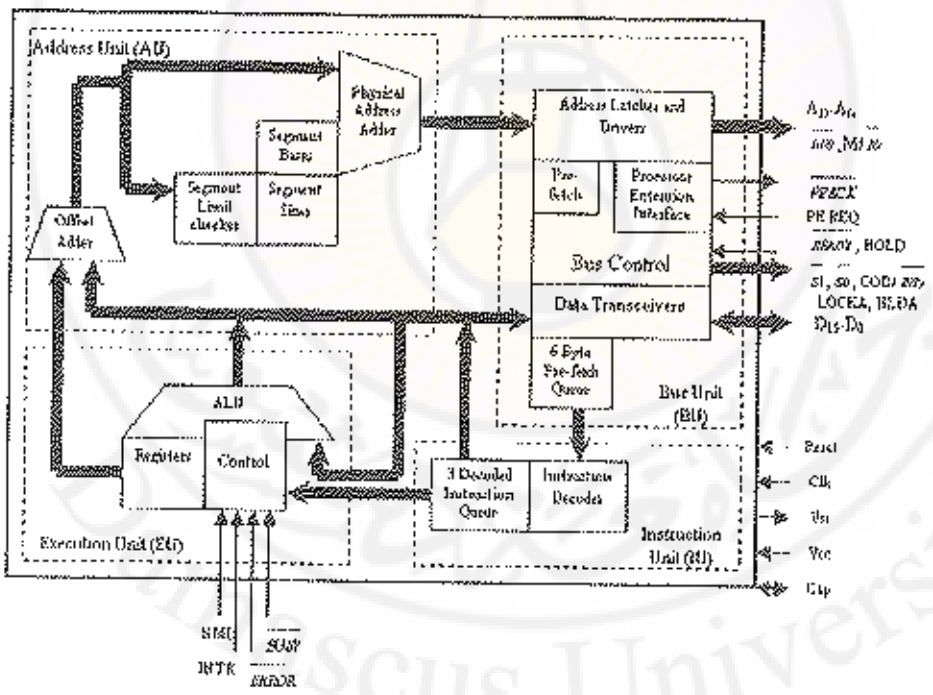
يضم المعالج 80186 أيضاً ثلاثة عدادات / مؤقتات مؤلفة من 16bit وقابلة للبرمجة. تكون مداخل ومخارج العدادين "0" و "1" متاحة على مآخذ 80186 بحيث يمكن استخدام هذين العدادين لتقسيم تردد الإشارات الخارجية لإنتاج نبضات ذات عرض مبرمج كما تفعل عدادات الدارة 8254 ، أما مدخل العداد الثالث يوصل داخليا للساعة.

ونظراً لان طريقة العدادات في المعالج 80186 تكون متناقصة ، فإن العداد 2 سيتناقص كل أربع نبضات ساعة من نبضات المعالج ، وهكذا برفع أو تصفير البتات المناسبة في كلمة التحكم تستطيع أن توجه الخرج للعداد 2 إلى مدخل DMA أو إلى مدخل المقاطعة أو إلى مدخل العداد و/ أو العداد 0 .

وعلى الصعيد البرمجي فقد زود المعالج 80186 ببعض التعليمات الإضافية عن المعالج 8086.

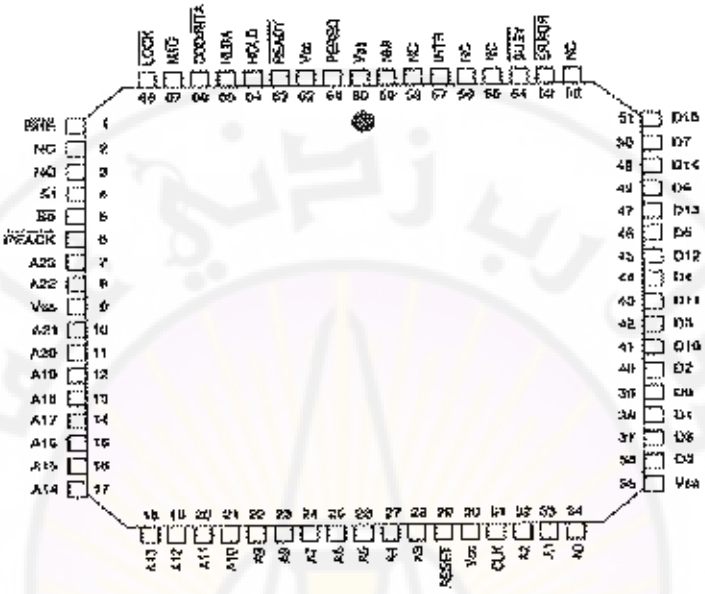
٢-١٢ المعالج الصغري INTEL 80286

يمثل المعالج الصغري 80286 الجيل الثاني لمعالجات العائلة Intel 8086 Family ففي عام 1984 قدمت شركة IBM جهاز الكمبيوتر IBM/PC AT الذي يستخدم المعالج 80286والرقاقة 80287 كمعالج حسابي. وبمقارنة هذا الحاسوب الذي يعمل بممر للمعطيات بعرض 16 bit مع الحاسوب IBM PC (المبني على أساس المعالج 8088) يعدّ أقوى من 3 إلى 4 أضعاف بسبب ضعف حجم ممر (ناقلة) البيانات في النظام وقدرة المعالج على العمل بتواتر 6MHz أو 8MHz بدلا من 4.77 MHz . أنتج المعالج الصغري 80286 (اختصارا 286) بنفس الوقت الذي ظهر فيه المعالج 80186 وقد احتوى زيادة عنه في التركيبة الداخلية له على دارات الحماية ودارات إدارة الذاكرة الافتراضية Virtual Memory Management Circuit. يبين الشكل 5-12 التركيبة الداخلية للمعالج 80286.



شكل5-12: المخطط الصندوقي الداخلي للمعالج 80286

يصنع المعالج 286 بالتغليف PGA ويملك ممر عناوين بعرض 24 Bits غير مضاعف Non Multiplexed ما يتيح له عنونة ذاكرة رئيسية بسعة 16 MB. يبين الشكل 6-12 الصندوق المؤلف من 68 Pin للمعالج 80286.



الشكل6-12: توزع إشارات المعالج 80286

يلاحظ من الشكل أعلاه أن عدد مآخذ المعالج 286 مماثلا لعدد مآخذ المعالج 80186 وأن اسم ومفهوم كل من إشارات المعالج 286 قد سبق ومر معنا أثناء دراسة المعالجات الأقدم من إنتل ولا داعي هنا لإعادة التعريف بأي منها.

يعدّ المعالج 286 أول معالج بالعائلة 8086 Family صمم خصيصا للاستخدام كوحدة معالجة مركزية في الحواسب الصغرية متعددة المستخدمين، وبالتالي هو أول معالج يحقق كل ميزات أنظمة التشغيل متعدد المستخدمين/ متعدد المهام بصورة سهلة نسبياً (أي المحافظة على البيئة أثناء تبادل المهام، وحماية المستخدم ونظام التشغيل وكذلك حماية الذاكرة الافتراضية). يظهر من المخطط الصندوقي للمعالج 286 بأن المعالج يحتوي على أربع وحدات منفصلة للمعالجة هي:

- الأولى: تنفذ وحدة الممر BU (Bus Unit) في المعالج كل كتابات وقراءة الذاكرة والمدخل والمخرج I/O وتجلب مسبقا بايتات التعليمات، وتشرف على نقل المعطيات بين المعالج وأجهزة توسيع المعالج مثل المعالج المساعد 80287.

- الثانية: تفكك وحدة التعليمات IU (Instruction Unit) بشكل كامل حتى ثلاث تعليمات مجلوبة مسبقا، وتحتفظ بها في الرتل بحيث يتثنى لوحدة التنفيذ الولوج إلى هذه التعليمات في أي وقت، وهذا مثال إضافي للمعالجات الحديثة عن كيفية الاحتفاظ بعدة تعليمات (في الخط الأنبوبي للتعليمات بدلا من الانتظار لإنهاء تعليمة واحدة قبل جلب التعليمات التالية .

- الثالثة: تستخدم وحدة التنفيذ EU (Execution Unit) وحدتها ALU المكونة من 16 Bit لتنفيذ التعليمات التي تستلمها من وحدة التعليمات. وعند تشغيل المعالج 286 في نمط العنوان الحقيقي تصبح مجموعة مسجلاته هي نفسها كما في المعالج 8086 باستثناء المسجل الإضافي MSW المسمى بمسجل كلمة أحوال الآلة Machine Status Word ذي الطول 16 Bit.

- الرابعة: تحسب وحدة العناوين Address Unit (AU) العنوان الفيزيائي الذي سيرسل إلى الذاكرة أو I/O بوساطة الوحدة BU .

يستطيع المعالج 286 أن يعمل في أحد نمطي العناوين الذاكرية الحقيقي والمحمي، بحيث إذا عمل المعالج 286 في نمط العنوان الحقيقي، فإن وحدة العناوين ستحسب العناوين باستخدام قاعدة القطعة Base والإزاحة Offset ، أي تستخدم المسجلات المألوفة CS, SS, DS, ES لحفظ العناوين القاعدية للقطع الحالية في الاستخدام، ويكون المجال الأعظمي للعناوين الفيزيائية في هذا النمط مساويا 1MB.

أما إذا عمل المعالج بالنمط المحمي فتستعمل وحدة العناوين كما لو كانت وحدة MMU كاملة وبالتالي فإنه سيستخدم كل خطوط العنونة (وعددها 24)، ويستطيع بهذا الشكل العمل بمجال ذاكري قدره 16MB بدلا من إمكانية عنونة 1MB فقط في النمط الحقيقي تجهز المكونات الصلبة الذاكرية للمعالج 286 كبنك فردي Odd Bank وآخر زوجي

Even Bank كما يحصل تماما في المعالج 8086. وسيمكن Enable البنك الزوجي عندما يكون $A_0 = 0$، بينما يمكن البنك الفردي عندما يكون BHE' = 0 (وللولوج إلى كلمة على نفس النسق aligned word سيكون BHE' = A_0 = 0). تفكك شفرة إشارات الأحوال S_0 ' و S_1 ' و M/IO' بوساطة متحكم الممر الخارجي 82288 للحصول على ممر التحكم CB وإشارات القراءة والكتابة والاعتراف بالمقاطعة Interrupt ACK . تعمل الإشارات INTA' , LOCK' , READY' , RESET INTR , NMI, HLDA , HOLD بشكل أساسي بنفس الصيغة التي تعمل بها على المعالج 8086. ويستخدم مولد الساعة الخارجي 82284 لإنتاج إشارة نبضة الساعة ولمزامنة الإشارتين RESET, READY'.

يوجد أربع إشارات PEREQ, PEACK', BUSY', ERROR' تلزم للربط مع توسعات المعالج كالربط مع المعالج الحسابي 80287 وهي تعمل وفق التالي:

- يمكّن مدخل PEREQ (Processor Extension Request) " طلب توسّع المعالج" بوساطة المعالج المساعد وذلك لإخبار المعالج 286 القيام بنقل معطيات من الذاكرة تخصص المعالج المساعد، وعندما يسعى المعالج 286 للقيام بعملية النقل فإنه
- يمكّن إشارة PEACK' (Processor Extension Acknowledge) " الاعتراف بتوسّع المعالج" لتصل إلى المعالج المساعد، ولتجعله يعرف أن نقل المعطيات قد بدأ. إذ تحدث عمليات نقل المعطيات بهذه الطريقة من خلال المعالج 286، وبالتالي فإن المعالج المساعد يستخدم إمكانيات الحماية والذاكرة الافتراضية الوحدة MMU .
- تعمل إشارة المدخل BUSY' على المعالج 286 بنفس الطريقة التي يعمل بها المدخل $TEST_1$ على المعالج 8086، عندما ينفذ المعالج 286 تعليمة الانتظار Wait فإنه سيبقى في حلقة Wait حتى يجد الإشارة BUSY' من المعالج المساعد مرتفعة، وإذا وجد المعالج المساعد أحد الأخطاء فإنه سيمكّن
- المدخل ERROR' للمعالج 286، وهو ما يسبب تنفيذ استدعاء المقاطعة من النوع 16 H بشكل آلي .

وبالرغم من أن مختلف التحسينات التي حدثت للمعالج 286 بالمقارنة مع المعالج السابق له 80186 تركزت على التركيبة الداخلية للمعالج، إلا أن مسجلات المعالج بقيت نفسها. وهنا نشير فقط إلى مسجل الأعلام Flags للمعالج 286 الذي طور عن مسجل الأعلام لدى المعالج 8086 بثلاث مؤشرات كالتالي:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	NT	IO	PL	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

تسمى المؤشرات الجديدة بالمؤشرات الخاصة Special Flags، وهي PL و IO وتسمى I/O Privilege Level و NT وتعني Nested Task Flag. وعلى الصعيد البرمجي، يمتلك المعالج 286 مجموعة تعليمات جديدة يشار لها ولمجموعة تعليمات المعالجات الأحدث بالمجموعة المتفوقة " Superset " لتعليمات المعالج 8086. وقد صمم المعالج 286 من الناحية البرمجية بالشكل الذي يسمح بتشغيل كافة برامج المعالج 8086 وما تبعه من معالجات إنتل، حيث تمت المحافظة على التوافقية بين معالجات العائلة 8086 Family بعد الأخذ بعين الاعتبار أن البرامج المطورة للمعالج 286 قد لا تعمل على حواسيب مبنية على معالجات أقدم منه.

ينفذ المعالج 286 بنفس تردد إشارة الساعة معظم البرامج التي ينفذها المعالج 8086 بشكل أسرع عدة مرات. تقسم التعليمات الجديدة أو الإضافية للمعالج 286 إلى نوعين، إحداها يتعلق بتعليمات النمط الحقيقي Real Mode Instruction والأخرى من التعليمات مخصص للنمط المحمي Protect Mode Instruction، وبالنظر إلى أن النمط المحمي لعمل المعالج قد تم اعتماده لدى معالجات إنتل ابتداء من المعالج 286، نشير فيما يلي بشكل مختصر إلى كل من هذين المفهومين.

١٢-٢-١ مفهوم النمط الحقيقي للمعالج 286

يبدأ المعالج 286 بالتنفيذ في نمطه الحقيقي بعد عملية تصفيره RESET، ويشار لهذا النمط بهذا الاسم، لأن عناوين الذاكرة الفيزيائية يتم الحصول عليها مباشرة بجمع الإزاحة مع قاعدة القطعة كما في المعالج 8086، حيث يستطيع المعالج في هذا النمط أن يعنون حتى 1MB من الذاكرة الفيزيائية، ويعمل بشكل أساسي وكأنه معالج 8086 بفعالية أكثر وفقاً لبعض التحسينات الأخرى في المكونات الصلبة للمعالج. عند عمل المعالج 286 بنمط العنوان الحقيقي، تكون الاستجابة للمقاطعة هي نفسها كما لو كانت للمعالج 8086 .

وهنا نشير بخصوص المقاطعات لدى معالجات إنتل ابتداء من المعالج 80186 إلى أن المعالجات تقسم هذه المقاطعات إلى صنفين رئيسيين، هما: المقاطعات Interrupt والاستثناءات Exceptions

يشار إلى الحوادث الخارجية غير المتزامنة Asynchronous التي تؤثر على المعالج من خلال المداخل INTR أو NMI عادة بالمقاطعات.

أما المقاطعة من نوع استثنائي (Exception--Type Interrupt) فتولد عن طريق أحد شروط الخطأ التي تحدث أثناء تنفيذ التعليمات . فمثلاً يعدّ التقسيم على صفر مثالاً عن التعليمة التي تسبب استثناء .

وهكذا تصنف المقاطعات البرمجية الناجمة عن التعليمة INT n على أنها "استثناءات" لأنها تكون متزامنة Synchronous مع المعالج.

والاستثناءات تقسم بدورها أيضاً إلى نوعين هما: أخطاء Faults ومصائد Traps. فالأخطاء هي استثناءات يتم اكتشافها والإعلان عنها قبل تنفيذ التعليمة الخاطئة. فمثلاً يمكن اعتبار استثناء عدم وجود القطعة Segment not-- present على أنه مثال لخطأ Fault ، أما المصائد فهي الاستثناءات التي يعلن عنها بعد تنفيذ التعليمة التي سببت الاستثناء، فمثلاً استثناء التقسيم على الصفر ومقاطعات التعليمة INT n تعدّ أمثلة على هذه المصائد Traps .

٢-٢-١٢ مفهوم النمط المحمي للمعالج 286

إذا كان الحاسوب المبني على المعالج 286 يعمل تحت نظام التشغيل MS-DOS فإن الحاسوب يترك للعمل بنمط العنوان الحقيقي، لان إصدارات النظام DOS لم تصمم للاستفادة من ميزات النمط المحمي للمعالج 286.

أما إذا كان الحاسوب يعمل تحت نظام تشغيل مثل Microsoft OS/2 الذي يستخدم النمط المحمي، فإن النمط الحقيقي سيستخدم فقط لتهيئة initialize الأجهزة المحيطية ولتحميل الجزء الأساسي لنظام التشغيل من القرص إلى الذاكرة، ولتحميل بعض المسجلات ولتمكين المقاطعات ولتجهيز جداول الواصفات ولنقل المعالج من النمط الحقيقي إلى النمط المحمي .

الخطوة الأولى في عملية نقل المعالج 286 إلى النمط المحمي هي رفع بت تمكين الحماية Protection Enable Bit بواحد في مسجل (كلمة أحوال الآلة MSW) لدى المعالج 286، أما الخطوة الأخيرة لجعل المعالج 286 يعمل في النمط المحمي، فهي تنفيذ قفز خارج القطعة Intersegment إلى بداية برنامج النظام الأساسي. وهذا القفز ضروري لتسوية Flush ريتل بايتات التعليمات، لان الرتل في النمط المحمي يعمل بطريقة مختلفة عن طريق عمله في النمط الحقيقي.

إن انتقال المعالج 286 إلى النمط المحمي يمكن Enable الوحدة المتكاملة MMU بغية تأمين الحماية والذاكرة الافتراضية. ويمكن العودة من النمط المحمي إلى النمط الحقيقي بتصفير النظام.

بما أن أهم تصنيف للمعالجات الصغرية يكون بحسب عرض ناقل البيانات، فقد أطلق على معالجات إنتل ابتداء من المعالج 80386 وحتى Pentium4 مصطلح IA32 (أي Intel Architecture, 32 bit)، بينما المعالج Itanium تم تصنيفه بالمصطلح IA64 (Intel Architecture, 64 bit) وهو أول معالج عرض ناقله 64 bit.

كان من بين القيود الرئيسية للمعالج 286 أنه يملك وحدة ALU بعرض 16 Bits وأن حجم قطعه الأعظمي 64 KB ولا يمكنه الانتقال مجيئا وذهابا بسهولة بين النمط الحقيقي والنمط المحمي، ولهذا صممت معالجات جديدة من إنتل للتغلب على هذه القيود مع المحافظة على توافقية المكونات اللينة مع المعالج 286. ونتعرف فيما يلي من هذه المعالجات الجديدة معالجين، هما المعالج 80386 و المعالج 80486 والذي كل منهما يمثل معالجاً بممر معطيات بعرض 32 Bits وخطوة تالية من خطوات تطور العائلة 8086 Family.

١٣-١ المعالج الصغري 80386

يمتلك المعالج 80386 (اختصارا 386) أوامر إضافية تسمح له باستخدام نظام تشغيل متعدد الأعمال، مثل نظام التشغيل Windows/286 الذي كان محدوداً بأوامر البيانات بطول 16 bit، وقد عدّ المهتمون بعالم المعالجات ظهور المعالج 386 بأنه ثورة انطلقت شرارتها لتعلن عن تكنولوجيا جديدة في عالم المعالجات. صنف ظهور المعالج 386 في العام 1986 بالجيل الثالث للعائلة 8086 Family وبالنظر إلى وجود اختلاف بسيطة بين معالجات الـ 386 في أن بعضها يتعامل مع الأوامر ذات الـ 16 bit والبعض الآخر يمكنه التعامل مع أوامر الـ 32 bit فإن Intel قامت بوضع علامة مميزة على قمة رقاقة المعالج للتمييز بين نوعي معالجات الـ 386، فبالنسبة لمعالج أوامر 16 bit تم تمييزه عن طريق العلامة " 16 – bit SW only "

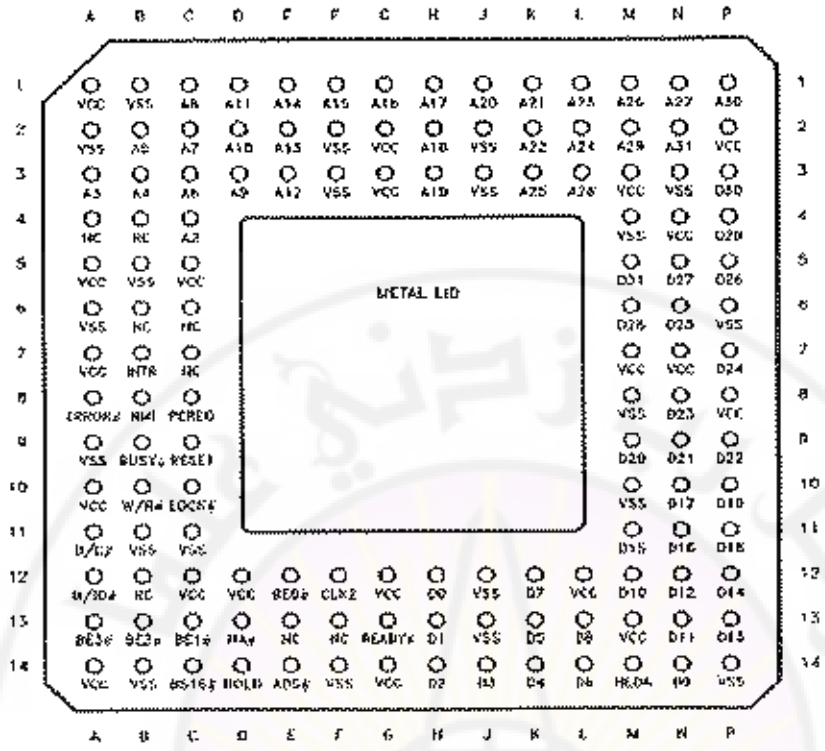
وسمي بالمعالج 386 SX أما بالنسبة للمعالج الذي لا يجد مشاكل عند التعامل مع أوامر الـ 32 bit فقد تم تمييزه بالعلامة " ∑∑ " وسمي بالمعالج 386 DX. للمعالج 386 SX

نفس القدرات الداخلية التي يمتلكها المعالج 386 DX أما خارجياً فكانت ناقلة بياناته بعرض 16 bit وناقلة العناوين بعرض 24 bit أي أن الناقلات الخارجية لهذا المعالج هي بنفس حجم الناقلات الخارجية للمعالج 286 .

أما بخصوص المعالج 386 DX فيتضمن وحدة ALU بعرض 32 bits، وبالتالي يمكنه العمل مباشرة مع كلمات بطول 32 bits، أي أن ناقل بياناته بعرض 32 bit وناقل العناوين Address Bus بعرض 32 bit أيضاً، وهذا ما يمكنه من عنونة مواقع فيزيائية لذاكرة بحجم 4 GB = 4096 MB أي ما يعادل 256 ضعف حجم الذاكرة للمعالج 286. يمكن لقطعة المعالج 386 DX أن تكون ضخمة إلى حدود 4 GB وباستطاعة البرنامج امتلاك العديد من القطع segment التي يصل عددها إلى 16384 قطعة. وهكذا يصبح فراغ العناوين الافتراضي للمعالج 386 DX مساوياً = 4 GB x 16384. 64 Tera Bytes

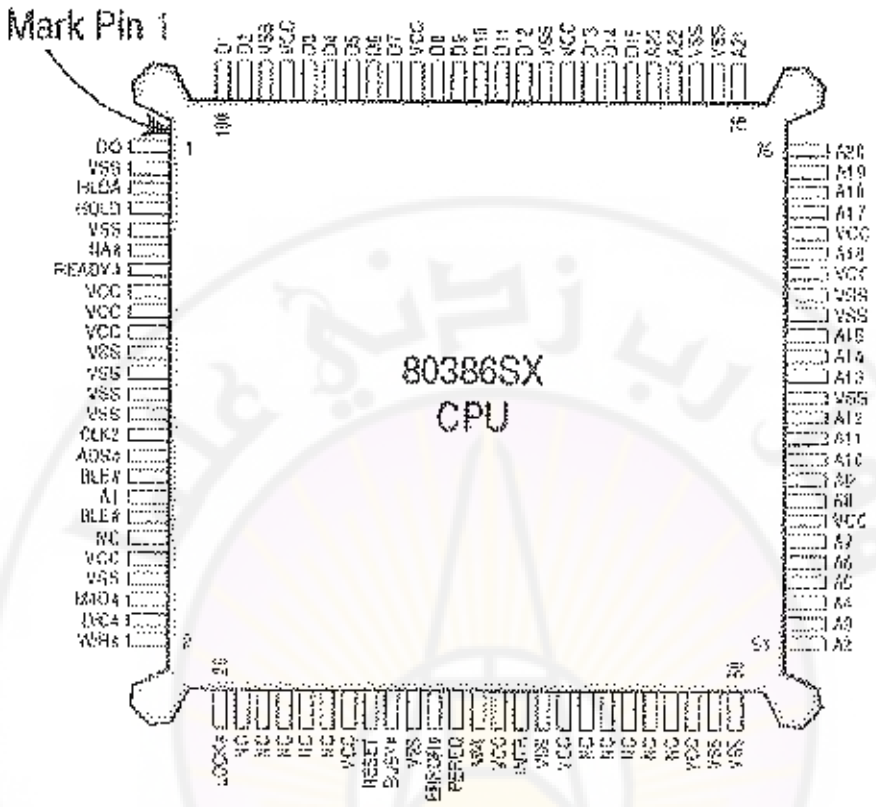
يتمتع المعالج 386 أيضاً بنمط المعالج 8086 الافتراضي Virtual 8086 Mode الذي يسمح له بالانتقال مجيئا وذهابا بين مهام النمط المحمي للمعالج 386 ومهام النمط الحقيقي للمعالج 8086 بسهولة .

ظهر مع المعالج 386 SX الرقاقة 80387 التي تمثل المعالج الحسابي المرتبط معه، بينما الرقاقة 80387 DX تمثل المعالج الحسابي المرتبط بالمعالج الصغري 80386DX. معالجات الـ 386 ذات السرعات العالية 33 MHz أو 40 MHz لم تستطع العمل إلا بنصف سرعتها بسبب بطء ذاكرة RAM الرئيسية من نوع DRAM ، ولذلك اضطرت الشركة الصانعة من إضافة نوع سريع من ذواكر RAM على اللوحة الأم من نوع Static RAM بسعة 64 KB، وأدى هذا الإجراء إلى تمكين المعالج من أن يعمل بسرعته الكاملة % 90من الوقت مقابل بعض الزيادة في التكاليف. تم تجميع مختلف إشارات المعالج 386 DX بما في ذلك ممر العناوين وممر المعطيات في صندوق سيراميكي بـ 132 pin متوضعة على شكل مصفوفة شبكية المآخذ وفق ما هو مبين بالشكل 1-13.



الشكل 1-13 المصفوفة الشبكية للمعالج 386 DX التي تبين مآخذه

يلاحظ من توزع مآخذ المعالج 386 DX وجود عدد كبير من مآخذ التغذية V_{cc}، وكذلك خطوط الأرضي V_{ss}، التي يجب وصل كل منها إلى قيمتها المقابلة. ووجود عدد من المآخذ NC التي لا توصل بشيء. أما بالنسبة للمعالج 386 SX الذي له نفس البنية الداخلية للمعالج 386 DX بممر عناوين بعرض 24bit فقط، وممر معطيات 16 bit فقط (كما هو الحال في المعالج 286)، فإن كافة إشاراته متوضعة بصندوق من 100 pin فقط وفق ما هو مبين بالشكل 2-13. وكما هو الحال بالنسبة للمعالج 386 DX يوجد لدى المعالج 386 SX العديد من المآخذ للتغذية الكهربائية ومآخذ NC. وبالتالي الحديث سيدور من الآن وصاعدا حول المعالج 386 الذي أصبحت معظم إشاراته مألوفة المعنى من خلال المناقشات السابقة لمعالجات من نفس العائلة.



الشكل2--13: توزع إشارات المعالج SX 386 على مأخذه

تقسم إشارة الساعة المطبقة على المدخل CLK2 داخليا على 2 باستخدام الدارة 82384 وذلك لإنتاج إشارة الساعة التي تقود كل عمليات المعالج فعليا. يتألف سمر العناوين في المعالج 386 من خطوط العناوين A_2 – A_{31} ومن خطوط تمكين البايتات BE'_0 – BE'_3 (Byte Enable line). تفكك إشارة الخطوط BE'_0–BE'_3 من إشارات العنوان الداخلية A_0 و A_1 وتعمل بشكل مشابه جدا لطريقة التي يعمل بها الخطان A_0 و BHE' لدى المعالج 8086 حيث يمكن أن تجهز الذاكرة الرئيسية كبنوك ذاكريه بعرض 4 Bytes. وفي هذه الحالة تعمل الإشارات BE'_0–BE'_3 كخطوط تمكين لهذه البنوك الأربعة مما يسمح للمعالج 386 بنقل بايتات أو كلمات أو كلمات مضاعفة من الذاكرة وإليها .

تعين إشارات تعريف دورة الممر BUS Cycle definition Signals نوع العملية التي تحدث أثناء دورة الممر. وتشير الإشارة 'WR/R إلى أن العملية الجارية الآن هي إما عملية قراءة أو كتابة، كما تشير الإشارة 'D/C إلى أن عملية الممر هي إما كتابة / قراءة معطيات أو نقل كلمة تحكم مثل جلب شفرة العملية op-code. وتشير الإشارة 'M/IO إلى أن العملية هي عملية ذاكرة أو عملية دخل / خرج مباشر. إن بنية منافذ الدخل /الخرج المباشر للمعالج 386 ما هي إلا توسيع Extension لبنية المنافذ في المعالج 8086 بغرض الحصول على منافذ بعرض 32 bits. إذ يمكن تشكيل منافذ I/O بعرض 32 bits بسهولة بتوصيل منفذا I/O بعرض 8 bit على التوازي في دارات مثل الدارة 8255. يستطيع المعالج 386 استخدام التعليمة IN أو OUT المتبوعة بعنوان المنفذ ذي العرض 8 bits، وذلك لعنونة 256 منفذا بعرض 8bit أو 128 منفذا بعرض 16 bits أو 64 منفذ بعرض 32 bits. لكن استخدام المسجل DX للاحتفاظ بعناوين عرضها 16 bits سيمكن المعالج 386 من الولوج إلى 64K منفذا بعرض 8 bit أو 32K منفذ بعرض 16 bits أو 16K منفذ بعرض 32 bits.

يتم إخراج الإشارة PEREG من قبل معالج مساعد (مثل معالج الفاصلة العائمة 80387) وذلك لإخبار المعالج 386 بجلب الجزء الأول لكلمة المعطيات من أجل المعالج المساعد، وبعد ذلك سيتولى المعالج المساعد قيادة الممرات ويقرأ تتمة كلمة المعطيات كما هو الحال مع المعالج 8087.

تستخدم الإشارة 'BUSY من قبل المعالج المساعد لمنع المعالج 386 من متابعة تعليمته التالية قبل أن ينتهي المعالج المساعد من التعليمة الحالية . تعمل المداخل Reset , NMI , INTR , HOLD , HLDA بشكل مشابه للطريقة التي تعمل بها في المعالجات السابقة المجموعة الأخيرة لإشارات المعالج 386 تمثل مجموعة Bus Control Group التي تضم إشارات التحكم بالممر منها.

الإشارة 'Ready مثلا تستخدم لغرس حالات الانتظار Wait states في دورات الممر وذلك عند الحاجة للربط مع أجهزة دخل وخرج أو مع ذواكر بطيئة. وكذلك المدخل 'BS16 يسمح للمعالج 386 بالعمل مع ممر معطيات بعرض 16 bit و/أو ممر بعرض 32 bits

فإذا كان المدخل BS'16 ممكنا Enabled فإن المعالج 386 سينقل المعطيات على النصف السفلي فقط لممر المعطيات ذي العرض 32 bits، وإذا كان المدخل BS'16 ممكّنا وكان المطلوب قراءة معامل بعرض 32 bits من ذاكرة عرضها 16 bits ، فسيولد المعالج 386 آليا دورة ممر ثانية لقراءة الكلمة الثانية .

ستمكّن الإشارة 'ADS عندما تكون العناوين الصحيحة ، وإشارات تمكين البايتات BE وكذلك إشارات تعريف دورة الممر جميعا موجودة على الممرات .

إن ممر عناوين المعالج 386 ليس مضاعفا Not Multiplexed وبالتالي لا يوجد ضرورة لاستخدام إشارة ALE .

تستخدم الإشارة 'ADS في بعض المنظومات المعتمدة على المعالج 386 وذلك لنقل العنوان إلى مخارج الماسكات الخارجية للعناوين، وهذا ما يدعى بالتخطيط الأنبوبي للعناوين Address pipelining . إذ ينص هذا المبدأ على أنه إذا حفظ أحد العناوين على مخارج الماسكات الخارجية ، يمكن للمعالج 386 عندئذ أن يزيل (أي يلغي) العنوان القديم عن مأخذ عناوينه ويخرج العنوان الجديد من أجل العملية التالية في دورة الممر. حيث تمكن جملة دارات التحكم الخارجية إشارة العنوان التالي 'NA لتخبر المعالج 386 بإخراج عنوان العملية التالية في اللحظة المناسبة .

يحتوي المعالج 386 على كمية ضخمة من دارات الفحص الذاتي المبيتة BIST (Built –In self–Test Circuitry) فإذا حافظ المدخل 'BUSY على قيمته منخفضا عندما يكون Resel منخفضا سيقوم المعالج 386 آليا بفحص 60% تقريبا من دارته الداخلية ويتطلب الفحص الذاتي حوالي 2^{20} دورة على المدخل CLK2 . إذا مرّر المعالج 386 كل الاختبارات بشكل صحيح فانه سيترك توقيعا Signature كله أصفار في المسجل EAX دلالة على ذلك. ومن مخططات الممرات المستخدمة في الأنظمة 386 التي تستخدم لربط الألواح (البطاقات) المحيطية مع اللوحة الأم نذكر الممرات التالية:

– الممر PC/AT المستخدم في حواسيب IBM والذي يسمى اختصارا ISA Industry أي البنية القياسية الصناعية وهو أحد الممرات المستخدمة في أنظمة المعالج 386 لربط الألواح المحيطية مع اللوحة الرئيسية في هذه الأنظمة .

– البنية القياسية الصناعية الموسعة EISA من

Extended Industry standard Architecture

– بنية القناة المايكروية أو الصغرية Micro Channel Architecture من IBM والتي تسمى بالممر MCA.

يملك الممر ISA فقط 16خط معطيات و 24 خط عناوين وبالتالي لا يستطيع الاستفادة الكاملة من ممر المعطيات أو العناوين للمعالج 386 التي هي بعرض 32 bit وهذا ما يقلل من سرعة نقل المعطيات على الممر ، وبالتالي يستخدم الممر ISA في بعض الأنظمة فقط للاتصال بالمحيطات مثل ألواح متحكمات الأقراص والواح متحكمات الشاشة والواح ربط الشبكة.

وبما أن الممر ISA لا يصلح للتطبيقات ذات الأداء العالي مثل مخدمات ملفات الشبكة (Network File Servers) ومخدمات الاتصال وغيرها من الأنظمة متعددة المستخدمين/ متعددة المهام تم اللجوء إلى استخدام الممرات EISA أو MCA يستخدم EISAعلى أنه توسع للممر ISA بهدف تلبية حاجيات المعالج 386 وكذلك الأنظمة متعددة الأسياد Multimaster System وقد طور هذا الممر من قبل تسع شركات ليكون بديلا للممر القياسي MCA المقدم من شركة IBM.

يستخدم الممر EISA موصلات Connectors بنفس الحجم الفيزيائي للممر ISA وهذا يسمح بغرس ألواح EISA, ISA في الفتحات Slots المخصصة لذلك ضمن الحاسوب، إلا أن موصلات EISA لها مستويان من التماسات Contacts. وإذا ادخل أحد الألواح المعتمدة على الممر ISA فإن هذا اللوح سيدخل في الموصلات القريبة فقط ليصل إلى المستوى العلوي للتماسات التي تحتوي على إشارات الممر ISA. بينما يسمح الثلم الموجود في ألواح EISA بدخول الموصلات بشكل كافٍ لتتلامس أيضاً مع المستوى السفلي للتماسات. حيث تحتوي هذه التماسات السفلية على الإشارات الإضافية للممر EISA. وهنا نشير إلى أن ممر بنية القناة المايكروية MCA من IBM يحتوي على نفس أنواع الإشارات للممر EISA كما انه ينجز نفس الوظائف لكن الممرين غير متوافقين نهائيا إذ أن الواح الممر MCA أصغر وتستخدم موصلات Connectors مختلفة.

مسمم الممر MCA للعمل مع الألواح المحيطية التي تنقل المعطيات المؤلفة من 8 Bits , 16 Bits , 32 Bits. حيث يستخدم للألواح المحيطية ذات الـ 8 Bits مثلا موصل له 46 Pin فقط ويستخدم للألواح ذات الـ 16 Bits موصل إضافي له 12 Pin، ومن الواح ذات 32 Bits يستخدم موصل إضافي له 44 Pin. وبخصوص أنماط تشغيل المعالج 386 يلاحظ وجود ثلاثة أنماط للمعالج هي نمطا التشغيل الحقيقي والمحمي (نفس النمطين الموجودين لدى المعالج 286) والنمط الافتراضي Virtual 8086 Mode ، ونشير فيما يلي باختصار إلى كل منها.

١٣-١-١٠ النمط الحقيقي للمعالج 386

يعمل المعالج 386 بالنمط الحقيقي بعد التصفير مباشرة وكأنه معالج 8086 سريع ، أو كأنه معالج 386 في النمط الحقيقي. وتكون بهذا النمط مجموعة مسجلات المعالج 386 متوافقة مع مسجلات النمط الحقيقي للمعالجين 8086 ، 80286 . يشار إلى مسجلات الأغراض العامة ذات الطول 32 bit بالمسجلات الموسعة EAX , EBX, ECX, EDX (يضاف لأسماء المسجل 0868 الحرف E الذي يعني توسع Extended). وهكذا تستطيع التعليمات الرجوع إلى المسجل AL أو AH أو AX أو EAX حيث إن المجمع يشفر التعليمة آليا' من أجل حجم المسجل المشار إليه في التعليمة. يحسب المعالج 386 في النمط الحقيقي العناوين الذاكرية باستخدام نفس قاعدة القطعة ونفس أسلوب الإزاحة Offset المستخدم من قبل المعالج 8086. يلاحظ أيضاً أن المعالج 386 يملك مسجلين إضافيين لقطعة المعطيات هما : FS و GS، وبالتالي يمكن للبرامج أن تحتوي حتى 4 قطع للمعطيات ، ويكون طول القطع في النمط الحقيقي للمعالج 386 ثابتا عند المقدار 64 KB. يكون مجال العناوين في النمط الحقيقي للمعالج 386 محدودا بـ 1MB وبالتالي فإن خطوط العناوين من A_{31} – A_{20} تكون كلها منخفضة بشكل طبيعي .

من إحدى المميزات الجديدة للمعالج 386 هي مسجلات التنقيح Debug Registers حيث يستطيع المنقح تحميل عناوين نقاط التوقف Break Points في هذه المسجلات بهدف المساعدة في عملية التنقيح .

يعدّ المسجل E Flags بطول 32 bit في المعالج 386 امتدادا أو توسعا لنظيره المؤلف من 16 bit في المعالج 80286 حيث البيتات التي لها معنى في النمط الحقيقي هي البيتات الستة عشر السفلى فقط.

١٣-١-٢ النمط المحمي للمعالج 386

تكمن القوة الحقيقية للمعالج 386 في نمطه المحمي وفي مميزات النمط 8086 الافتراضي. وقد صممت هذه المميزات بطريقة متعددة الجوانب ، بحيث إن أي نظام تشغيل تقريبا أو أي برنامج يمكن تحقيقه على المعالج 386، يندرج تحت هذا العنوان العديد من المفاهيم التي لا يتسع المجال هنا لتوصيفها ومنها:

- كيف يحسب المعالج العناوين الفيزيائية في نمط القطع فقط وفي النمط الصفحي
- كيف يؤمن المعالج الحماية من أجل شفرة نظام التشغيل ومن أجل مهام المستخدمين
- العملية الأساسية للبوابة Gate
- استجابة مقاطعة النمط المحمي
- معالجة انتقال المهمة (أي الانتقال من مهمة إلى أخرى)
- وأخيرا" نموذج النظام الناتم Flat System Model

ومن السجلات الواجب ملاحظتها بالنمط المحمي، مسجلات التحكم CR0 ÷ CR3 حيث إن البيتات الستة عشر السفلى للمسجل CR0 تقابل كلمة أحوال الآلة MSW للمعالج 286، وكما يحصل في المعالج 286 فإن المعالج 386 ينتقل إلى العمل في النمط المحمي بوضع 1 في البت LSB لهذا المسجل .

أما المسجل CR1 فهو محجوز من قبل شركة Intel بينما المسجلان CR2 و CR3 يستخدمان من أجل وظائف النمط الصفحي Paged Mode.

وبخصوص التحسينات والإضافات الحاصلة على مجموعة تعليمات المعالج 386 يمكن توسيع هذه التعليمات للعمل مع كلمات معطيات بطول 32 bits. ويتضمن المعالج 386 عدة تعليمات جديدة يمكن تصنيفها كالتالي :

- تعليمات مسح واختيار البتات Bit scan and test instruction

وتضم التعليمات: BSF – BSR – BT – BTC – BTR – BTS

- تحويلات من نوع معطيات Data – type conversions

مثل: CDQ و CWDE

- تعليمات تحميل القطع Segment load instructions

مثل: LFS – LGS – LSS

- تعليمات النقل والتوسيع Move and expand instructions

مثل: MOV ZX , MOV SX

- تعليمات رفع الكلمة من المؤشرات الذاكرية

Set memory flag word instruction وشكلها SET xx

- الإزاحات ما بين الكلمات Shift between words

ومنها SHLD – SHRD.

ويوجد أيضاً بعض التحسينات المضافة على التعليمات.

٢-١٣ المعالج الصغري 80486

من السمات التي يتميز بها المعالج 80486 (اختصارا 486) الذي يمثل الجيل الرابع لمعالجات Intel هو أنه يحتوي داخلياً على ذاكرة Cache بسعة 8 KB ويحتوي أيضاً على وحدة نقطة عائمة FPU (معالج حسابي) داخل الرقاقة. وقد وصلت سرعة المعالج 486 بالتحسينات المضافة إلى مرة ونصف المرة من سرعة المجموعة 386 و 387 عند نفس الـ Clock Speed .

يستخدم المعالج 486 قاعدة (أو حافظة) Socket جديدة يطلق عليها Socket1 وهي من النوع PGA تحتوي على 168 Pin . يوجد للمعالج 486 عدة إصدارات، كان أولها المعالج 486 SX وهو مماثل للإصدار 486 DX لكن بوحدة FPU غير ممكنة (Disabled)، وكان أطلق على آخر إصدار للمعالج 486 الاسم 486DX4 – 100 وهو يعمل عند Core Speed مقدارها 100 MHz وكان أول معالج يعمل عند هذه

السرعة ويملك Level-1 Cache مضاعفة، أي بحجم 16KB.

تحتاج رقاقة المعالج 486DX4 إلى Socket خاص ليوفر تغذية منخفضة الجهد أي 3.3 فولت بدلا من 5 فولت لتفادي ارتفاع حرارة المعالج نتيجة السرعة العالية. لذلك قامت شركة Intel بتعديل هذا المعالج وأطلقت على الإصدار المعدل اسم DX4 Over Drive، حيث تحتوي الرقاقة نفسها على منظم جهد خاص لتحويل الجهد من 5V إلى 3.3V بالإضافة إلى تزويد الرقاقة بمصرف حراري (Special Heat Sink) خاص للتخلص من الحرارة الزائدة.

يعّد المعالج 486 نسخة مطورة عن المعالج 386 حيث يمتلك نفس وحدة المعالجة ALU بعرض 32 Bits وله نفس قدرات العنونة وإدارة الذاكرة وميزات الحماية، وبالتالي كافة المناقشات حول المعالج 386 تنطبق على المعالج 486.

سنكتفي هنا بعرض بعض التحسينات التي استطاع المصممون إضافتها من خلال زيادة عدد الترانزيستورات على الرقاقة من 300000 لدى المعالج 80386 إلى حوالي 1200000 ترانزيستور لدى المعالج 80486 .

حيث إن من أهم الميزات الأساسية الجديدة المضمنة في المعالج 486 وجود المعالج الحسابي المساعد مع الـ CPU على نفس الشريحة، علما أن هذا المعالج المساعد يعدّ أساساً هو نفس المعالج 80387 المستخدم مع المعالج 386، لكن عملية تكامله على نفس الشريحة أدت إلى زيادة الإمكانية لتنفيذ التعليمات الحسابية بسرعة تصل إلى حوالي ثلاثة أضعاف السرعة التي تحققها التركيبة 386/387.

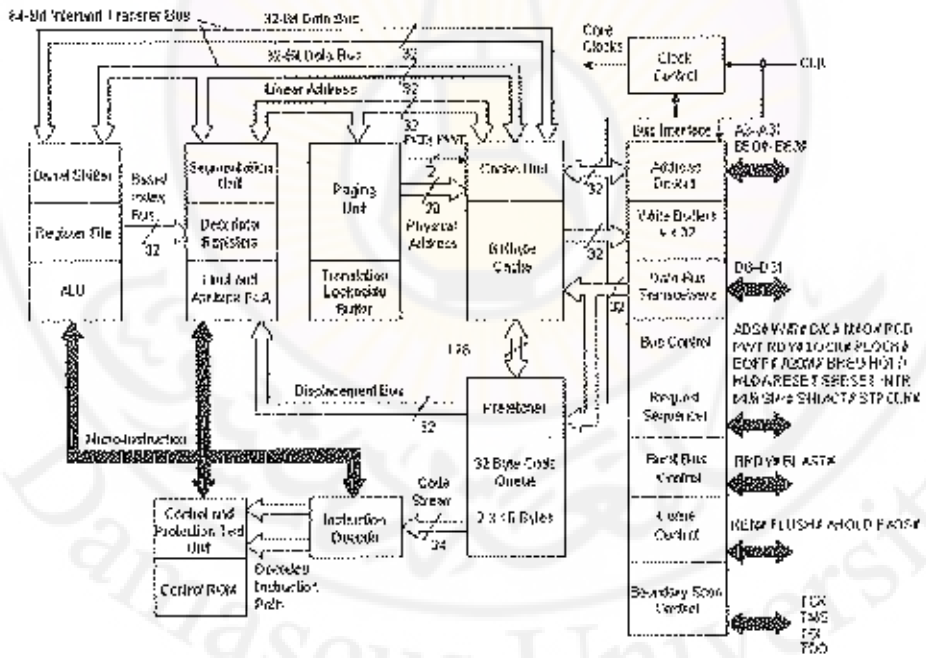
ويوجد أيضاً في المعالج 80486 تحسين آخر أقل وضوحاً هو نظام "الخط الأنبوبي للتعليمات" أو "خط تنفيذ التعليمات Instruction Pipeline" والذي يتكون من خمس مراحل Five stages بحيث يسمح للمعالج 486 بتنفيذ التعليمات بسرعة أكبر بكثير من سرعة تنفيذ المعالج 386 لها. يستخدم هذا النظام عموما في معالجات RISC ويسمح بوجود عدة تعليمات في الخط الأنبوبي دفعة واحدة . بمعنى أن المعالج 486 سيجلب عدة تعليمات مسبقا، وبينما يكون في حالة تنفيذ تعليمة واحدة فإن المعالج سيفكك شفرة تعليمة أخرى بأسرع ما يمكن ويبدأ بتنفيذ التعليمة التالية.

وهكذا فقد ينفذ المعالج 486 فعليا أجزاء لعدة تعليمات في نفس الوقت . فمثلا" عمليات الجلب والتفكيك والتنفيذ لكل تعليمة قد يأخذ عدة دورات ساعة.

فعملية الكتابة الذاكرية لقيمة بطول 16 Bit تأخذ 22 دورة ساعة لتنفيذها على المعالج 8088 وتأخذ 4 دورات ساعة لتنفيذها على المعالج 386، بينما تأخذ دورة واحدة فقط لتنفيذها على المعالج 486 .

إن معظم التحسينات الأخرى المشمولة في المعالج 486 تتضمن إشارات المكونات الصلبة Hardware signals والربط البيني لها Interfacing .

ولاحتواء كل هذه الإشارات الإضافية فقد وضعت شريحة المعالج 486 في صندوق سيراميكي (أو بلاستيكي) بـ 168 pins على شكل مصفوفة شبكية PGA من المآخذ بدلا من الشكل المستخدم لتوضع 132 مأخذاً للمعالج 386 . يبين الشكل 3-13 مآخذ المعالج والإشارات الموزعة عليها ضمن الصندوق المقابل له.



الشكل3-13: توزع مآخذ المعالج 80486

تعمل الإشارات BS16', HLDA, HOLD, LOCK', W/R', D/C', M/IO', NMI, RESET, INTR, RDY', ADS' وإشارات تمكين البايت، وممر العناوين، وممر المعطيات في المعالج 486 كما تعمل في المعالج 386.

أما مجموعة الإشارات الجديدة للمعالج 486 تتمثل بمجموعة التكافؤ DP0 -- DP3 (Parity group) وكذلك الإشارة 'PCHK التي تسمح للمعالج باكتشاف / توليد التكافؤ من أجل القراءات من الذاكرة والكتابات فيها. فأثناء عملية كتابة ذاكريه يولد المعالج 486 بت التكافؤ الزوجي لكل بايت ويخرج هذه البتات على الخطوط DP3 -- DP0 وتخزن في بنك ذاكري منفصل التكافؤ. وأثناء عملية القراءة فإن بتات التكافؤ المخزنة هذه ستقرأ من ذاكرة التكافؤ وتطبق على المأخذ DP3 – DP0 . وبعد ذلك يختبر المعالج 486 التكافؤ ات المتعلقة ببايتات المعطيات المقروءة ويقارنها مع الإشارات $DP_3 – DP_0$. فإذا وجد خطأ تكافؤ فإنه سيمكّن الإشارة 'PCHK لتكون فعّالة .

يوجد مجموعة أخرى من الإشارات الجديدة لدى المعالج 486 تتألف من إشارة جاهزة التدفيق 'BRDY (Burst ready signal) ومن إشارة آخر تدفق 'BLAST وتستخدم للتحكم بقراءات وكتابات الذاكرة في النمط التدفقي Burst mode.

ومن الإشارات الجديدة أيضاً، إشارة طلب الممر BREQ (Bus Request) والإشارة 'BOFF حيث تستخدم هذه الإشارات للتحكم بالمشاركة على الممر المحلي للمعالج 486 من قبل معالجات متعددة تمثل أسياد الممر.

عندما يحتاج سيد ما Master موجود على الممر إلى استخدام هذا الممر، فإنه يمكّن إشارته BREQ بجعلها مرتفعة وستقوم دارة خارجية للتعامل مع الأولويات بتقييم الطلبات على استخدام الممر ثم تمنح grant استخدام الممر إلى السيد ذي الأولوية. ولكي يتم سؤال المعالج 486 عن تحرير الممر فإن محكم الممر يمكّن المدخل HOLD أو المدخل 'BOFF في المعالج 486.

وبخصوص الجانب البرمجي للمعالج 486 تم اعتماد ست تعليمات إضافية زيادة على تعليمات المعالج 386.

٣-١٣ المعالج الصغري 80586

توقع الناس بعد الجيل الرابع لمعالجات العائلة Intel 8086 Family، أن يكون اسم معالج الجيل الخامس هو 80586 ، لكن شركة Intel أرادت أن تستخدم اسما تجارياً بدلا من استخدام العدد، فكان الاسم هو Pentium كلاحق للمعالج I80486. يستخدم معالج بنتيوم المعمارية المصغرة Micro architecture من الجيل الخامس، أي P5 وقد أعطت شركة إنتل Code لكل جيل من أجيال الـ Pentium . استخدمت معالجات البنتيوم منذ بدايات ظهورها المعمارية CISC وضمت بداخلها عدداً كبيراً من الترانزستورات، مما سمح لها بتكامل عدة وحدات على نفس الشريحة مثل وحدة الفاصلة العائمة ووحدة لإدارة الذاكرة وذاكرة كاش وغيرها. وبشكل عام يمتاز معالج بنتيوم (المعالج I80586) بالخصائص التالية: – تاريخ ظهوره: 1993

– مسار (Process) التصنيع: 0.8 ميكرون

– عدد الترانزستورات الداخلة في تركيبته: 3.1 مليون

– السرعة (الساعة): 60 and 66 MHz

– حجوم ذاكرة الكاش (Cache): L1: 8 K Instruction 8k data

تضم المعالجات الحديثة ابتداء من المعالج 80586 (البنتيوم) أكثر من معالج على نفس الشريحة وهذا ما تم التعبير عنه باستخدام مصطلح الـ Core الذي يعني الصميم أو النواة (القلب)، حيث كما نعلم أن كل معالج صغري يملك وحدة معالجة مركزية CPU واحدة على الأقل تسمى بالـ Core، لكن في هذه الأيام أصبحت معظم المعالجات مجهزة بأكثر من وحدة للمعالجة المركزية. إن مصطلح مثل Dual-core chip يستخدم للإشارة إلى معالج بشريحة CPU واحدة، لكنه في الواقع يملك وحدتي CPU على نفس الشريحة وهذا يعني معالج بسرعة وكلفة أعلى. وبالتالي تعبير Dual core يعطي عدد الـ Core لدى CPU، ومن هذه التعابير قد نصادف معالجات

Dual Core: Two cores,

Quad Core: Four cores, Hexa Core: Six cores,

Octo Core: Eight cores. Deca Core: Ten cores.

الفصل الرابع عشر: معالجات Pentium

ظهر معالج البنتيوم Pentium منذ ما يقارب العشرين سنة من تاريخه وقد ظهر له بهذه الفترة من التطور السريع على صعيد تقنية المعالجات الصغرية، عدة إصدارات، بالتالي يلزم أكثر من كتاب للتعريف بشكل كامل بهذا المعالج، لكن سنكتفي هنا بإفراد هذا الفصل للتعريف بالأساسيات المتعلقة بهذا الموضوع، خاصة وأن معالجات البنتيوم أصبحت تمثل أوسع استخدام للمعالجات في الحواسيب على مختلف أنواعها. تمثل كلمة "بنتيوم Pentium" العلامة التجارية المستخدمة لسلسلة المعالجات المتوافقة مع X86 التي أنتجتها إنتل. ويشتق اسم بنتيوم أصلا من الكلمة اليونانية Pente التي تعني "خمسة" واللاحقة اللاتينية ium–.

١٤-١ تطور معالجات Pentium

تميزت معالجات بنتيوم بإضافة وحدات معالجة إضافية على نفس الشريحة تسمح للحاسوب بفعل الكثير من الأشياء بوقت واحد ، فمثلا: إذا أردت أن تنتزع ملفاً من الأرشيف وتتصفح الويب في نفس الوقت وكان لديك حاسوب بوحدة معالجة CPU واحدة (Single core) فإن تصفح الويب لن يكون متجاوباً بشكل تام، لان الصميم الواحد سيقسم الزمن بين التصفح وانتزاع الملف، أما إذا كان الحاسوب يملك dual core (وحدتي معالجة)، فإن صميم واحد يعمل لانتزاع الملف في الوقت الذي يقوم به الصميم الآخر بتصفح الويب. أي أن مضاعفة عدد الـ Core تزيد السرعة، لكنها لا تضاعف الأداء.

كانت المعالجات الأولى من Pentium تعمل بنفس سرعة اللوحة الأم عند جهد 5 V وتستهلك قدرة عالية، وتنشأ عنها حرارة عالية. تم تحديث معالجات بنتيوم وتطويرها لتتمكن من استخدام جهد منخفض واستهلاك أقل للقدرة وزيادة في سرعة الأداء. صمم المعالج Pentium Pro للحواسيب الشخصية التي تعمل كمخدمات Servers ومحطات عمل Workstations لتخديم عدة مستخدمين.

يحتوي المعالج Pentium Pro على ما يقارب 5.5 مليون ترانزيستور، ويتضمن ميكرو شيب يحتوي على ذاكرة كاش التي تزيد من سرعته.

انتهى عهد Pentium Pro بتحسينات جديدة طرأت على تكنولوجيا الـ Pentium أطلق عليها الاسم MMX (Multimedia Extension) التي تمتاز بوجود أوامر للتعامل مع الوسائط المتعددة.

وبما أن معالج الـ Pentium Pro انطلق قبل المعالج Pentium MMX لم يتمتع بالتحسينات التي زود بها المعالج MMX ، لكنه لم يكن في حاجة إلى هذه التحسينات لان سوق الـ MMX كان يتركز في الوسائط المتعددة، بينما سوق الـ Pentium Pro كان يتركز في أجهزة الحاسوب الخادمة، وبالتالي فكل له هدف مختلف عن الآخر. بسبب زيادة حجم الذاكرة L2 Cache وارتفاع سرعتها أدى ذلك إلى وضع المعالج أمام بعض المشاكل، لذلك أتى المعالج Pentium II الذي ظهر في عام 1997 بذاكرة L2 Cache تعمل عند سرعة تعادل نصف سرعة الصميم Core Speed.

يعّد المعالج Pentium II هو نفس المعالج Pentium Pro بتقنية MMX التي تجعله يعمل بساعة 233MHz , 266 MHz, and 300 MHz ومناسبا لتطبيقات الفيديو والصور ثلاثية الأبعاد. من بين ميزات المعالج Pentium II وجود 512 KB من ذاكرة L2 – Cache التي تحوي شيفرة تصحيح الخطأ ECC.

الإصدارات الأخيرة لـ Pentium II كانت Celeron (low-end Pentium) بدون الكاش L2 و Xeon (high-end Pentium) الذي حل محل Pentium Pro بمشروع الحواسيب الصغرية كمخدمات ومحطات عمل. ونتيجة كبر حجم المعالج Pentium II (لاحتوائه على L2 Cache) لم يستخدم Socket وإنما استخدم Slot1.

جاء المعالج المميز والمعروف بالاسم Celeron ليخدم الزبائن متوسطي الدخل (لأنه رخيص الثمن) فهو يعد Pentium II ولكن بدون L 2 Cache مما أدى إلى تخفيض مستوى الأداء.

وبالرغم من أن معظم معالجات الـ Pentium السريعة تستخدم الجهد 3.3 V لصميم المعالج CPU Core إلا أن جهد الصميم Core Voltage في بعض الحالات يتطلب أن يتغير عن ذلك للحصول على أقصى اعتمادية Reliability عند السرعات العالية. يطلق على متطلبات التغيير الطفيف في جهد الصميم المصطلح VR (Voltage

(Reduced Range للسماح بتغيير الجهد في المدى من 3.3 V إلى 3.45 ، بينما المصطلح VRE (Voltage Reduced Range Extended يستخدم للسماح بتغيير الجهد في المدى من 3.45 V إلى 3.6 V .

يستخدم المصطلح STD (Standard) للسماح بتغيير الجهد في المدى من 3.15 V إلى 3.45 V. تتطلب بعض المعالجات مثل Pentium MMX بأن يكون جهد الصميم للمعالج صغيراً جداً وهو 2.8 V مما أدى إلى مشاكل كثيرة عند تطوير نظام Pentium. وبعد ذلك جاء معالج Pentium 2 Xeon بسرعة صميم كاملة وفيه Heat Sink كبير جداً نظراً للحرارة العالية.

ومع إطلاق شركة Intel المعالج Pentium III عام 1999 تمتع الإصدار الأول منه بسمة جديدة عن Pentium II هي SEEs (Streaming SIMD Extensions) أو المصطلح MMX2 . وقد تمتعت معالجات Pentium III التي جاءت بعد ذلك بتكنولوجيا الـ Copper Mine (وهي تقايص حجم الدوائر بالمعالج بنسبة % 30). قدمت تكنولوجيا الـ Copper Mine كثيراً من المميزات التي تمكّن كل معالج من أن يقفز قفزة للأمام سواء على صعيد سرعة الصميم Core Speed أو سرعة الـ Cache وهذا ليس فقط لـ Pentium III إنما أيضاً لمعالجات الـ Celeron و Xeon . فمثلاً بالنسبة لمعالجات الـ Celeron إذا أضيف حرف a لاسم المعالج مثل Celeron 533a فإن ذلك يشير إلى أن المعالج يتمتع بتكنولوجيا Copper Mine.

وللتخلص من الحرارة فقد تم تجميع جميع رقاقات Copper Mine في شكل يسمى FCPGA (Flip – Pin Grid Array Chip) وهذا يعني أن الجانب السفلي لقالب السليكون الفعلي يتعرض للجو الخارجي بدلا من انحصاره داخل غلاف. وبالنسبة لمعالجات الـ Pentium III التي تستخدم Slot 1 ومعالجات الـ Pentium III Xeon التي تستخدم Slot 2 لابد من تثبيت مصرف حراري بها قبل تركيبها في الـ PC. ويمكننا تلخيص أهم إصدارات المعالج Pentium على النحو التالي:

١- تم التعريف بالمعالج 80586 بأنه Pentium (الأول أو الأساسي Original) بالعام 1993 (60--200 MHz) Pentium – 1993--1996 على أنه:

- مماثل للمعالج 486 ولكن يمرر معطيات بعرض bit 64.
- أضاف خط أنابيب ثاني للتنفيذ بأداء سوبرسكلار Superscalar وبتعليمتين بالحلقة الساعية الواحدة (علما أن مصطلح Superscalar يوصف تصميم المعالج الصغري الذي يسمح بتنفيذ أكثر من تعليمة بحلقة ساعية واحدة. وتدعى معمارية Superscalar CPU بمستوى التعليمات المتوازية، كما يدعى Superscalar design بالجيل الثاني لـ RISC)
- ذاكرة L1 Cache مضاعفة بمعدل 8 KB للتعليمات و 8 KB للمعطيات
- الشرائح كانت مغلفة (محفوظة) بتغليف من نوع PGA

٢- وتم التعريف بالمعالج Pentium Pro بالعام 1995

Pentium Pro – 1995–1997 (150–200 MHz) على أنه:

- معالج بـ 36 خط للعنونة، تسمح بعنونة مجال ذاكري بحجم 64 GB
- أدخل ديناميكية التنفيذ وتنفيذ المضاربة speculative execution
- إضافة لذاكرة L1 Cache يملك L2 Cache بسعة 256 KB.
- Three-way superscalar حيث يقوم بتنفيذ ثلاث تعليمات بحلقة ساعة واحدة.
- housed in a dual cavity PGA package

٣- ظهر المعالج Pentium MMX بالعام 1997 أي

Pentium MMX – 1997–1999 (233–300 MHz) وهو يمتاز بأنه أضاف تعليمات الوسائط المتعددة multimedia لوحدة CPU للبنتيوم. كما أن عدد الترانزستورات المشكلة للمعالج زاد إلى 4.5 مليون ترانزيستور. تستخدم وحدة المكتب Desktop units التغليف PGA، بينما وحدة الموبايل تستخدم TCP.

٤- تم التعريف بالمعالج Pentium II بالعام 1997

Pentium II – 1997–1999 (233–450 MHz) الذي يتميز في أنه:

- أضاف تعليمات الوسائط المتعددة MMX لـ Pentium Pro
- عرض خرطوشة موصل بحافة واحدة Single Edge Connector Cartridge اختصارا SECC للشق Slot1.
- يستخدم ممراً للنظام بسرعة 66 or 100 MHz

– النموذج المكتبي يحتوي على 7.5 مليون ترانزيستور ، بينما نموذج الموبايل يحتوي على 27.4 مليون ترانزيستور

– ذاكرة L1 Cache مضاعفة بمعدل 16 KB للتعليمات و 16 KB للمعطيات

– بالإضافة لـ L1 Cache يملك 256 KB من ذاكرة L2 Cache.

– عرض ميزات شاملة لإدارة الطاقة من نوم Sleep والنوم العميق Deep Sleep

– were housed in SECC packages

وهكذا تتالت إصدارات المعالج Pentium

٥– المعالج Pentium II Xeon – 1998–1999 (400–450 MHz)

Typically used in high-end and 2-way and 4-way servers, Xeon specs were like Pentium II with L2 cache from 512KB to 2MB and 100 MHz system bus.

٦– المعالج Pentium III – 1999–2001 (500 MHz–1.13 GHz)

The Pentium III added 70 additional instructions to the Pentium II. The Pentium III used a 100 or 133 MHz system bus and either a 512KB L2 cache or a 256KB L2 Advanced Transfer Cache. Depending on the model, it contained from 9.5 to 28 million transistors, and came in SECC packages.

٧– المعالج Celeron – Introduced in 1998 (266 MHz–2.8 GHz)

First Celerons had no L2 cache, but 128KB on-die cache was added in 1999. Celerons started out with 66 and 100 MHz system buses that migrated to 400 MHz.

٨– المعالج Pentium III Xeon – 1999–2001 (500–933 MHz)

Typically used in 2-way to 8-way servers, Xeon specs were like Pentium III with L2 cache up to 2MB. The Xeon used the SECC2 and SC330 chip packages.

٩- المعالج Pentium 4 – Introduced in 2000 (1.4–3.4 GHz)

Latest Pentium architecture started out with a 400 MHz system bus and 256KB L2 cache (later increased to 800 MHz and 2MB). The first models contained 42 million transistors, and came in 423-pin and 478-pin PGA packages.

١٠- المعالج Pentium 4 Dual Cores – Introduced in 2005

The Pentium D and Pentium Processor Extreme Edition were the first dual-core Pentium chips from Intel. Although both chips included Intel's 64-bit EM64T technology .

* معالج أيتانيوم Itanium Processor ويمتاز بـ:

- RISC Design حيث إن التصميم الأسبق كان CISC ، علما أن RISC هي معالجات صممت لأداء عدد أقل من أنواع التعليمات، بحيث يمكن أن تعمل بسرعة أعلى، بينما CISC تعرض مجال كبير وقوي لتعليمات يمكن إنجازها بمرونة أقل.
- معالج بـ 64 bit (يستخدم عناوين بعرض 64 bit ومعطيات بعرض 128 bit).
- يعرض عدة ميزات متقدمة منها: تنفيذ المضاربة speculative execution، والتنبؤ للقضاء على الفروع prediction to eliminate branches تم إلغاء الاسم Pentium بعد 13 سنة من استخدامه وذلك بحلول العام 2006 عندما حل محله خط النواة من إنتل Intel Core. أسند الاسم كور Core لثلاث معماريات (أبنية) مختلفة هي:

- Core Duo التي ظهرت مع مطلع العام 2006 استناداً إلى معمارية المعالج Pentium M (للحاسوب المحمول) ثنائي النواة dual core. وكانت أول شريحة 32 bit core duo تم التعرف عليها مع النواة الرئيسية Core brand.
- Core 2 وتم التعرف عليها بمنتصف عام 2006 وكانت المغادر الرئيسي من شرائح core duo السابقة. وباستخدام المعمارية المصغرة Penryn أصبحت شريحة core 2 متاحة بنموذج أحادي، ثنائي ورباعي كور.

– المعمارية الأحدث Core i Series (i3, i5, i7)، حيث إن المعمارية المصغرة Micro architecture لكور من سلسلة i رمزت بالاسم نيهالم Nehalem ، ورمز الجيل الثاني من الخط بالاسم ساندي بريدج Sandy Bridge. تم في العام 2009 إسقاط لاحقة " ثنائي النواة " ، ومعالجات X86 الجديدة انطلقت باسم بنتيوم عادي مرة أخرى.

٢-١٤ المعماريات المصغرة المستخدمة بمعالجات Pentium

يمكن استعراض المعماريات المصغرة المستخدمة بالمعالجات الموسومة ببينتيوم كالتالي:

1- P5 micro architecture based

كان بنتيوم الأصلي Original ومعالجات بنتيوم MMX سوبرسكيلر superscalar متابعة للمعالج 80486 وتم تسويقها 1999- 1993 وكانت بعض الإصدارات منها متاحة كا Pentium Overdrive التي كانت تلائم CPU sockets القديمة.

2- P6 micro architecture based

بالتوازي مع المعمارية المصغرة P5 ، طورت شركة إنتل المعمارية المصغرة P6 التي أخذت بالعام 1995 تسويقها باعتبارها معالج Pentium Pro وعرفت

بـ out-of-order execution وبتكامل second level cache بغلاف DIL.

الجيل الثاني لـ P6 استبدل P5 الأصلي بالاسم Pentium II بتغيير العلامة التجارية إلى النسخة الراقية Pentium II Xeon.

تبع ذلك إصدار ثالث يسمى Pentium III and Pentium III Xeon على التوالي . وكانت تسمى الإصدارات من هذه المعالجات لسوق أجهزة الحاسوب المحمول في البداية Mobile Pentium II and Mobile Pentium III وكانت تسمى الإصدارات الأحدث بـ Pentium III-M . تم بدءا من Pentium II استخدام العلامة التجارية سيليرون لمعظم إصدارات بنتيوم المنخفضة التكلفة.

3- Netburst micro architecture based

عرفت إنتل في عام 2000 معمارية مصغرة بالاسم NetBurst مع خط أنابيب أطول بكثير تسمح بترددات ساعية أعلى لما في المعالجات القائمة على P6. بداية كان يسمى

Pentium 4 والنسخة the high-end كانت تدعى Xeon وكما هو الحال مع Pentium III كان يوجد هناك معالجات لسوق الحاسوب المحمول بالاسم Mobile Pentium 4 and Pentium 4 M ، مع بنتيوم Pentium 4 M الذي يرمز لإصدار أكثر كفاءة في استخدام الطاقة. وقد سميت نسخ المتحمسين من Pentium 4 مع تردد ساعي أعلى بـ Pentium 4 Extreme Edition. المعالج Pentium D كان أول بنتيوم متعدد النوى يكامل شريحتين من Pentium 4 بنفس الحزمة أو التغليف

4- Pentium M micro architecture based

في عام 2003 ، قدمت شركة إنتل المعالج الجديد بالاسم Pentium M الذي كان أكثر كفاءة من Mobile Pentium 4, Pentium 4 M and Pentium III M استنادا إلى المعمارية المصغرة P6. الإصدار ثنائي النواة من Pentium M طور بالاسم المشفر Yonah وتم بيعه وتسويقه تحت الاسم Pentium Dual-Core و Core Duo وعلى عكس Pentium D تمت مكاملة النواتين على شريحة واحدة .

5 - Core micro architecture based

استمر الاسم Pentium Dual-Core ليستخدم عندما التصميم Yonah توسع ليدعم 64 bit وحاليا يدعى Core micro architecture . تستبدل هذه المعمارية المصغرة كافة المعالجات القائمة على NetBurst عبر العلامات التجارية الأربعة:

معالجات Celeron, Pentium, Core and Xeon المبنية على المعمارية المصغرة. تستخدم تصاميم Allendale and Wolfdale-3M لمعالجات سطح المكتب و Merom-2M لمعالجات الموبايل.

وفي عام 2009 غيرت إنتل نظام التسمية لمعالجات بنتيوم وإعادة تسمية المعالجات القائمة على Wolfdale-3M ببنتيوم بدون اسم ثنائي النواة وقدمت معالجات أحادية وثنائية النواة جديدة تقوم على Penryn باسم بنتيوم .

حققت إنتل أول Penryn Core أي Pentium T4200 عام 2008. في عام 2009 أصدرت شركة إنتل أول معالج أحادي النواة الذي استخدم اسم Consumer Ultra-Low Voltage (CULV) (المستهلك فائقة الجهد المنخفض).

6- Nehalem micro architecture based

قدمت المعمارية المصغرية Nehalem في أواخر عام 2008 خلفا للمعمارية المصغرة كور. وقد ادخل بعام 2010 معالج بنتيوم Pentium G6950 على أساس التصميم Clarkdale استنادا إلى التحديث لـ Nehalem . أطلقت إنتل في عام 2010 نموذجاً جديداً لبنتيوم باستخدام رقاقة Clarkdale بالتوازي مع وحدات معالجة CPU لسطح المكتب والمحمول على أساس المعمارية المصغرة Westmere الجديدة .

النموذج الأول في هذه السلسلة هو بنتيوم G6950 . يتم استخدام رقاقة Clarkdale أيضاً في i3 - 5XX كور وسلسلة كور i5-6xx

7- Sandy Bridge micro architecture based

تم الإفراج عن المعمارية المصغرية Sandy Bridge بخط بنتيوم عام 2011.

8- Ivy Bridge micro architecture based

توجد حاليا نماذج G2010، G2020، G2120، G2130 وكلها ثنائية النواة.

9- Haswell micro architecture

يمثل Haswell الاسم الرمزي codename لمعمارية المعالج التي وضعتها إنتل خلفا للمعمارية Ivy Bridge، وهي تستخدم مسار 22. nm أعلنت إنتل عن هذه المعمارية المصغرة في العام 2013 وبهذا الشكل قدمت إنتل مع Haswell معالج منخفض الطاقة صمم لـ Ultrabooks القابل للتحويل (الهجين) الذي ينتهي باللاحقة Y. وقد تم تصميم المعمارية Haswell تحديدا لتحسين توفير الطاقة وأطلقت بثلاث أشكال رئيسية:

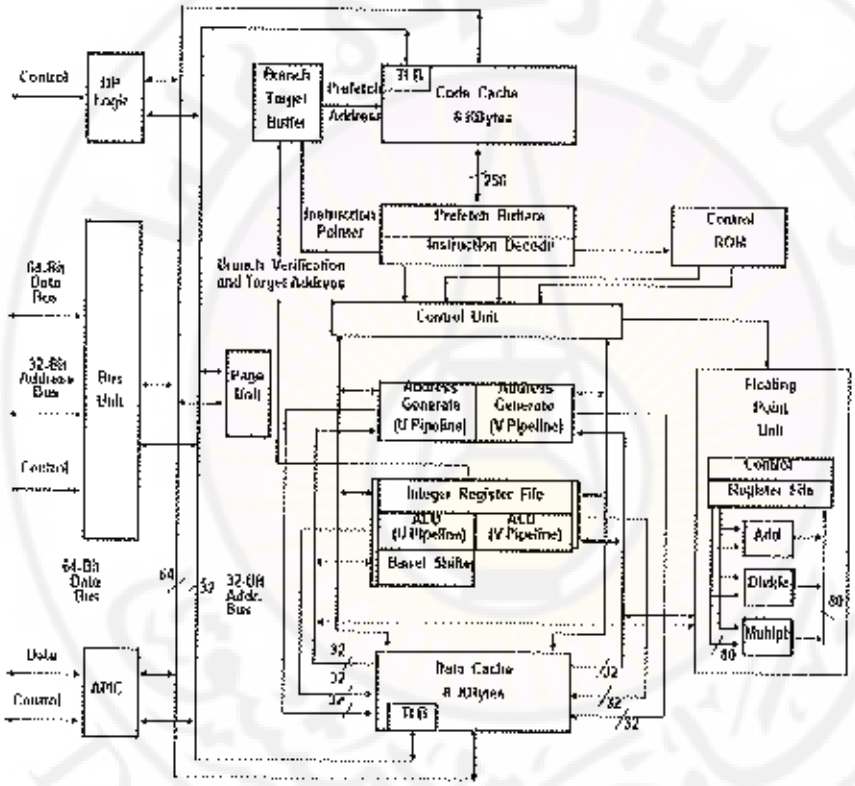
- Desktop version: Haswell-DT
- Mobile/Laptop version: Haswell-MB
- BGA version: Haswell-H, Haswell-ULT, Haswell-ULX

٣-١٤ التركيبة الداخلية لمعالج Pentium الأساسي

للتعريف عن قرب بالمكونات الأساسية لإحدى معالجات Pentium من حيث البنية والإشارات، نعرف المعالج Pentium 510/60, 567/66 حيث إن الشكل 1-14 يبين

المخطط الصندوقي لهذا المعالج وهو مماثل لكل المعالجات التي تعمل بالتوترات:
75/90/100/120/133/150/166/200 MHz.

يوجد تباين بالتركيبة الداخلية لكل من معالجات بنتيوم، حيث يصعب هنا عرض هذه التباينات بالتركيبة الداخلية لكل من هذه المعالجات. نكتفي بالإشارة إلى التركيبة الداخلية الأساسية المبينة بالشكل أدناه التي تضم الوحدات الأساسية بكل من معالجات البنتيوم دون الدخول بالتفاصيل، لان مفهوم معظم هذه الوحدات أصبح واضحا.



الشكل 1-14: المخطط الصندوقي لمعالجات Pentium بالتوترات
75/90/100/120/133/150/166/200 MHz

أكثر مايهمنا من التركيبة الداخلية للمعالج كمستخدمين هو مسجلات المعالج، حيث إن للمعالج Pentium مجموعة من المسجلات المماثلة للمسجلات بالمعالجات الأقدم والتي يمكن توزيعها كالتالي:

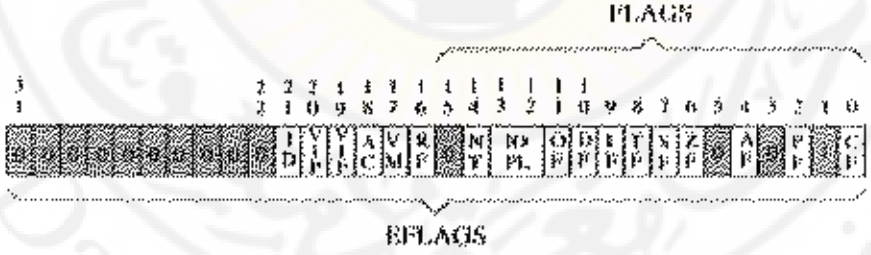
١- مجموعة المسجلات العامة: وهي مبينة بالشكل 2-14 وتضم:

- أربعة مسجلات كل منها بـ 32 bit هي: EAX, EBX, ECX, EDX
- أربعة مسجلات كل منها بـ 16 bit هي: AX, BX, CX, DX
- ثماني مسجلات كل منها بـ 8 bit هي: AH, AL, BH, BL, CH, CL, DH, DL

الشكل2-14: المسجلات العامة في معالج البنتيوم

٢- مسجلات التحكم Control Registers وتضم مسجلين هما:

- (E)IP عداد البرنامج Program Counter وهو مسجل بعرض 32 bit ويمكن التعامل مع قسمه السفلي فقط بعرض 16 bit.
- (E) Flags ويحتوي على 17 مؤشر وفق ما هو مبين بالشكل 3-14.



الشكل3-14: مسجل المؤشرات في معالجات بنتيوم

وتقسم هذه المؤشرات إلى ثلاث مجموعات هي:

الأولى: مؤشرات الحالة Status Flags التي تسجل معلومات عن حالة النتيجة بعد آخر تعليمية حسابية/منطقية. وعدد هذه المؤشرات هو ستة مؤشرات المعروفة:

CF, PF, AF, ZF, SF, OF والموجودة بكل معالجات إنتل.

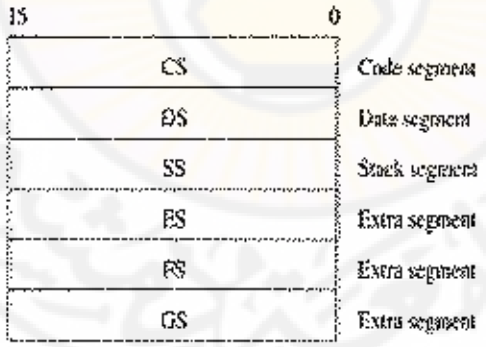
الثانية: مؤشر الاتجاه Direction Flag وتضم هذه المجموعة مؤشراً واحداً معروف هو DF للاتجاه Forward/Backward لنسخ البيانات

الثالثة: مؤشرات النظام System Flags وتضم عشرة مؤشرات، يظهر الجدول 1-14 اسم كل منها.

الجدول1-14: أسماء المؤشرات الإضافية في معالجات بنتيوم عن مؤشرات المعالج 8086

NT – Nested Task	AC – Alignment Check
RF – Resume Flag	IOPL – I/O Privilege Level
ID – ID Flag	VIF – Virtual Interrupt Flag
VM – Virtual 8086 mode	VIP – Virtual Interrupt Pending

٣- مسجلات المقاطع Segment Registers وعددها ستة مسجلات كل منها بطول 16 bit بالأسماء المبينة بالشكل 4-14.



الشكل4-14: مسجلات المقاطع في معالجات بنتيوم

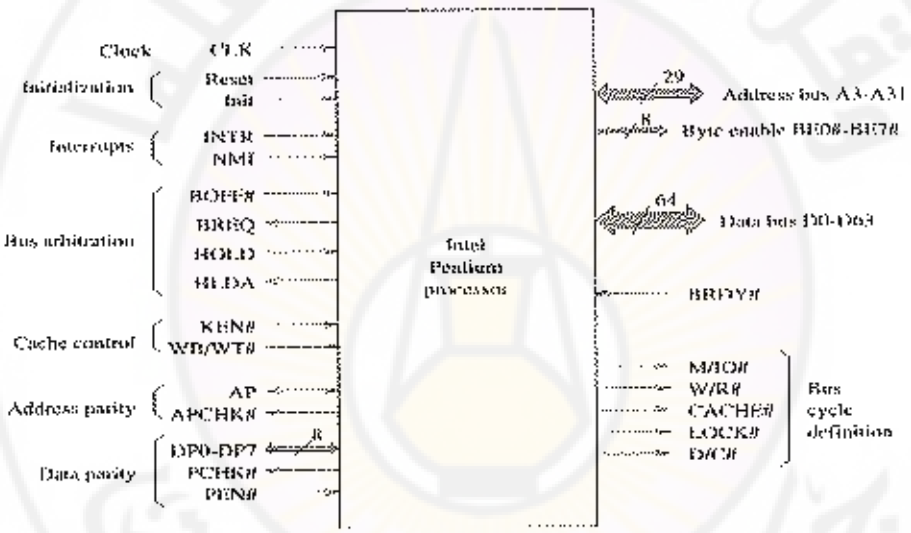
وهي مخصصة للتعامل مع مقاطع البرنامج (Code, Stack, Data, Extra)، حيث إن برامج المعالج يمكنها الولوج إلى المقاطع الستة بأي وقت.

٤- مسجلا دليل Index Registers هما SI و DI كل منهما بطول 16 bit تستخدم بتعليمات السلاسل ويمكن استخدامها كمسجلات للمعطيات العامة. ويوجد منها ضمن معالج بنتيوم النسخ ESI و EDI بطول 32 bit لكل منها.

٥- مسجلا مؤشر Pointer Registers هما SP و BP كل منهما بطول 16 bit تستخدم للتعامل مع المكدس، ويوجد منها ضمن معالج بنتيوم النسخ ESP و EBP بطول 32 bit لكل منها.

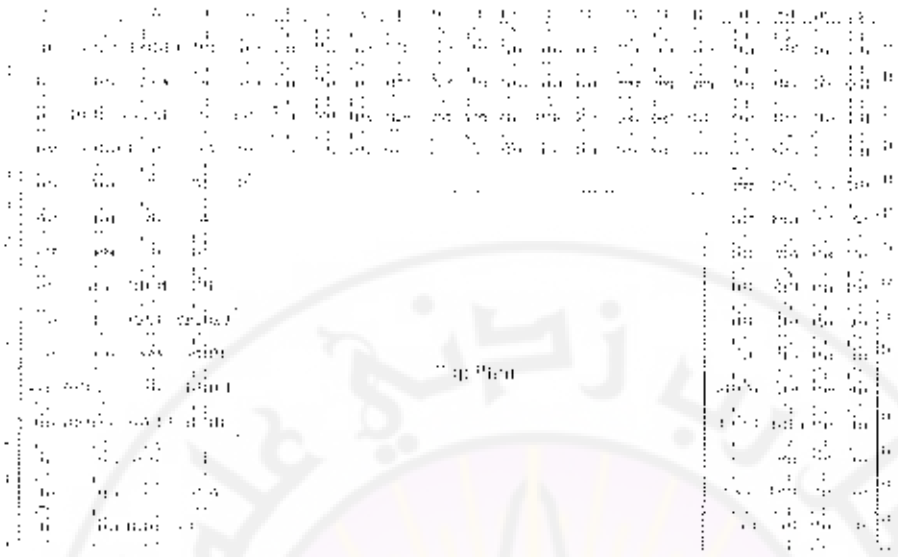
١٤-٤ إشارات معالج Pentium الأساسي

يبين الشكل 5-14 توزع الإشارات الأساسية لمعالج بنتيوم بشكل صندوقي.



الشكل5-14: مجموعات الإشارات الأساسية للمعالج Pentium

يغلف المعالج بأشكال عدة لا داعي لعرضها خاصة وإن التغليف يختلف باختلاف اسم ونوع المعالج، الإشارات المبينة بالشكل أعلاه هي إشارات أساسية بمعظم معالجات البنتيوم. يبين الشكل 6-14 الوجه العلوي لتوزع مآخذ معالج Pentium للتركيبة المبينة بالشكل 1-14.



الشكل6-14: الوجه العلوي لمآخذ المعالج Pentium 510/60, 567/66

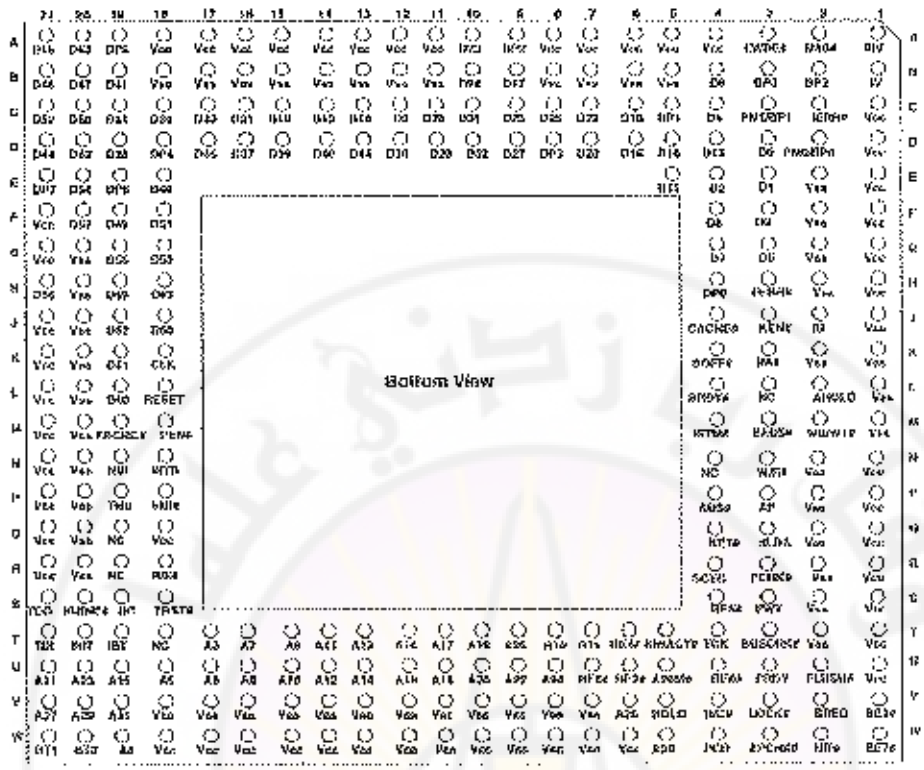
ويبين الشكل 7-14 الوجه السفلي لنفس المعالج.

اسماء إشارات المعالج Pentium 510/60, 567/66 وفق توزعها على مآخذه، وعددها 273 مأخذاً موجودة بالملحق٢.

يوجد على 168 مآخذ من مآخذ المعالج اشارة واحدة لكل مآخذ . ويخصص 50 مأخذاً للتغذية Ucc و 49 مأخذاً للأرضي Uss و6 مآخذ غير موصولة Not Connected .

توزع إشارات معالجات البنتيوم في مجموعات، نعرض باختصار مفهوم كل منها كالتالي:

١- ممر العناوين Address Bus : وتتكون من 29 lines من $A_3 - A_{31}$ أي تستثني الاشارات الثلاثة $A_0 - A_2$ التي تلحق بممر المعطيات بعرض 8 byte.



الشكل7-14: الوجه السفلي لمآخذ المعالج Pentium 510/60, 567/66

٢- بايتات التأهيل Byte Enable : وتمثل ثماني إشارات خرج #BE7 - #BE0 (حيث إن الرمز # يعني بأن الإشارة فعّالة بالصفر). تعرف مجموعة بايتات القراءة والكتابة، بحيث إن:

- #BE0 لأجل البايت الأدنى أهمية $D_0 - D_7$
- #BE1 لأجل البايت التالي $D_8 - D_{15}$
- #BE7 لأجل البايت الأعلى أهمية $D_{56} - D_{63}$

٣- ممر المعطيات Data Bus: بعرض 64 bit من $D_0 - D_{63}$ لنقل بالاتجاهين.

٤- إشارة #BRDY (Bus Ready): تمثل إشارة دخل وتستخدم لتوسيع حلقة الممر، وتعريف حالات الانتظار.

٥- تعريف حلقة الممر Bus Cycle Definition : وتضم خمس إشارات جميعها خرج من المعالج وهي:

- M/IO# (Memory/IO): وتعرف حلقة الممر – ذاكرة أو بوابات I/O
- W/R# (Read/ Write): وتميز بين حلقة القراءة أو الكتابة
- CACHE# (Cache ability): وتؤشر بحلقة القراءة لـ Cache ability
- LOCK# (Bus Lock): وهي إشارة تفيد بتنفيذ الـ Semaphores وتستخدم في حلقة read–modify–write
- D/C# (Data/Code): وتميز بين البيانات Data والشفرة Code.

٦- CLK (Clock): إشارة دخل للمعالج من مولد الساعة لتأمين ساعة للنظام

٧- التهيئة Initialization وتضم إشارتي دخل للمعالج هما:

- RESET : تصفر المعالج وتبدأ التنفيذ من العنوان FFFFFFF0H وتبطل كل ذواكر كاش الداخلية.
- INIT (Initialization): تعمل مثل RESET ولكن ذواكر كاش الداخلية ومسجلات FP لا تمسح. وبعد تشغيل التغذية يستخدم RESET وليس INIT.

٨- المقاطعة Interrupts وتضم إشارتي دخل للمعالج هما:

- INTR (Interrupt): وهي إشارة مقاطعة خارجية
- NMI (Nonamskable Interrupt): وهي إشارة مقاطعة خارجية لا تحجب.

٩- تحكيم الممر Bus Arbitration وتضم أربع إشارات هي:

- BOFF# (Backoff): وهي إشارة دخل تجهض كل حلقات الممر المعلقة وتعوم الممر ، وهي تفيد في حل الجمود بين سيدين للمر
- BREQ (Bus Request): طلب الممر وهي إشارة خرج من المعالج.
- HOLD (Bus Hold): وهي إشارة دخل للمعالج يتم عبرها طلب الممر فتقوم بتعويم الممر بعد إنهاء حلقات الممر القائمة
- HLDA (Bus Hold Acknowledge):وهي إشارة خرج يشير من خلالها إعطاء التحكم لسيد محلي آخر ، حيث إن بنتيوم يكمل بالتنفيذ لكن من ذاكرة كاش الداخلية.

١٠- التحكم بالكاش Cache Control وتضم إشارتي دخل للمعالج هما:

- KEN# (Cache Enable): تأهيل الكاش، أي إذا كانت هذه الإشارة مفعّلة يتم نقل الحلقة (الدورة) الحالية إلى خط الكاش لتعبئته.
- WB/WT# (Write-back/Write-through): ويحدد من خلالها السياسة التي يجب أن تستخدم للكتابة بذاكرة الكاش.

١١- تكافؤ العناوين Address Parity وتضم إشارتين هما:

- تكافؤ العنوان AP (Address Parity): وهي إشارة بالاتجاهين للإشارة إلى تكافؤ العنوان المبيّن خلال حلقة الاستفسار.

١٢- تكافؤ المعطيات Data Parity وتضم عشرة إشارات هي:

- 8 إشارات لبيانات التكافؤ $DP_0 - DP_7$ (Data Parity) للنقل بالاتجاهين التكافؤ الزوجي لـ 8 Byte من البيانات، بحيث: DP_0: $D_0 - D_7$

.....

DP_7: $D_{56} - D_{63}$

- إشارة لاختبار التكافؤ PCHK# (Parity Check): وهي إشارة خرج تشير إلى نتيجة اختبار التكافؤ عند قراءة البيانات. ويختبر التكافؤ فقط من أجل البايت الصحيح، ويؤشر عليه بوساطة إشارات BE#.
- وإشارة لتأهيل التكافؤ PEN# (Parity Enable): وهي إشارة دخل المعالج تحدد فيما إذا كان من اللازم اختبار التكافؤ.

ونشير هنا إلى أن المعالج Pentium يدعم نمطي التشغيل:

١- الحقيقي وفيه يستخدم 16 bit Address ويشغل برامج المعالج 8086 ويتصرف وكأنه معالج 8086 سريع.

٢- المحمي وهو يمثل نمط تشغيل بـ 32 bit وهو النمط الموطني للمعالج بنتيوم ويدعم التقطيع Segmentation والتصفيح Paging.

١٤-٥ إصدارات معالج Pentium

نعرض فيما يلي باختصار أهم منتجات شركة إنتل من معالجات Pentium كدليل سريع ومرتب بحسب ظهور كل من هذه العائلات:

Intel486™ Processors and Earlier Pentium:

1- 32-bit processors: P5 micro architecture
- 1.1 Original Pentium
- 1.2 Pentium with MMX Technology

2- 32-bit processors: P6/Pentium M micro architecture
- 2.1 Pentium Pro
- 2.2 Pentium II
- 2.3 Celeron (Pentium II-based)
- 2.4 Pentium III
- 2.5 Pentium II and III Xeon
- 2.6 Celeron (Pentium III Coppermine-based)
- 2.7 Celeron (Pentium III Tualatin-based)
- 2.8 Pentium M
- 2.9 Celeron M
- 2.10 Intel Core
- 2.11 Dual-Core Xeon LV

3- 32-bit processors: NetBurst micro architecture
- 3.1 Pentium 4
- 3.2 Xeon
- 3.3 Mobile Pentium 4-M
- 3.4 Pentium 4 EE
- 3.5 Pentium 4E
- 3.6 Pentium 4F

4- 64-bit processors: IA-64
- 4.1 Itanium
- 4.2 Itanium 2

5- 64-bit processors: Intel 64 -- NetBurst micro architecture
- 5.1 Pentium 4F
- 5.2 Pentium D

- 5.3 Pentium Extreme Edition
- 5.4 Xeon

6- 64-bit processors: Intel 64 – Core micro architecture

- 6.1 Xeon
- 6.2 Intel Core 2
- 6.3 Pentium Dual-Core
- 6.4 Celeron
- 6.5 Celeron M

7- 64-bit processors: Intel 64 – Nehalem micro architecture

- 7.1 Intel Pentium
- 7.2 Core i3
- 7.3 Core i5
- 7.4 Core i7
- 7.5 Xeon

8- 65-bit processors: Intel 64

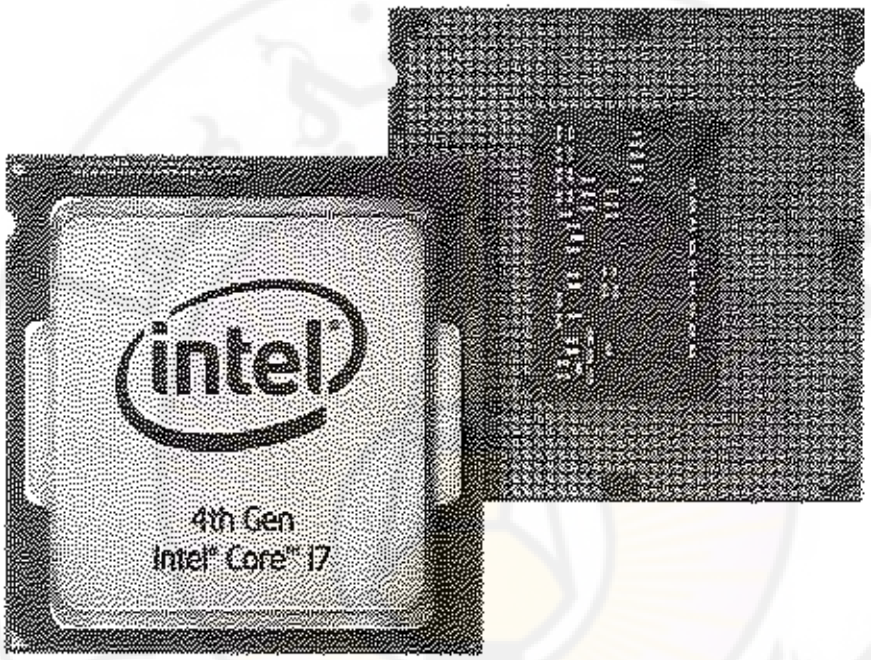
Sandy Bridge / Ivy Bridge micro architecture

- 8.1 Celeron
- 8.2 Pentium
- 8.3 Core i3
- 8.4 Core i5
- 8.5 Core i7

علما أن المواصفات الأساسية للمعالج Core i7 (Introduced,2011) من آخر سلسلة لمعالجات بنتيوم هي المواصفات المبينة أدناه، كما أن المظهر العام لهذا المعالج يبدو كما هو مبين بالشكل 8-14.

- Sandy Bridge – 32 nm process technology
  - 4 physical cores/25 threads
  - 32+32 Kb (per core) L1 cache
  - 256 Kb (per core) L2 cache
  - 8 MB L3 cache

- 995 million transistors
- Socket 1155 LGA
- 2-channel DDR3-1333
- Integrated GPU



الشكل8-14: المظهر العام لمعالج Core i7 من إنتل

- All variants have base GPU frequencies of 850MHz and peak GPU turbo frequencies of 1.35 GHz
- Variants ending in 'K' have Intel HD Graphics 3000
- Variants
  - i7-2600S – 2.8 GHz/3.8 GHz Turbo Boost
  - i7-2600 – 3.4 GHz/3.8 GHz Turbo Boost
  - i7-2600K – 3.4 GHz/3.8 GHz Turbo Boost

        - i7-2700K – 3.5 GHz/3.9 GHz Turbo Boost
- Sandy Bridge-E – 32 nm process technology
    - Up to 8 physical cores/16 threads
    - 32+32 Kb (per core) L1 cache
    - 256 Kb (per core) L2 cache
    - Up to 20MB L3 cache depending on model number
    - 2270 million transistors
    - Socket 2011 LGA
    - 4-channel DDR3-1600
    - Variants
        - i7-3820 – 3.6 GHz/3.8 GHz Turbo Boost, 4 cores, 10 MB L3 cache
        - i7-3930K – 3.2 GHz/3.8 GHz Turbo Boost, 6 cores, 12 MB L3 cache
        - i7-3960X – 3.3 GHz/3.9 GHz Turbo Boost, 6 cores, 15 MB L3 cache
        - i7-3970X – 3.5 GHz/4.0 GHz Turbo Boost, 6 cores, 15 MB L3 cache.

٦-١٤ مقارنة بين المعالجات

بعد العرض السابق حول معالجات إنتل والتي بقيت متحكمة بالسوق العالمية لفترة طويلة ظهرت شركات أخرى منافسة على صعيد المعالجات مثل شركة AMD و ARM وغيرها الكثير من الشركات الأخرى.

لعبت وسائل الدعاية والإعلان دورا كبيرا في انتشار منتجات كل من هذه الشركات التي تعتبر الأوسع انتشارا عن غيرها من المنتجات الأخرى، علما أن شركة AMD هي ألمانية

وشركة ARM هي بريطانية، وعلى أثر ذلك يلاحظ مثلا انتشار واسع لمعالجات Intel الأمريكية في منطقة الشرق الأوسط، أما الانتشار الواسع لمعالجات AMD و ARM فيقتصر على أوروبا.

لكل من المنتجات الثلاثة ميزات خاصة، فمعالجات AMD مناسبة لتطبيقات الألعاب مثلا وهي رخيصة الثمن بالمقارنة مع معالجات إنتل، لكن معالجات إنتل هي الأكثر انتشاراً. وبمقارنة ARM مع إنتل من حيث الأداء، يلاحظ أن منتجات ARM تمتاز بالاستهلاك المنخفض للطاقة وبالتالي انخفاض الحرارة، لأن معمارية ARM تتبع طريقة تصميم RISC (أوامر بسيطة – سرعة في التنفيذ)، أي أن تصميم المعالج بسيط ويتم وضع العبئ على نظام التشغيل والبرامج (تعتمد على الحل البرمجي) على خلاف المعمارية المعتمدة لدى X86 والتي تتبع طريقة تصميم CISC (تصميم معقد يلزمه عدد أكبر من الترانزستورات وترددات مرتفعة لتعويض البطىء مما يجعل استهلاك الطاقة كبير أيضا) أي أن إنتل تعتمد طاقم تعليمات قوي ومعقد مع المعالج ليحمل عبئ توزيع وترتيب عن نظام التشغيل والبرامج (أي تعتمد على العتاد).

تتميز معالجات Intel بالسرعة في إتمام العمليات عموماً ولكن عندما نتحدث عن السرعة عند Intel أنها أكبر من السرعة عند AMD فإن لهذا الجانب عيوب تتمثل في أنه قد يحدث خلل أو فقدان احد البيانات بسبب سرعة المعالج وهذا يجعل المعالج يتجمد بالمفهوم العام (يصبح بطئ) مما يستدعي إعادة تشغيل العملية.

وبالعودة للتركيز على منتجات إنتل من المعالجات، يلاحظ أن المنتجات الأخيرة لـ Intel أتت تحت تسميات مثل Dual Core والأخرى Core2Duo وهذا بدوره له تسميه أخرى هي Core Duo. ولنوضح فيما يلي باختصار كل من هذه التسميات:

١- معالجات Dual Core Processors

يطلق هذا الاسم على أي نوع من أنواع المعالجات التي تستخدم نواتين مدمجتين في معالج واحد . أول إنتاج لهذه التقنية كان مع الأجهزة التي تحمل معالج من نوع Pentium D.

٢- Core2Duo

هذه التقنية هي التطور الطبيعي للتقنية السابقة وهي Dual Core التي أصبحت تستخدم

لاجهزة الكمبيوتر العادي واجهزة الكمبيوتر المحمول أيضا ، ومن مميزاتها انها اصبحت تحكم بشكل افضل في استخدام الطاقة. المعالجات التي تحمل نواة مزدوجة (بالترتيب حسب ظهورها) هي: Core2duo ، Dual Core ، Pentium D.

بعد ذلك طرح في الاسواق من شركة intel معالجات Core i حيث i متبوعة برقم يقصد به عدد التطبيقات Application الذي يقدر المعالج على تشغيلها بأن واحد وبنفس السرعة نشير فيما يلي باختصار إلى أهم الفوارق بين آخر ثلاث معالجات i7-i5-i3 التي ظهرت بالفترة ما بين أخر 2009 وأوائل شهور 2020 حيث أن:

*- Core i3: هو معالج مشابهة لمعالج Core 2 Duo ولكن الفرق بينهما أن الأخير يحتوي على شريحتين بداخله وخطين معالجة ، أما الـ i3 فيحتوي على شريحتين بداخله (أي معالجين) وأربع خطوط معالجة وطبعاً مستوى متقدم من الذاكرة العشوائية ولكن بدون خاصية الـ Turbo الخاصة بإنتل...

*- Core i5: يندرج تحت هذا المعالج عدة موديلات مقسمة إلى مجموعتين: الأولى- تحتوي على شريحتين وعلى أربع خطوط معالجة ولكن بمستوى Turbo عادي أما المجموعة الثانية- فتحتوي على أربع شرائح (أي أربع معالجات في معالج واحد) وأربع خطوط معالجة ولكن بمستوى متقدم لخاصية الـ Turbo والذاكرة العشوائية أيضاً...

*- Core i7: يعتبر هذا المعالج ثورة في عالم الحوسبة لاحتوائه على أربع شرائح في داخله ولكن بـ 8 خطوط معالجة لذلك يظهر في نظام التشغيل أنه يحتوي على 8 أنويه (أي وكأنه 8 أجهزة كمبيوتر في جهاز واحد)، وبالطبع يأتي هذا المعالج بمستوى متقدم من الذاكرة العشوائية وبنظام Turbo عالي الأداء.

يفترض بعد هذه المعالجات المذكورة أعلاه ظهور المعالج Core i9 والذي يتوقع أن يحتوي على ستة شرائح وعلى ١٢ خط معالجة.

- وعلى صعيد تصنيع الحواسيب يوجد معالجات core موجهة للحواسيب المحمولة (اللابتوبات). وأخرى لموجهة للحواسيب المكتبية (بالطبع الأداء أقوى هنا)

حيث يمكننا تلخيص الفوارق بين هذه المعالجات بالنسبة للحواسيب المحمولة بالجدول أدناه

Core i7	Core i5	Core i3	
2-4	2	2	عدد المعالجات
1.1-2.8 GHz	1.1-2.6 GHz	1.1-2.6 GHz	السرعة
4.0 – 8.0 MB	3.0 MB	3.0 MB	الذاكرة المخبأة
لايوجد	متوفر	لايوجد	Turbo boost
متوفر	متوفر	متوفر	Hyper-Threading
متوفر	متوفر	متوفر	HD Graphic

عدد العمليات (الخيوط) threads التي يمكن استخدامها في معالجات i3 و i5 هي ٤ خيوط فقط، بينما معالج i7 فبالإمكان شراء معالج يستخدم 8 خيوط.

ويمكننا تلخيص الفوارق بين المعالجات المنتجة للاستخدام في الحواسيب المكتبية بالجدول

Core i7	Core i5	Core i3	
6	2-4	2	عدد المعالجات
2.3-3.46 GHz	2.3-3.6 GHz	2.5-3.3 GHz	السرعة
8.0 – 12.0 MB	3.0 – 8.0 MB	3.0 – 4.0 MB	الذاكرة المخبأة
لايوجد	متوفر	لايوجد	Turbo boost
متوفر	متوفر	متوفر	Hyper-Threading
لايوجد	متوفر	متوفر	HD Graphic

عدد العمليات (الخيوط) threads التي يمكن استخدامها في معالجات i3 و i5 هي ٤ خيوط فقط، بينما معالج i7 فبالإمكان شراء معالج يستخدم 12خيوط. قد يبدو أن السرعات متقاربة ، ويكاد لا يوجد فرق بين core i3 أو core i7. و لكن الفرق الكبير الذي يحدث فارق في السرعة هما ذاكرة المخبأة وعدد العمليات (الخيوط) . إذ أن هذه الخاصيتين كلما زاد حجمهما أو عددهما كلما أصبح المعالج أسرع . لذا دائما عند اختيار المعالج المناسب يجب الانتباه لحجم الذاكرة المخبأة و لعدد العمليات الممكنة.

الجزء الثالث: البرمجة بلغة الأسمبلي

ويضم أربعة فصول

الفصل الخامس عشر: المفاهيم الأساسية للبرمجة بلغة الأسمبلي

الفصل السادس عشر: توصيف تعليمات نقل المعطيات والتعليمات الحسابية والمنطقية

الفصل السابع عشر: توصيف تعليمات نقل البرنامج والتحكم بالمعالج

الفصل الثامن عشر: البرمجة البنوية وأدوات تطوير برامج لغة الأسمبلي

Damascus University

وقل رب زدني علما
جامعة دمشق
Damascus University

يخصص هذا الجزء لبرمجة المعالجات الصغرية بلغة التجميع (الأسمبلي)، حيث كما هو معلوم أن لكل معالج صغري جملة من التعليمات (Instruction Set) الخاصة به وأن عدد هذه التعليمات يختلف من معالج لآخر. بالتالي إن كتابة برنامج بلغة الأسمبلي لمعالج من شركة ما مثل Intel بطول كلمة 16 bit ينفذ بسهولة على حاسوب مبني على معالج آخر من نفس الشركة مثل المعالج 80486 بطول كلمة 32 bit لوجود التوافق Compatibility بين المعالجين، ولكن لا يمكن تنفيذ هذا البرنامج مباشرة على منظومة مبنية على معالج من شركة أخرى مثل Motorola أو غيرها لعدم وجود التوافقية بالتعليمات بين معالجين من شركتين مختلفتين.

وإذا تغاضينا عن عرض الـ Instruction Set للمعالجات الأولى من شركة Intel بطول 8 bit (أقدمها)، سنركز بهذا الجزء على التعريف بتعليمات المعالج 8086 الذي يعدّ العنصر الأول بالعائلة 8086 Family وعلى آلية كتاب برامج باستخدام تعليمات هذا المعالج التي تصلح للعمل مع كل المعالجات الأحدث.

يلزم بالنسبة للبرمجة بلغة الأسمبلي، بالإضافة إلى إيجاد التعليمات اللازمة لكتابة الجزء الأساسي من البرنامج، التعرف أيضاً على بعض التعليمات الإضافية (المعائدة للمجمع المستخدم) التي نحتاجها لتأسيس وتهيئة أجزاء مختلفة من النظام مثل مسجلات القطع والأعلام (المؤشرات)، وأجهزة المنافذ القابلة للبرمجة وغيرها من الأمور المتفرقة. يتم بعد كتابة برنامج بلغة الأسمبلي، تحويله إلى لغة الآلة وتحميله بالحاسوب الصغري لتنفيذه، حيث يوجد طريقتان للتحويل، إما باستخراج الشيفرة الثنائية (op code) لكل تعليمة باستخدام نماذج التشفير Templates كما سنجد لاحقا، وإما باستخدام المجمع. وهنا نشير أيضاً إلى أن معالجات إنتل صممت بشكل يحتاج إلى استخدام مفهوم تقطيع الذاكرة Memory Segmentation لسببين رئيسين هما:

الأول: عند العمل مع قطعة ذاكرية مؤلفة من 64 KB كل واحدة على حدة، فإن المعالج 8086 يستطيع التعامل فقط مع عناوين فعّالة Effective Address مؤلفة من 16 bit للولوج إلى موقع في الذاكرة، لأن مخطط التقطيع للمعالج يعالج ويخزن مركبات العناوين ذات الطول 16 bit فقط (بما يتناسب مع أطوال المسجلات لديه).

الثاني: عمليات المشاركة في الزمن لمنظومات الحواسيب الصغرية التي يتشارك فيها عدة مستخدمين بوحدة معالجة مركزية واحدة. يخصص بالحواسيب الصغرية التي تستخدم وحدات معالجة مركزية من عائلة المعالج 8086 لكل برنامج فترة زمنية محددة 20 ms مثلا. بالتالي إذا كانت الوحدة CPU قد خصصت لبرنامج المستخدم الأول 20 ms فإنها ستعمل مع المستخدم التالي 20 ms أيضاً، وبعد أن تنهي الوحدة CPU التعامل مع كل المستخدمين فإنها تعود ثانية إلى المستخدم الأول وهكذا. وعند كل انتقال للوحدة CPU من برنامج مستخدم إلى برنامج آخر ينبغي عليها الولوج إلى مقطع جديد للشيفرة ومقاطع جديدة للمعطيات وهذا ما يجعل عملية الانتقال أكثر سهولة باستخدام مفهوم التقطيع، بحيث يعين لكل برنامج مستخدم مجموعة مستقلة من القطع المنطقية، وسيحتوي برنامج المستخدم على الإزاحات المحددة عن قواعد هذه القطع. وبالتالي للانتقال من أحد برامج المستخدم إلى برنامج مستخدم ثان، يكفي إعادة تحميل المسجلات CS, DS, SS, ES بعناوين قاعدة القطع المعينة إلى برنامج المستخدم الثاني. وهكذا تسهل عملية التقطيع عملية الانتقال من برنامج مستخدم إلى برنامج مستخدم آخر، وتسهل عملية حفظ برامج المستخدمين ومعطياتهم كل بمعزل عن الأخر.

الفصل الخامس عشر: المفاهيم الأساسية للبرمجة بلغة الأسمبلي

نتعرف بداية على إحدى تعليمات المعالج 8086 لنوضح من خلالها بعض المفاهيم المتعلقة بالبرمجة بلغة الأسمبلي، ولتكن تعليمة النقل MOV (Move Data)، حيث إن الصيغة العامة لهذه التعليمة هو:

MOV Destination, Source وتعني تحريك (أو نقل) المعطيات من مكان محدّد (المصدر Source) إلى مكان آخر محدّد (الهدف Destination)، وتستطيع هذه التعليمية نقل معطيات بطول Byte أو Word.

يمكن أن يكون المصدر في التعليمة مسجلا، أو موقع ذاكرة، أو عدداً فورياً (مباشر) Immediate Number، أما الهدف فيمكن أن يكون مسجلا أو موقع ذاكرة فقط. لا يجوز بكل الحالات بالنسبة لتعليمة MOV أن يكون المصدر والهدف مواقع ذاكرية معاً في نفس التعليمة، كما ويجب أن يكون كل من المصدر والهدف من نفس النوع، أي إما كلاهما من نوع Byte أو كلاهما من نوع Word.

فإذا كتبنا الصيغة العامة للتعليمة MOV بالشكل: MOV op1, op2 بحيث إن op1 (Operand 1) تمثل الهدف Destination يمكننا إيجاز التوافقيات الممكنة لكل من op1 و op2 بالنسبة لهذه التعليمة وغيرها بالتوافقيات المبينة بالجدول 1-15.

الجدول 1-15 التوافقيات الممكنة لمعاملات تعليمة MOV في المعالج 8086.

Op1	Op2
Register	Number
Register	Register
Register	Memory
Memory	Register
Memory	Number

وبأخذ الوظيفة المسندة لكل مسجل من مسجلات المعالج 8086، يمكننا عرض الأشكال الممكنة لهذه التعليمة من أجل أحد المسجلات وفق التالي:

1. MOV CL , 00H
2. MOV CH , 00H
3. MOV CX , 00H
4. MOV CL , CH
5. MOV CH , CL
6. MOV CX , [100H]
7. MOV [100H], CX
8. MOV CX, AX ...

يفهم من هذه الأشكال لنفس التعليمة من أجل المسجل CX وأقسامه CL, CH ، بأن التعليمة 1. MOV CL , 00H تنقل معطيات بطول Byte (00H) إلى القسم السفلي من المسجل CX أي إلى المسجل CL فقط.

ويتم بالتعليمة الثانية 2. MOV CH , 00H نقل Byte إلى القسم العلوي من CX أي إلى المسجل CH وهكذا، حيث يتم بالتعليمة الثالثة نقل Word إلى المسجل CX بطول 16 bit دفعة واحدة ، لان طول الهدف هو 16 bit.

وبالتعليمة MOV CL , CH يتم نقل (نسخ Copy) لمحتوى المسجل CH بطول 8 bits إلى المسجل CL بطول 8 bit وبعد تنفيذ التعليمة سيصبح محتوى المسجلين نفسه. والتعليمة MOV CH , CL تنفذ العكس.

وأخيرا التعليمة MOV CX , [100H] تقوم بنقل محتوى موقع بالذاكرة ذي العنوان 100H بطول 16 bits إلى المسجل CX، ويمكن بتعليمة مماثلة نقل محتويات نفس الموقع بطول 8 bits إلى أي من المسجلين CL أو CH. (بالطبع تعليمة من الشكل MOV [100H], CX تقوم بتنفيذ عكس ما سبق).

من الضروري أيضاً أثناء استخدم المعاملات بتعليمات المعالج 8086 وبالأخص إن كان المعامل مسجلا أن نعرف الوظيفة الخاصة بكل مسجل على حدة، فمثلا المسجلات AX, BX, CX, DX هي للأغراض العامة ولكن لكل منها وظيفة خاصة به ينفرد فيها عن بقية المسجلات، إذ إن المراكم AX مثلا يستخدم بشكل ضمني في بعض التعليمات كالضرب، كما أن المسجل CX يستخدم للعَد (في حالة التعامل مع الحلقات....) كعداد. وإن المسجل BX هو الذي يستخدم للتعامل مع الذاكرة، بمعنى إذا أردنا جلب Byte من

الذاكرة إلى المعالج وبالعكس، يجب أن يوضع عنوان المعطيات في المسجل BX وبعد ذلك يتم التعامل مع محتواه الذي يعبر عنه بأقواس متوسطة. وبناء على ذلك فإن تعليمة MOV بالصيغة MOV [BX] , [100H] خاطئة، لأنه لا يجوز أن يكون معاملا التعليمة MOV من نوع ذاكرة.

تستخدم التعليمة MOV في نقل بايت سلسلة بإضافة حرف S من String إلى رمز التعليمة وبإضافة حرف B من Byte إلى التعليمة الأساسية MOV أو بنقل كلمة سلسلة بإضافة حرف W من Word ويصبح الشكل العام للتعليمة:

MOVS destination , source ولهذه التعليمة شكلان، هما: MOVSB و MOVSW التي تقوم بنسخ Byte أو Word من موقع في DS إلى موقع في القطعة الإضافية ES. بالطبع يجب أن تكون إزاحة Offset بايت أو كلمة المصدر في DS موجودة في المسجل SI (وإزاحة المقطع ES في المسجل DI).

وبعد أن ينقل Byte أو Word يضبط المسجلان SI و DI بشكل آلي ليشيرا إلى المصدر والوجهة التاليتين، بحيث إذا كان مؤشر الاتجاه DF صفراً سيتم عندئذ زيادة SI و DI بمقدار 1 بعد نقل Byte و2 بعد نقل كلمة. ويتم العكس عندما DF = 1.

١٥-١ أنماط العنونة لدى المعالج 8086

يستخدم المعالج 8086 عدة طرق للولوج إلى المعطيات التي يتعامل معها، تسمى 'بأنماط العنونة Addressing modes ' ويشار عادة إلى نمط العنونة ضمن عبارات لغة الأسمبلي في التعليمة نفسها. وخلال التطرق لأنماط العنونة يجدر التمييز بين كل من العنوان الفعّال Effective Address الذي يكون في التعليمة بطول 16 bit والعنوان الفيزيائي للذاكرة الذي يحسب وهو بطول 20 bit.

يملك المعالج 8086 ثمانية أنماط مسجلية Register Mode العنونة (بعدد مسجلات الوحدة EU، أي: AX, BX, CX, DX, SP, BP, SI, DI) و 24 نمط ذاكري Memory Mode، نستعرض باختصار مفهوم البعض منها وفق التالي:

1- العنونة الفورية Immediate Addressing Mode

عند استخدام طريقة العنونة الفورية فإن المعامل Operand يعدّ جزء من التعليمة، كما في المثال MOV AL, 50H، حيث القيمة 50H هي جزء من شيفرة التعليمة.

2- العنونة باستخدام المسجلات Register Addressing Mode

ويتم الإشارة هنا إلى المعاملات المرتبطة بالتعليمة باستخدام مسجلات المعالج، كما في المثال: MOV AL, BL وينفذ عادة هذا النوع من التعليمات بسرعة عالية، لأنها لا تتطلب الرجوع إلى الذاكرة لإحضار البيانات اللازمة.

3- العنونة المباشرة Direct Addressing Mode

وتستخدم هذه الطريقة للتعامل مع مواقع مباشرة في الذاكرة. ولتوضيح ذلك أفترض أن X معرفة في مقطع البيانات وأن قيمة إزاحتها (أي بعدها عن بداية المقطع) 3A5H فإن الصيغة MOV AX, X تستخدم العنونة المباشرة، لان X تشير إلى عنوان البيانات المطلوب بشكل مباشر .

4- العنونة المباشرة المؤشرة Direct Indexed Addressing Mode

ويتم الوصول إلى موقع المعامل في الذاكرة باستخدام هذا النمط (الطريقة) باستخدام مسجلي الدليل SI أو DI مضافا لها (أو مطروحا منها) قيمة محددة، كما في المثال MOV AX, [SI + 5]، بمعنى آخر: معامل التعليمة بطول 8 bit أو 16 bit يجمع إلى محتوى المسجل(SI, or DI)، وقيمة الناتج تشير إلى الموضع الذي تستقر فيه المعطيات.

5- العنونة غير المباشرة باستخدام المسجلات Register Indirect Addressing

يتم الوصول إلى المعامل المطلوب في الذاكرة عن طريق عنوان يتم الاحتفاظ به في أحد المسجلات BX, BP, SI, DI. فمثلا: التعليمة MOV AX, [BX] تستخدم العنونة غير المباشرة باستخدام المسجلات، لأنها تقوم بنقل المعامل الذي يقع في الذاكرة على العنوان المخزن في المسجل BX إلى المسجل AX.

6- عنونة القاعدة النسبية Base Relative Addressing Mode

وتستخدم في هذه الطريقة مسجلات القاعدة BX أو BP، كما في المثال التالي:

MOV AX, [BX – 5]

7- عنونة القاعدة المؤشرة Base Indexed Addressing Mode

وتستخدم في هذه الطريقة مسجلات التأشير SI أو DI ومسجلات القاعدة BX أو BP للحصول على العنوان الحقيقي، كما في الأمثلة:

MOV AX, [BX + SI + 4]

MOV BX, [BX + DI + 4]

[MOV CX, [BP + SI – 5 أي محتوى المسجل القاعدي (BX, or BP) يجمع إلى محتوى المسجل المؤشر (SI, or DI)، وقيمة الناتج تشير إلى الموضع الذي تستقر فيه المعطيات.

8- نمط العنونة القاعدي Based Addressing Mode

معامل التعليمة بطول 8 bit أو 16 bit يجمع إلى محتوى المسجل القاعدي (BP, or BX)، وقيمة الناتج تشير إلى الموضع الذي تستقر فيه المعطيات.

9- نمط العنونة المؤشر Indexed Addressing Mode

معامل التعليمة بطول 8 bit أو 16 bit يجمع إلى محتوى المسجل المؤشر (SI, or DI)، وقيمة الناتج تشير إلى الموضع الذي تستقر فيه المعطيات.

10- نمط العنونة القاعدي المؤشر بإزاحةBased Indexed with Displacement

معامل التعليمة بطول 8 أو 16 bit يجمع إلى محتوى المسجل القاعدي (BX, or BP) وإلى محتوى المسجل المؤشر (SI, or DI)، وقيمة الناتج تشير إلى الموضع الذي تستقر فيه المعطيات.

١٥-٢ تركيب شيفرة الآلة لتعليمات المعالج 8086

نشير فيما يلي إلى كيفية إنشاء الشيفرة الثنائية لتعليمات المعالج 8086 كي نفهم التمايز بينها، علما أن تشفير تعليمات المعالج هي من مهام المجمع Assembler فمثلا بتعليمة من الشكل MOV CX, Source يوجد 32 طريقة لتحديد المصدر Source في التعليمة. إذ يمكن أن يكون المصدر في أي مسجل من المسجلات الثمانية للمعالج ذات الطول 16 bits، أو في أي موقع ذاكري محدد بأحد أنماط العنونة الـ 24 المستخدمة لعنونة

الذاكرة وأن كل تعليمة محتملة من أصل الـ 32 تعليمة ممكنة تتطلب شفرة ثنائية مختلفة. وإذا كان المسجل CX يشكل المصدر بدلا من الوجهة (الهدف) أي من الشكل: MOV Destination, CX فان هناك 32 طريقة لتحديد الوجه وكل تعليمة محتملة من أصل الـ 32 تعليمة ممكنة لها شفرة ثنائية مختلفة.

يوجد إذاً 64 شيفرة مختلفة لتعليمة MOV إذا استخدمنا CX مصدراً أو وجهة. وبطريقة مماثلة يمكن القول إن هناك 64 شيفرة أخرى مطلوبة لتحديد كل أشكال تعليمة MOV باستخدام المسجل CL مصدراً أو وجهة.

وكذلك 64 شيفرة أخرى إذا استخدمنا المسجل CH مصدراً أو وجهة. وهكذا الحال بالنسبة لبقية مسجلات المعالج. وبما أن المعالج 8086 يملك الكثير من التعليمات فانه ليس من العملي أن نشكل جدولا لكل هذه التعليمات. لهذا السبب يستخدم " قالب Template' لكل نوع من التعليمات الأساسية وتملاء البيتات ضمن هذا القالب لتعيين نمط العنونة المطلوب ونوع المعطيات. تعرض المراجع المختلفة لشركة Intel القوالب الخاصة بتعليمات المعالج 8086 بصيغتين مختلفتين بعض الشيء.

ويقصد تعرّف كيفية استخدام القوالب الموجودة في جدول معطيات المعالج 8086 سنورد عدة أمثلة باستخدام تعليمة MOV، حيث يبين الشكل 1-15 قالب التشفير (أو صيغة تعليمة MOV) ، حيث يلزم بالحد الأدنى بيتان للتعليمة.

Byte 1								Byte2 ~ Register Select								Byte3	Byte4
1	0	0	0	1	0											LOW	HIGH
Op Code						D	W	MOD		REG			R/M			DISP	DISP

الشكل 1-15: القالب المستخدم من إنتل لتشفير تعليمات المعالج 8086

وبخصوص الرموز الواردة بكل من البايتين الأول والثاني بالشكل أعلاه، نشير إلى أن البيتات الستة العليا للبايت الأول تستخدم من أجل شيفرة العملية Opcode أي نوعها. والرمز D يعني Direction To/From Reg (بمعنى 0 = From, 1 = To) ويستخدم للدلالة على أن المعطيات المطلوبة سوف يتم نقلها إما "إلى" المسجل المعروف في الحقل

REG للبايت الثاني أو "من" ذلك المسجل (فإذا نقلت المعطيات "إلى" المسجل اجعل البت D مساويا "1"، وإذا نقلت "من" ذلك المسجل فاجعل D مساويا "0". الرمز W يعني Byte/Word Data (بمعنى 0 = Byte, 1 = Word) ويستخدم W في الكلمة الأولى ليحدد ما إذا كان النقل سيتم لبايت أو لكلمة وفي هذه الحالة يجب أن يكون أحد المعاملات مسجلا بشكل دائم. وتستخدم ثلاثة بيتات في البايت الثاني REG لتحديد المسجل اللازم. وفيما يلي نبين بالجدول 2-15 الكود المقابل لكل من مسجلات المعالج 8086. الرمزان MOD و R/M بطول 5 bits تخصص لـ Addressing Mode. وبما أن أحد المعاملات في هذه التعليمة يجب ان يكون مسجلا، وأن الحقل MOD المؤلف من 2 bits وكذلك الحقل R/M المؤلف من 3 bits في البايت الثاني لشيفرة التعليمة يستخدمان لتحديد نمط العنونة المطلوب من أجل المعامل الآخر. تعرض نماذج الحقلين MOD و R/M من أجل كل نمط من أنماط العنونة الممكنة $2^5 = 32$ نمطاً نستغني عن إظهارها. ويمكن أن يكون الـ Byte3 بالاسم Low Displacement بالاسم Direct Address – Low Byte و الـ Byte4 بالاسم High Displacement بـ Direct Address – High.

الجدول2-15: الترميز (الكود) المقابل لكل من مسجلات المعالج 8086

If W = 1 16-bit Register		If W = 0 8-bit Register		Segment Register	
AX	000	AL	000	ES	00
CX	001	CL	001		
DX	010	DL	010	CS	01
BX	011	BL	011		
SP	100	AH	100	SS	10
BP	101	CH	101		
SI	110	DH	110	DS	11
DI	111	BH	111		

وكتطبيق مباشر عما تم ذكره أعلاه نعرض المثال التالي حول تشفير إحدى تعليمات المعالج 8086 ولتكن التعليمة: MOV SP, BX

بما أن هذه التعليمة ستقوم بنسخ كلمة من المسجل BX إلى المسجل SP لابد عند بداية التشفير Coding اليدوي لهذه التعليمة من العودة إلى نشرات شركة إنتل لمعرفة الـ Op Code لهذه التعليمة والمكون من 6 bit وهو بالتحديد (100010) وبعد ذلك نكمل كالتالي: نجعل قيمة w مساويا 1 لأننا ننقل كلمة.

وإذا فكرنا في نسخ كلمة إلى SP نجعل D = 1، ونضع 100 في الحقل REG لتمثيل SP ، وسيكون MOD مساويا إلى 11 وذلك لتمثيل نمط العنونة المسجلية. نجعل الحقل R/M مساويا إلى 011 لتمثيل المسجل الآخر BX وبذلك تصبح الشيفرة النهائية للتعليمة المشار إليها أعلاه هو: 1000 1011 1110 0011 وهكذا الحال بالنسبة لكل تعليمة، حيث إن المجمع هو الذي يقوم بهذه الأعمال.

٣-١٥ توجيهات تسمية العناوين والمعطيات

يتألف البرنامج بلغة الأسمبلي من ثلاثة مقاطع على الأقل وقد يتألف من أربعة مقاطع من الذاكرة المعنونة بالمعالج 8086 كل منها بسعة 64KB وهذه المقاطع هي: المقطع CS لحفظ تعليمات البرنامج والمقطع SS للمكدس والمقطع DS لتخزين البيانات. وقد يلزم ببعض البرامج الطويلة التي تستخدم السلاسل مقطع رابع هو ES. علاوة على أن التسلسل المعروف في الخوارزميات التي تصاغ بإحدى لغات البرمجة العالية المستوى لا يبدو مشابها جدا لبرنامج بلغة التجميع، لأنه أثناء معالجة المعطيات التي يتعامل معها برنامج الأسمبلي، من أولى الأشياء التي يجب التفكير بها هو تجهيز

بنية المعطيات Setting up the data structure بالإجابة عن تساؤلات من الشكل:

- هل ستكون هذه المعطيات موجودة في الذاكرة أم في المسجلات؟
- هل هذه المعطيات من نوع Byte أو Word أو غيرها؟
- كم عدد بنود المعطيات المتوفرة؟
- هل تمثل المعطيات أعداداً موجبة أم أعداداً موجبة وسالبة (signed)؟

- هل تتواجد المعطيات على شكل مصفوفة أم على شكل سجل؟

وتتم الإجابة على مثل هذه الأسئلة باستخدام بعض توجيهات المجمع المستخدم، حيث إن برامج EXE تحتاج مقطع المعطيات لتعريف ثوابت المعطيات ومناطق العمل ومناطق الإدخال والإخراج. ويوفر المجمع إمكانية تعريف عناصر المعطيات في هذا المقطع بأطوال مختلفة وفقا لنوع التوجيه Directive المستخدم لتعريفها.

وتقدم العوامل Pointer آلية في تسهيل وتحليل المعاملات خلال عملية تجميع البرنامج يملك المجمّع Assembler للمعالج 8086 العديد من التوجيهات والعوامل لاستخدامات متعددة. نتعرف فيما يلي التوجيهات والعوامل الأكثر استخداما عند كتابة برنامج بلغة التجميع بشكل متكامل.

بعد انتقاء التعليمات اللازمة لكتابة الجزء الأساسي من برنامج ما، يجب تأسيس أو تهيئة أجزاء مختلفة من النظام، مثل مسجلات القطع والمؤشرات Flags وغيرها، حيث يجب تحميل مسجلات القطع CS, SS, DS, ES بالبتات الستة عشر العليا للعنوان الذاكري الذي نريد أن نبدأ من عنده القطعة المستخدمة في البرنامج.

لا يملك المعالج 8086 تعليمة لنقل عنوان بداية القطعة إلى مسجل القطعة بشكل مباشر، ولذلك سيتم نقل العنوان المطلوب إلى أحد مسجلات الأغراض العامة بالطول 16 bits ومن ثم نسخه إلى مسجل القطعة المطلوبة باستخدام تعليمتي MOV. وبما أن عملية تهيئة مسجلات القطع تعدّ أول عمل يلزم تنفيذه في البرنامج، يمكننا إظهار ذلك وفق التالي: بفرض إننا نريد أن يبدأ مقطع المعطيات مثلا من العنوان 100H فنكتب التعليمة الأولى MOV AX , 100H لتحميل البيتات الستة عشر العليا للعنوان الذي افترضناه لتخزين المعطيات وبعد ذلك بالتعليمة الثانية MOV DS , AX يمكننا نسخ هذه القيمة من المسجل AX إلى مسجل قطعة المعطيات DS.

وهنا نشير أيضاً إلى انه عند كتابة برنامج بلغة الأسمبلي، تتوفر بلغة التجميع جداول أو صفحات من أجل المساعدة في صياغة البرنامج، تسمى بصفحات تشفير لغة التجميع Assembly Language Coding Sheets بترويسة من الشكل:

Address	Data or Code	Label	mnemonic	Operands	Comment
200	B8	Next:	MOV	AX , 7H	مثلا

بحيث يستخدم عمود العناوين Address لتبيان العنوان أو ابتعاد بايت الشفرة أو بايت المعطيات. أما القيمة الفعلية لبايتات الشفرة أو بايت المعطيات فتوضع في عمود الشيفرة أو المعطيات Data / Code Column.

تمثل اللافتة Label العنوان الذي يتم العودة له باستخدام تعليمات القفز والاستدعاء وتتبع بنقطتين (:). ويقابل عمود المختصرات mnemonics رموز التعليمات المستخدمة (أو عمود الشفرة opcode). ويحتوي عمود المعاملات Operands على المسجلات أو المواقع الذاكرية أو المعطيات.

وأخيراً عمود التعليقات Comments يحتوي على شروحات لأمور تتعلق بالبرنامج، حيث كل شرح لا يراد أن يدخل في ترجمة البرنامج يسبق بفاصلة منقوطة(;). تمكننا التوجيهات من التحكم بالطريقة التي يتم بها تجميع البرنامج (ولذلك تسمى أيضاً بالعمليات الزائفة pseudo--operations) وتقوم بعملها فقط عند عملية تجميع البرنامج ولا ينتج عنها أية شيفرة آلة تنفيذية (أي انها تعليمة للمجمع وليس للمعالج). وبما أن المجمّع المايكروي Macro Assembler من شركة INTEL هو المجمع الخاص بالمعالجات 8086/8088/80186/80188 والمجمع MASM للحواسيب الشخصية IBM (Microsoft Macro Assembler) متوافق معه، سنشير هنا فقط للتوجيهات المستخدمة بهذه المجمّعات المتوافقة، لان تغير أي منهما يتطلب الاستعانة بالكاتالوك الخاص بالمجمّع الجديد للتعرف على التوجيهات الخاصة به. ومن مجمل توجيهات MASM سنتعرف بهذه المرحلة على التوجيهات التالية:

1- توجيهة المساواة EQU (Equate)

ويستخدم هذا التوجيه لإعطاء قيمة لاسم ما أو لرمز Symbol معين، وبعد ذلك في كل مرة سيجد فيها المجمّع هذا الاسم المعطى في البرنامج يبدله بالقيمة المكافئة له. مثلا العبارة: Factor EQU 03H عند بداية البرنامج، تعني عند استخدام تعليمة من الشكل

MOV AL , Factor لاحقاً في البرنامج، يقوم المجمّع بتشفير نص هذه التعليمة ويتعامل معها وكأنها من الشكل: MOV AL , 03H.

تكمن الفائدة في استخدام مثل هذا التوجيه EQU في أنه عند استخدام رمز ما مثل Factor في البرنامج عشر مرات مثلاً، يكفي إذا أردنا لسبب ما تغير قيمته أينما يردّ تغير القيمة لمرة واحدة في عبارة EQU وإعادة تجميع البرنامج (أي إعادة ترجمته) حتى يقوم المجمّع آلياً بالبحث عن كل تعليمة استخدمت العامل Factor ليحدّث قيمتها. وبعدم استخدام EQU يفترض إيجاد التعليمات العشر وتغير كل منها على حده. بالطبع قيمة الثابت لا تتغير خلال تنفيذ البرنامج، أما المتحولات فهي الأسماء التي تعطى لبنود المعطيات التي يمكن تغيرها أثناء تنفيذ البرنامج.

2- توجيه التعريف Define

يوجد لهذا التوجيه عدة أشكال هي:

DB – من Define Byte وتعرّف Byte واحد

DW – من Define Word وتعرّف Word واحدة مكونة من 2 Bytes

DD – من Define Double Word وتعرّف كلمة مضاعفة مكونة من 4 Byte

DF – من Define Far Word وتعرّف 6 Byte

DQ – من Define Quad Word وتعرّف كلمة رباعية مكونة من 8 Byte

DT – من Define Ten Words وتعرّف 10 Byte

جميع هذه التوجيهات متماثلة من حيث الوظيفة وتختلف عن بعضها البعض فقط بعدد الـ Bytes المعرفة. بالتالي سنعرفها من خلال التعرف على DB الذي يستخدم لتعريف متحول من النوع Byte أو لحجز موقع تخزين بطول Byte في الذاكرة. فمثلاً العبارة: X DB 40H تعلم المجمّع لحجز Byte واحد في الذاكرة من أجل المتحول X ووضع القيمة 40H في هذا الموقع الذاكري عندما يحمّل البرنامج للتنفيذ. وإذا لم يكن بند المعطيات المطلوب تهيئته واضحاً لنا، ينبغي تحديد ذلك بإشارة الاستفهام " ؟ " وفق الشكل: Z DB ? التي تعني حجز Byte لقيمة غير معروفة كما أن عبارة :

'M' DB تعني حجز Byte لكود ASCII المقابل للحرف M. أما العبارة: Y DB 20H, 40H, 60H فتصرح عن بند معطيات بطول 3 Bytes وتهيئ البايتات الثلاثة بالقيم الثلاثة المحدّدة بالعبارة ابتداء من الاسم Y الذي يشير للقيمة 20H. ويمكن استخدام الصيغة 'Hello' DB التي تعني حجز 5 Bytes للأحرف الخمسة الواردة في الكلمة وراء بعضها البعض، بمعدل Byte واحد يوضع به كود ASCII لكل من هذه الأحرف، ابتداء من الحرف 'H' وكأننا كتبنا الصيغة:

DB 'H' DB 'e' DB 'l' DB 'l' DB 'o'

والصيغة FLD3 DB 11, 12, 13, 14, 15 مثلا تسمح للمجمّع أن يعترف هذه الثوابت في بايتات متجاورة، بحيث إن الاسم FLD3 يقابل البايت الأول بالقيمة " 11 " وبالتالي عند استخدام تعليمة من الشكل: MOV AL , FLD3 + 3 مثلا، يعني تحميل القيمة الثالثة ابتداء من الصفر، أي " 14 " بالمسجل AL.

ويوجد توجيه من الشكل Repeat-Count DUP (Expression) الذي يستخدم لحجز مواضع متعددة. فمثلا: W DB 5 DUP (10H) تعني حجز 5 Bytes بالاسم W كل منها بالقيمة 10H، أي تكرار DUP (مضاعفة Duplicate) ويمكن أن تكون باسم أو بدونه. فمثلا: V DW 3 DUP (?) تعني حجز مكاناً لثلاث كلمات في الذاكرة لتخزن بالاسم V كل منها بقيمة غير معروفة.

3- التوجيه ASSUME الافتراض

قد يمتلك برنامج المعالج 8086 عدة قطع منطقية للشفرة وعدة قطع أخرى للمعطيات وعدة قطع يمكن استخدامها كمكدس. يحدد التوجيه ASSUME (الذي يستخدم في كل برنامج) للمجمّع القطعة المنطقية التي سيستخدمها البرنامج في الزمن المحدّد. فمثلا العبارة ASSUME CS : CODE-Here تخبر المجمّع على ان تعليمات البرنامج موجودة في قطعة منطقية اسمها CODE-Here.

4- التوجيهان Segment و ENDS

يستخدم التوجيه Segment للإشارة إلى بداية قطعة منطقية وأن الاسم الذي يسبق هذا التوجيه، هو الاسم الذي يراد إعطاؤه للقطعة. فمثلا CODE--Here SEGMENT تشير للمجمّع إلى بداية قطعة منطقية تدعى CODE-Here.

يستخدم التوجيه Segment مع التوجيه ENDS لحصر القطعة المنطقية التي تحتوي على الشفرة أو المعطيات (وكأنهما قوسين) وتسمى مجموعة العبارات المنطقية أو التعليمات المحتواة بين التوجيهين Segment و ENDS بالقطعة المنطقية. يوجد لبعض المجمّعات قيود حول الاسم، كأن لا يزيد عن 31 حرف وأن لا يحتوي على Space.

5- التوجيه ORQ الأصل أو نقطة البداية

عندما يجمّع المجمّع جزءاً من تصريحات المعطيات أو عبارات التعليمات، فإنه يستخدم عدّاد المواقع لمعرفة عدد البايتات الموجودة اعتباراً من بداية القطعة في أي وقت. ويوضع عداد المواقع آلياً على 0000H عندما يبدأ المجمّع بقراءة القطعة. يسمح ORG بوضع عداد المواقع على القيمة المطلوبة عند أي نقطة في البرنامج. فمثلا العبارة: ORG 2000H تخبر المجمّع بوضع عداد المواقع على القيمة 2000H ويستخدم الرمز $ لتمثيل القيمة الحالية لعداد المواقع بشكل رمزي ولإعلام المجمّع بإجراء تغيير في عداد المواقع نسبة إلى قيمته الحالية. فمثلا ORG $ + 100 تخبر المجمّع لزيادة قيمة عداد المواقع بمقدار 100 من قيمته الحالية.

6- التوجيهان PROC و ENDP

يستخدم التوجيه PROC لتعريف بداية إجراء Procedure وهو يلي الاسم الذي يعطى للإجراء. يستخدم بعد التوجيه PROC الحد NEAR أو FAR لتحديد نوع الإجراء. فمثلا Divide PROC FAR يخبر المجمّع بأن الإجراء Divide من النوع البعيد Far (أي في مقطع آخر غير المقطع الموجود فيه). يستخدم مع التوجيه PROC دوماً التوجيه ENDP للإشارة إلى نهاية الإجراء وحصر الإجراء بكل تعليماته بين PROC و ENDP وكأنهما قوسين.

7- التوجيه PUBLIC عام

يتم في البرامج الضخمة المكتوبة على شكل عدة وحدات مستقلة ومنفصلة تجميع كل وحدة واختبارها بشكل إفرادي وبعد ذلك تربط هذه الوحدات مع بعضها لتشكل برنامجاً كاملا. ولكي يتم ربط هذه الوحدات معاً بشكل صحيح فإن اسم أي متحول مشار إليه في واحدات أخرى، يجب التصريح عنه تصريحاً عاماً Public في الوحدة، حيث يكون معرّفاً. فمثلا التصريح PUBLIC X ,Y سيجعل X,Y متاحين من قبل وحدات أخرى

8- التوجيه GROUP مجموعة

يمكن لاي برنامج أن يحتوي على عدة مقاطع من نفس النوع (شفرة تعليمات، معطيات، مكدس) والتوجيه GROUP يؤمن عملية ربط هذه المقاطع تحت اسم واحد لجعلها متجاورة على شكل مقطع واحد.

9- التوجيه MODEL نموذج

ينشيء هذا التوجيه المقاطع ضمنيا مع عبارتي ASSUME و GROUP المطلوبة. ويملك هذا التوجيه الصيغة العامة MODEL memory-model. حيث إن memory-model هو أحد نماذج الذاكرة التالية:

SMALL	جميع المعطيات في مقطع واحد، وجميع التعليمات في مقطع واحد
MEDIUM	جميع المعطيات في مقطع واحد، ولكن شفرة التعليمات يمكن أن تكون في أكثر من مقطع واحد
LARGE	كلا المعطيات والتعليمات في أكثر من مقطع، ولكن التعليمات في مقطع واحد
COMPACT	جميع المعطيات يمكن أن تكون في أكثر من مقطع، ولكن التعليمات ضمن مقطع واحد

١٠- التوجيه LABEL لافتة

عندما يجمّع المجمع قسما من تصريحاتالمعطيات أو عبارات التعليمات، فإنه يستخدم عداد المواقع Location Counter لمعرفة عدد البايتات الموجودة اعتبارا من بداية القطعة في أي وقت. يستخدم التوجيه LABEL لاعطاء اسم للقيمة الحالية في عداد المواقع ويجب أن يتبع LABEL بالحد الذي يعين النوع الذي يراد ربطه بذلك الاسم.

١١- العامل TYPE

يخبّر العامل TYPE المجمع بتحديد النوع لمتحول محدّد. ويحدد المجمع عمليا عدد البايتات في نوع المتحول.

12- العامل OFFSET إزاحة

يخبّر التوجيه OFFSET المجمّع بالإزاحة أو الابتعاد Displacement لبند (أو متحول) معطيات مسمى اعتباراً من بداية القطعة التي تحتويه. ويستخدم هذا العامل عادة لتحميل إزاحة المتحول في مسجل، بحيث يمكن الولوج إلى هذا المتحول. فمثلا عندما يقرأ المجمّع العبارة: MOV BX , OFFSET Prices ، فانه سيحّدد إزاحة المتحول Prices اعتبارا من بداية القطعة المعّرف فيها Prices ويشفر هذا الابتعاد كجزء من التعليمة. وعند تنفيذ التعليمة سيحمل هذا الابتعاد المحسوب في BX

13- العامل PTR مؤشر

يستخدم هذا العامل لتعيين نوع للمتحول أو اللافتة، وهناك ضرورة لعمل ذلك في أي تعليمة عندما يكون نوع المعامل غير واضح. فمثلا عندما يقرأ المجمّع التعليمة INC [BX] يصادف غموض، لأنه من غير المعروف ما الذي يجب إضافته. هل المطلوب زيادة البايت المشار اليه بوساطة BX أم زيادة الكلمة المشار لها بنفس العنوان؟ يصحح ذلك باستخدام المؤشر PTR.

فمثلا التعليمة INC BYTE PTR[BX] تخبّر المجمّع بأننا نريد زيادة البايت المشار إليه بـ BX ، أما INC WORD PTR[BX] فتخبّر المجمّع بأننا نريد زيادة الكلمة.

14- التوجيه END نهاية البرنامج

يوجد هذا التوجيه بعد العبارة أو التعليمة الأخيرة البرنامج وذلك لإعلام المجمّع بأن هذه العبارة هي نهاية الوحدة المتكاملة للبرنامج وهذا يعني أن المجمّع سيتجاهل أية عبارة تأتي بعد END، بالتالي يجب عدم استخدام أكثر منEND واحدة في كل برنامج

يوجد لدى المجمع MASM بعض العوامل والتوجيهات الأخرى الأقل استخداما ومن هذه العوامل نذكر SHIFT, WIDTH, MASK, SIZE, LENGTH,..

ومن التوجيهات: IF, ELSE, ERR, LIST, INCLUDE, COMMENT, PAGE, EXTRN, CONST, RECORD وغيرها.

١٥-٤ تعليمات المعالج 8086

يملك المعالج 8086 مايقارب 117 تعليمة مختلفة بـ 300 Op Code وأن تعليمات المعالج Instruction Set قد تكون بدون معامل Operand أو بمعامل واحد أو بمعاملين. وباستثناء تعليمات السلاسل التي تتضمن عمليات على الحقول Arrays فإن بقية تعليمات المعالج لا تسمح بأن يكون معاملان في التعليمة الواحدة من نوع ذاكرة. يمكن تصنيف مجمل هذه التعليمات إلى ست مجموعات أساسية، وأن لبعض هذه المجموعات الأساسية تصنيفات جزئية.

بالطبع إن لبعض هذه التعليمات تراكيب متماثلة بالتنفيذ، أي لها نفس التركيب القواعدي Syntax، ولكنها تختلف عن بعضها برمز التعليمة Op Code. نعرض فيما يلي مجمل تصنيفات التعليمات للمعالج 8086 على أن يتم في الفصول اللاحقة تعريف مبسط لأكثر التعليمات استخداما، وهذه المجموعات هي:

1- تعليمات نقل المعطيات Data Transfer Instructions

وتضم أربع مجموعات جزئية هي:

- تعليمات النقل ذات الأغراض العامة لبايت أو كلمة، وتضم التعليمات

MOV: نسخ بايت أو كلمة من مصدر محدد إلى وجهة محددة

XCHG: تبادل بايتات أو كلمات

XLAT: نقل مكافيء بايت في المسجل AL باستخدام جدول موجود في الذاكرة

PUSH: نسخ كلمة محددة إلى قمة المكدس

POP: نسخ كلمة من قمة المكدس إلى موقع محدد

PUSHA: نسخ محتويات كل المسجلات إلى المكدس

POPA: نسخ كلمة من المكدس إلى كل المسجلات

(التعليمتان: PUSHA, POPA خاصتان بالمعالجين 80186/80188).

- تعليمات النقل البسيطة الخاصة بمنافذ الدخل والخرج، وتضم التعليمات

IN: نسخ بايت أو كلمة من منفذ محدد إلى المراكم

OUT: نسخ بايت أو كلمة من المراكم إلى منفذ محدد

- تعليمات نقل العناوين الخاصة

LEA: تحميل العنوان الفعّال للمعامل في مسجل محدد

LDS: تحميل مسجل القطعة DS ومسجل محدد آخر من الذاكرة

LES: تحميل مسجل القطعة ES ومسجل محدد آخر من الذاكرة

- تعليمات نقل الأعلام، وتضم التعليمات

LAHE: تحميل (نسخ إلى) AH بالبايت السفلي لمسجل الأعلام

SAHF: تخزين (نسخ) محتويات AH في البايت السفلي لمسجل الأعلام

PUSHF: نسخ محتويات مسجل الأعلام إلى قمة المكدس

POPF: نسخ كلمة من قمة المكدس إلى مسجل الأعلام.

2- التعليمات الحسابية Arithmetic Instructions

وتضم أربع مجموعات جزئية هي:

- تعليمات الجمع

ADD: جمع بايت محدد مع بايت آخر أو كلمة محددة مع كلمة أخرى

ADC: جمع بايت (كلمة) مع بايت (كلمة) آخر مع مسجل الحمل

INC: زيادة بايت محدد أو كلمة محددة بمقدار 1

AAA: ضبط الشيفرة ASCII بعد الجمع

DAA: الضبط العشري BCD بعد الجمع

- تعليمات الطرح

SUB: طرح بايت محدد من بايت آخر أو كلمة محددة من كلمة أخرى

SBB: طرح بايت والحمل من بايت آخر أو كلمة والحمل من كلمة أخرى

DEC: إنقاص بايت محدد أو كلمة محددة بمقدار 1

NEG: نفي أو عكس كل بت لبايت أو كلمة محددة وإضافة 1

(صيغة المتمم الثنائي أو عكس الإشارة)

CMP: مقارنة بايتين أو كلمتين محددتين

AAS: ضبط الشيفرة ASCII بعد عملية الطرح

DAS: الضبط العشري BCD بعد الطرح

- تعليمات الضرب

MUL: ضرب بايت أو كلمة غير مؤشرة ببايت أو بكلمة غير مؤشرة

IMUL: ضرب بايت مؤشر ببايت آخر أو ضرب كلمة مؤشرة بأخرى

AAM: ضبط الشيفرة ASCII بعد عملية الضرب

- تعليمات القسمة

DIV: تقسيم كلمة غير مؤشرة على بايت أو تقسيم كلمة مضاعفة على كلمة مكونة من 2 Bytes.

IDIV: تقسيم كلمة مؤشرة على بايت أو تقسيم كلمة مضاعفة مؤشرة على كلمة مكونة من 2 Bytes.

AAD: ضبط الشيفرة ASCII بعد عملية القسمة

CBW: ملء البايت العلوي للكلمة بنسخ عن بت الإشارة للبايت السفلي

CWD: ملء الكلمة العليا للكلمة المضاعفة ببت الإشارة للكلمة السفلى

3- تعليمات نقل تنفيذ البرنامج Program Execution Transfer Instructions

وتضم خمس مجموعات جزئية تستخدم بمجملها لإخبار المعالج 8086 في البدء بجلب التعليمات من أحد العناوين الجديدة بدلا من الاستمرار في التسلسل وهي:

- تعليمات النقل غير الشرطية

CALL: استدعاء برنامج فرعي(إجراء) وحفظ عنوان العودة على المكدس

RET: العودة من الإجراء إلى البرنامج المستدعي

JMP: الذهاب (القفز) إلى عنوان محدد للحصول على التعليمة التالية.

- تعليمات النقل الشرطية

تستخدم هذه التعليمات غالبا بعد تعليمة مقارنة، ويستخدم الرمز " / " لفصل المختصرات التي تمثل نفس التعليمة، كما يشير الحرف A من " أعلى Above" و B من " أدنى Below" إلى أعداد ثنائية غير مؤشرة بينما الحرف G من " أكبر من Greater than" و L من " أصغر من Less than" للإشارة إلى أعداد ثنائية مؤشرة والتعليمات هي:

JA/JNBE: القفز إذا كان أعلى/ القفز إذا كان ليس أدنى ولا يساوي

JAE/JNB: القفز إذا كان أعلى أو يساوي/ القفز إذا كان ليس أدنى

JB/JNAE: القفز إذا كان أدنى/ القفز إذا لم يكن أعلى ولا يساوي

JBE/JNA: القفز إذا كان أدنى أو يساوي/ القفز إذا كان ليس أعلى

JE/JZ: القفز إذا كان يساوي/ القفز إذا كان مؤشر الصفرية ZF = 1

JG/JNLE: القفز إذا كان أكبر من/ القفز إذا لم يكن أقل من ولا يساوي

JGE/JNL: القفز إذا كان أكبر من أو يساوي/ القفز إذا لم يكن أقل من

JL/JNGE: القفز إذا كان أقل من/ القفز إذا كان ليس أكبر ولا يساوي

JLE/JNG: القفز إذا كان أقل من أو يساوي/ القفز إذا لم يكن أكبر من

JNE/JNZ: القفز إذا كان لا يساوي/ القفز إذا كان لا يساوي الصفر

JC: القفز إذا كان مؤشر الحمل CF = 1 (يوجد حمل)

JNC: القفز إذا كان مؤشر الحمل CF = 0 (لايوجد حمل)

JO: القفز إذا كان مؤشر الفيضان يساوي الواحد (يوجد طفح)

JNO: القفز إذا كان مؤشر الفيضان يساوي الصفر (لايوجد صفح)

JS: القفز إذا كان مؤشر الإشارة يساوي الواحد (العدد سالب)

JNS: القفز إذا كان مؤشر الإشارة يساوي الصفر (العدد موجب)

JP/JPE: القفز إذا كان هناك تكافؤ Parity / القفز إذا كان التكافؤ زوجيا (PF = 1)

JNP/JPO: القفز إذا لم يوجد تكافؤ Parity / القفز إذا كان التكافؤ فرديا (PF = 0)

- تعليمات التحكم التكرارية

LOOP: حلقة تكرارية لتتابع من التعليمات تنفذ حتى يصبح CX = 0

LOOPE/LOOPZ: حلقة تكرارية لتتابع من التعليمات تنفذ طالما أن مؤشر الصفرية ZF = 1 و CX ≠ 0

LOOPNE/LOOPNZ: حلقة تكرارية لتتابع من التعليمات تنفذ طالما أن مؤشر الصفرية ZF = 0 و CX ≠ 0

JCXZ: القفز إلى عنوان محدد إذا كان CX = 0

- تعليمات المقاطعة

INT: مقاطعة تنفيذ البرنامج واستدعاء إجراء الخدمة

INTO: مقاطعة تنفيذ البرنامج إذا كان مؤشر الفيضان يساوي الواحد

IRET: العودة من إجراء خدمة المقاطعة إلى البرنامج الرئيسي

- تعليمات ربط لغة عالية المستوى

ENTER: إدخال (خاص بالمعالجين 80186/8188)

LEAVE: مغادرة (ترك) (خاص بالمعالجين 80186/8188)

BOUND: الاختبار إذا كان العنوان الفعال ضمن حدود المصفوفة المخصصة (خاص بالمعالجين 80186/8188)

4- تعليمات معالجة البتات Bit Manipulation Instructions

وتضم ثلاث مجموعات جزئية هي:

- التعليمات المنطقية

NOT: عكس كل بتات البايت أو الكلمة (المتمم الأحادي)

AND: تنفذ عملية الضرب المنطقي لكل بت من بتات البايت او الكلمة مع البت المقابل له في البايت أو الكلمة الأخرى

OR: تنفذ عملية الجمع المنطقي لكل بت من بتات البايت او الكلمة مع البت المقابل له في البايت أو الكلمة الأخرى

XOR: تنفذ عملية XOR لكل بت من بتات البايت او الكلمة مع البت المقابل له في البايت أو الكلمة الأخرى

TEST: عملية AND على المعاملات لتحديث محتوى الأعلام (المؤشرات) دون تغيير قيم المعاملات

- تعليمات الإزاحة

SHL/SAL: إزاحة بتات البايت أو الكلمة إلى اليسار ووضع أصفار في البايت الأقل أهمية LSB

SHR: إزاحة بتات البايت أو الكلمة إلى اليمين ووضع أصفار في البايت الأكثر أهمية MSB

SAR: إزاحة بتات البايت أو الكلمة إلى اليمين ونسخ البت القديم MSB في البت الجديد MSB

- تعليمات التدوير

ROL: تدوير بتات البايت أو الكلمة إلى اليسار ونقل MSB إلى LSB وإلى CF

ROR: تدوير بتات البايت أو الكلمة إلى اليمين ونقل LSB إلى MSB وإلى CF

RCL: تدوير بتات البايت أو الكلمة إلى اليسار ونقل MSB إلى CF و CF إلى LSB

RCR: تدوير بتات البايت أو الكلمة إلى اليمين ونقل LSB إلى CF و CF إلى MSB.

5- تعليمات السلاسل String Instructions

وتضم عشر تعليمات، حيث يستخدم الرمز " /" لفصل المختصرات المتماثلة، كما يستخدم الحرف B في المختصر ليحدد أن العمل يتم في السلسلة على البايتات والحرف W ليشير إلى أن العمل الجاري يتم في السلسلة على الكلمات، علما أن السلسلة String هي عبارة عن تسلسل من البايتات أو تسلسل من الكلمات المخزنة في مواقع ذاكرية متتالية وتتألف السلسلة من شفرات محارف ASCII، وهذه التعليمات هي:

REP: بادئة Prefix تستخدم لتكرار التعليمة اللاحقة لها حتى تصبح قيمة CX صفرا

REPE/REPZ: بادئة تستخدم لتكرار التعليمة حتى تصبح قيمة CX=0 أو ZF≠1

REPNE/REPNZ: بادئة تستخدم لتكرار التعليمة حتى يصبح CX= 0 أو ZF=1

MOVS/MOVSB/MOVSW: نقل بايت أو كلمة من سلسلة أخرى.

COMPS/COMPSB/COMPSW: مقارنة بايتي سلسلة أو كلمتي سلسلة.

INS/INSB/INSW: إدخال بايت أو كلمة سلسلة من منفذ ما

(خاصة بالمعالجين 80186/80188).

OUTS/OUTSB/OUTSW: إخراج بايت أو كلمة سلسلة إلى منفذ ما

(خاصة بالمعالجين 80186/80188).

SCAS/SCASB/SCASW: مسح سلسلة، مقارنة بايت سلسلة مع بايت في المسجل AL أو كلمة سلسلة مع كلمة في المسجل AX

LODS/LODSB/LODSW: تحميل بايت سلسلة في المسجل AL أو كلمة في AX

STOS/STOSB/STOSW: تخزين بايت من المسجل AL أو كلمة من AX.

6- تعليمات التحكم بالمعالج Processor Control Instructions

وتضم ثلاث مجموعات جزئية هي:

- تعليمة عدم التشغيل

NOP: لا توجد أعمال باستثناء الجلب وتفكيك الشفرة

- تعليمات رفع/ تصفير المؤشرات

STC: رفع مؤشر الحمل (CF = 1)

CLC: تصفير مؤشر الحمل (CF = 0)

CMC: إتمام حالة مؤشر الحمل

STD: رفع مؤشر الاتجاه (DF = 1)، أي إنقاص مؤشرات السلسلة

CLD: تصفير مؤشر الاتجاه (DF = 0)

STI: رفع مؤشر تمكين المقاطعة (تمكين المدخل INTR)

CLI: تصفير مؤشر تمكين المقاطعة (عدم تمكين المدخل INTR)

- تعليمات تزامن المكونات الصلبة الخارجية

HLT: التوقف عن العمل حتى تتم المقاطعة أو التصفير RESET

WAIT: الانتظار حتى تصبح الإشارة على المأخذ TEST منخفضة

ESC: الهروب إلى معالج مساعد خارجي مثل المعالج 8087

LOCK: بادئة تعليمة. تمنع معالج آخر من السيطرة على الممر عندما تكون التعليمة المجاورة قيد التنفيذ.

١٥-٥ اسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

١- برنامج التحويل المستخدم بلغة التجميع Assembly Language يدعى:

a. Compiler b. Interpreter

c. Assembler d. Translator

٢- اللغة التي يمكن أن يفهمها الحاسوب وينفذها تدعى:

a. operating system b. machine language

c. system program d. application software

٣- البرنامج بعد التحويل والتجرمة إلى لغة الآلة يدعى:

a. a source program b. an object program

c. an intermediate program d. a high – level program

٤- باستخدام لغة التجميع Assembly Language يمكننا أن نستخدم معطيات بصيغة

a. Decimal b. binary

c. hexadecimal d. all the above

٥- عدد طرق تحديد المصدر Source في تعليمة من الشكل MOV AX, Source من تعليمات المعالج 8086 هو

a. 8 b. 24 c. 32 d. 64

٦- لتوضع (تخزين) 6 Bytes بمواضع متتابع بالذاكرة دفعة واحدة، تستخدم التعليمة الزائفة pseudo-ops

a. DF b. DD c. DW d. DB 6Bytes

٧- في أي من التعليمات التالية يصادف المجمع Assembler غموضاً

a. INC BYTE PTR[BX] b. INC [BX]

c. INC WORD PTR[BX] d. all the above

الفصل السادس عشر: توصيف تعليمات نقل المعطيات والتعليمات الحسابية والمنطقية للمعالج 8086

نعرض بهذا الفصل تعليمات من ثلاث مجموعات من مجموعات التعليمات الستة العائدة للمعالج 8086 ونركز على توصيف التعليمات التي يمكن أن نصادفها بكثرة أثناء كتابة برامج بلغة الأسمبلي، وهذه المجوعات هي:

١٦-١ التعليمات الحسابية

يبلغ عدد التعليمات الحسابية لدى المعالج 8086 عشرين تعليمة، مقسمة على أربع مجموعات جزئية، نعرض فيما يلي باختصار مفهوم التعليمات الأساسية منها كالتالي:

أ- تعليمات الجمع: وعددها خمس تعليمات نتعرف منها التعليمات:

- **تعليمة الجمع ADD:** وصيغتها العامة هي ADD destination, source

ونقوم بجمع عدداً ما من مصدر مع عدد من وجهة ما ونضع الناتج في الوجه المحددة. ويمكن أن يكون المصدر عدداً فورياً أو مسجلاً أو موقع ذاكرة، أما الهدف فيمكن أن يكون مسجلا أو موقع ذاكرة فقط، حيث كل ما يتعلق بالمعاملات (Operands of Instruction) من حيث النوع والطول مماثل تماماً لما مر معنا بخصوص تعليمة النقل MOV.

وفي حال الحاجة لجمع Byte مع Word ينسخ الـ Byte إلى موقع كلمة ويعبأ البايت العلوي للكلمة بأصفار قبل الجمع.

تؤثر هذه التعليمة بالمؤشرات: CF, PF, AF, SF, ZF, OF وكمثال عن استخدامها

```
ADD [BX] , AL  ; Add AL to contents of memory at address in BX
```

- **التعليمة ADC:** وصيغتها العامة هي ADC destination, source

ونقرأ Add with Carry أي الجمع مع الحمل وصيغتها العامة مثل صيغة التعليمة ADD بعد إضافة الحمل Carry، أي بتعليمة من الشكل ADC op1 , op2 يضاف لقيمة المعاملين op1, op2 قيمة النقل CF وبعد تنفيذ التعليمة يصبح

op1 = op1 + op2 + CF حيث CF يمكن أن يكون 0 أو 1. بالطبع التراكيب المسموحة لكل من op2 ، op1 مماثل لما ورد بالنسبة التعليمتين MOV, ADD ، أما التركيبة (Memory , Memory) غير مسموحة.

وكمثال حول استخدام التعليمة ADC. بفرض أنه باستخدام مسجلات بطول 8bits فقط، يطلب كتابة التعليمات اللازمة لجمع القيمتين الست عشريتين 3AF7 + 25FB.

الجواب: بما أن الناتج يفترض أن يكون 60F2 ، فإن عملية الجمع باستخدام مسجلات بعرض 8bits يجب أن تتم وفق التالي:

```
  3 A    F 7
+ 2 5    F B
-----------
  5 F    F 2
+ 1
-----------
60 F2 <--
```

الناتج النهائي يكون 60 F2H ، لان 5F + 1= 60H وتكتب التعليمات المقابلة لهذه العمليات كالتالي:

```
MOV AL , F7
MOV BL  , FB
MOV AH , 3A
MOV  CL , 25
ADD   AL , BL
ADC   AH , CL
```

يحفظ الناتج بالمسجلين AH-AL بعد أن تم حفظ النقل من المسجل AL بـ AH.

- **التعليمة INC:** وصيغتها العامة هي INC destination (or INC op) وتعني Increment الزيادة وهي حالة خاصة من تعليمة الجمع. يتم بالتعليمة INC إضافة واحد فقط للمعامل الذي يكون موجود في التعليمة. بعد تنفيذ التعليمة يصبح op= op+1. يمكن أن تكون الوجهة (أو الهدف) مسجل أو موقع ذاكرة. تتأثر بهذه التعليمة كل المؤشرات الحسابية باستثناء المؤشر CF . وهذا يعني أنه إذا تمت زيادة الوجهة المؤلفة من 8 bits بعد القيمة FFH فإن النتيجة ستكون بتات كلها أصفار من الشكل 00.

II- تعليمات الطرح: وعددها سبع تعليمات نتعرف منها على بعض التعليمات:

- التعليمة **SUB**: وصيغتها العامة هي SUB destination, source

وهي من Subtraction وتعني الطرح، حيث كل ما يتعلق بالمعاملات (Operand of Instruction) من حيث النوع والطول مماثل تماماً لما مر معنا بخصوص تعليمة الجمع ADD

```
SUB CX, BX  ; Subtract contents of BX from contents of CX,
            ; leave result in CX.
```

- التعليمة **SBB**: وصيغتها العامة هي SBB destination, source

وتعني Subtraction with Borrow وهي مماثلة للتعليمة ADC بكل ما يتعلق بالمعاملات من حيث النوع والطول. الفارق الوحيد بين التعليمتان هو أنه بدل النقل يتم هنا التعامل مع الاستعارة Borrow.

```
SBB CH , AL ; Subtract AL and CF from CH, Result in CH
```

- التعليمة **DEC**: وصيغتها العامة هي DEC destination

وتعني Decrement، أي التنقيص وهي مماثلة تماماً للتعليمة INC بفارق واحد هو بدل إضافة واحد بالتعليمة INC، يتم طرح واحد بالتعليمة DEC، أي أنها عملية خاصة من الطرح SUB. ومؤشر الاستعارة Borrow لا يتأثر بهذه التعليمة.

- التعليمة **NEG**: وصيغتها العامة هي NEG destination

وتعني النفي Negative (أو القلب Invert) وهي تشكل المتمم الثنائي للمعامل. تستبدل هذه التعليمة العدد الموجود في وجهة محددة بالمتمم الثنائي لهذا العدد (أي إشارة السالب).

بمعنى NEG OP تقابل العملية: OP = 0 - OP أي تغير إشارة بايت أو كلمة. يمكن أن تكون الوجهة مسجلا أو موقع ذاكرة محدد. وهذه التعليمة تعدل كافة المؤشرات الحسابية.

- التعليمة **CMP**: وصيغتها العامة هي CMP Destination, source

وتعني مقارنة بايت (أو كلمة) من المصدر المحدّد مع بايت من الوجهة المحددة في التعليمة ، حيث يمكن أن يكون المصدر عدداً أو مسجلا أو موقع ذاكرة. المهم أن لا يكون المصدر والوجهة موقعين ذاكريين معاً في نفس التعليمة. تتم عملية المقارنة فعلياً بتخيل طرح ما في المصدر (Byte or Word) مما هو موجود في الوجهة دون تغيير قيم كل من الوجهة والمصدر ، حيث يتم رفع Set المؤشرات للدلالة على نتائج المقارنة. تستخدم هذه التعليمة مع تعليمات القفز الشرطي وتؤثر في كافة المؤشرات ، فالتعليمة CMP CX , BX تؤثر على المؤشرات وفق التالي:

```
                    CF ZF SF
IF CX = BX THEN     0  1  0  ; Result of subtraction is 0
IF CX > BX THEN     0  0  0  ; No borrow required, so CF=0
IF CX < BX THEN     1  0  1  ; subtraction required borrow
```

بمعنى إذا كان ZF = 1 فهذا يعني أن القيمتين التي يتم المقارنة بينهما متساويتان، وفي حال أن ZF ≠ 1 يعني أن القيمتين مختلفتان وبالتالي إذا كان CF = 0 فهذا يعني أن المطروح هو أصغر من المطروح منه، وهكذا بالنسبة للحالة الثالثة.

ملاحظة : لمقارنة بايتات أو كلمات المسلسلة يستخدم الرمز CMPS ويضاف رمز 'B' أو 'W' لتحديد معامل المقارنة إن كان Byte أم Word.

III- تعليمات الضرب: وعددها ثلاث تعليمات هي:

- التعليمة **MUL**: وصيغتها العامة هي MUL source

وتعني الضرب Multiply. يستخدم الرمز MUL فقط لضرب القيم غير المؤشرة Unsigned. تتعامل التعليمة MUL بشكل مباشر مع معامل Operand واحد، علماً أن عملية الضرب يلزمها معاملان، المعامل الثاني دوماً في المسجل AL إذا كان طول المصدر الوارد في التعليمة 8 bit أو المسجل AX إذا كان طول المصدر الوارد في التعليمة 16 bit حتى ولو لم يذكر ذلك، وهو شكل من أشكال العنونة الضمنية. يتلخص

عمل هذه التعليمة بأنها تضرب بايتاً غير مؤشر من أحد المصادر ببايت آخر غير مؤشر موجود في المسجل AL، أو تضرب كلمة Word غير مؤشرة من مصدر ما، بكلمة غير مؤشرة موجودة في AX.

وبما أن عملية الضرب تضاعف طول الناتج فإن الناتج سيكون بالحالة الأولى في المسجل AX بطول 16 bit وفي المسجلين DX AX بالحالة الثانية بطول 32bits. يمكن أن يكون المصدر مسجلا أو موقع ذاكرة محدداً. أما الهدف فهو دوماً إما AX وإما DX AX. وإذا كان الناتج بطول 16 bit يوضع البايت الأكثر أهمية للنتيجة (البايت العلوي) في AH، بينما يوضح البايت الأقل أهمية في AL.

- التعليمة **IMUL**: وصيغتها العامة هي IMUL source

وتعني Integer MUL وهي نفس تعليمة MUL مع فارق هو أن هذه التعليمة تستخدم لضرب الأعداد المؤشرة Multiply Signed Number علاوة على ذلك، إذا لم تتطلب قيمة ناتج الضرب كل بتات الوجهة Destination فإنه يتم ملء البتات غير المستخدمة بنسخ مماثلة عن بت الإشارة. وإذا أردنا ضرب Byte مؤشر بـ Word مؤشرة يمكننا، كما سنجد لاحقاً استخدام التعليمة CBW. تبقي التعليمتين MUL و IMUL المؤشرات PF, AF, ZF, SF غير معرفة.

```
IMUL BH    ; Signed byte in AL times signed byte in BH
           ; result: signed word in AX
```

IV- تعليمات القسمة: وعددها خمس تعليمات هي:

- التعليمة **DIV**: وصيغتها العامة DIV Source

وهي اختصار لـ Division لقسمة القيم غير المؤشرة وهي مماثلة لتعليمة الضرب من حيث مواصفات المصدر والوجهة. الفرق الجوهري بين DIV , MUL هو أنه أثناء عملية القسمة لا يسمح بقسمة 8 bit/8 bit أو 16 bit/16bit بل يفترض أن يكون طول البسط ضعف طول المقام. لإجراء عملية قسمة عدد على آخر، يتم وضع المقسوم إذا كان بطول Byte في المسجلين AHAL أي في AX (وإذا كان كلمة يوضع في

المسجلين DXAX) والمقسوم عليه يكون في الحالتين هو معامل التعليمة، ويمكن صياغة ذلك كالتالي:

DIV op ; If op is a Byte, then (AHAL)/ op = AL, obtain the result and AH obtain the remainder.

ونفس الشيء بالنسبة للقسمة على عدد بطول 16 bit، حيث الناتج سيكون في AX والباقي في DX. وإذا أردنا قسمة Byte على آخر يجب أولا وضع Byte المقسوم في AL وتعبئته AH كلها بأصفار. جميع المؤشرات بعد تنفيذ التعليمة DIV غير معرفة. وهنا نصادف حالة خاصة تتمثل بالتالي: إذا جرت محاولة للتقسيم على صفر أو كان ناتج القسمة كبيراً جداً، فإن المعالج 8086 ينفذ مقاطعة من النوع "0".

إذا كان المقسوم عليه لا يقسم المقسوم عدداً صحيحاً Integral number من المرات، فإن ناتج القسمة يبتر Truncated (ينتهي عند قيمة ما) ولن يقرب Not rounded . فمثلا : Let BH = (151)D = (97)H ، AX = (14295)D = (3707)H

فإن DIV BH تعني AX/BH وناتج القسمة Quotient سيكون في AL بالقيمة (94)D = (5E)H والباقي بالقيمة (101)D = (65)H في AH ، وعلى كل حال يكون ناتج القسمة مبتوراً (منتهياً) عند قيمة ما، لان باقي القسمة أكبر من نصف المقسوم عليه.

- التعليمة IDIV: وصيغتها العامة هي IDIV source

وهي اختصار لـ Integer Division، أي لقسمة القيم المؤشرة، بالتالي هي مماثلة لتعليمة DIV. ← DIV BL ; (AHAL)/ BL

; AL obtain the result and AH obtain the remainder.

بعد تنفيذ هذه التعليمة، ستكون إشارة باقي القسمة Remainder هي دوماً من نفس إشارة المقسوم، حيث عند تقسيم قيمة موجبة مثلا في AL أي بت الإشارة 0 يجب عندئذ ملء المسجل AH بأصفار وبالعكس.

- **التعليمة CBW:** وصيغتها العامة هي CBW بدون أي معامل

Convert Signed Byte to Signed Word، وتعني تحويل بايت مؤشر إلى كلمة مؤشرة (أي ملء البايت العلوي للكلمة بنسخ عن بت الإشارة للبايت السفلي). تتعامل هذه التعليمة مع المسجل AL حتى ولو لم يذكر ذلك. وتقوم بنسخ إشارة البايت في المسجل AL إلى كل البتات في المسجل AH ويسمى عندئذ المسجل AH بتوسيع الإشارة Sign Extension للمسجل AL . يجب أن تنفذ التعليمة CBW قبل تقسيم بايت في AL على بايت آخر بالتعليمة IDIV. لا تؤثر هذه التعليمة بالمؤشرات.

- **التعليمة CWD:** وصيغتها العامة هي CWD بدون أي معامل

Convert Signed Word to Signed Double Word وتعني تحويل كلمة مؤشرة إلى كلمة مضاعفة مؤشرة (أي ملء الكلمة العليا للكلمة المضاعفة ببت الإشارة للكلمة السفلى). تعدّ هذه التعليمة مماثلة تماماً للتعليمة CBW بكل شيء باستثناء أنها تحول الكلمة المؤشرة الموجود في المسجل AX إلى كلمة مضاعفة مؤشرة تحفظ بالمسجلين DX AX. وتستخدم هذه التعليمة دوما قبل عملية القسمة.

١٦-١-١ تعليق حول التعليمات الحسابية للمعالج 8086

يمكن أن يعمل المعالج 8086 مع أعداد ثنائية مؤشرة وغير مؤشرة، وعلى أعداد BCD أو أعداد بشيفرة ASCII. وإذا احتجت إلى تنفيذ بعض العمليات الحسابية على المعالج 8086 فهناك بضعة تعليمات إضافية تلزم مع التعليمات الأساسية نذكر منها:

- إذا تم جمع عددين BCD فنحن بحاجة لتعليمة إضافية DAA للضبط العشري
- وإذا تم طرح عددين BCD فنحن بحاجة لتعليمة إضافية DAS للضبط العشري
- أما إذا تم التعامل مع أعداد بشفرة ASCII فنحتاج بحسب نوع العملية إلى واحدة من أربع تعليمات إضافية هي:
  - AAA – لضبط ASCII بعد الجمع
  - AAS – لضبط ASCII بعد الطرح
  - AAM – لضبط ASCII بعد الضرب
  - AAD – لضبط ASCII بعد القسمة.

نكتفي هنا بهذه الإشارة لهذه التعليمات التي يرد كل منها بعد عملية حسابية معينة وأن كل من هذه التعليمات الستة هي بدون معامل.

مثال: اكتب برنامج بلغة الأسمبلي لحساب قيمة X المعطات بالعلاقة الرياضية:

$X = (y + z)*(a - b)/10$ علما أن y, z, a, b ثوابت كل منها 8 bit

البرنامج: بالتعليمات التي مرت معنا حتى الآن يمكننا حساب قيمة X وفق التالي:

```
MOV  AL, y
ADD  AL, z        ; Compute AL = y + z
MOV  BL, a
SUB  BL, b ;      ; Compute BL = a – b
MUL  BL           ; Compute AX = AL*BL
MOV  BL, 10D
IDIV  BL          ; compute X = AX/10
```

٢-١٦ تعليمات نقل المعطيات

يبلغ عدد تعليمات هذه المجموعة لدى المعالج 8086 ست عشرة تعليمة هي:

ا- تعليمات النقل ذات الأغراض العامة: وعددها سبع تعليمات، نعرض منها بالإضافة إلى تعليمة MOV التي سبق وتمت الإشارة لها بالتفصيل، التعليمات التالية:

- **التعليمة XCHG:** وصيغتها العامة هي XCHG Destination, source

تعليمة التبادل XCHG تبادل محتويات مسجل بمحتويات مسجل آخر أو بمحتويات موقع ذاكري، ولا تستطيع تبديل محتويات موقعين ذاكريين مباشرة . يمكن تحديد الموقع الذاكري إما كمصدر أو كوجهة ويجب أن يكون كلا من المصدر والوجهة بايتات أو يجب أن يكونا معاً كلمات . يمكن في هذه التعليمة استخدام مسجلات القطعة CS, SS, DS, ES وهي لا تؤثر بالمؤشرات.

مثال: اظهر من خلال برنامج بالأسمبلي عملية التبديل مابين كلمتين بالموقعين الذاكرين 20H و 30H باستخدام التعليمة XCHG وبدون استخدامها.

١ – بدون استخدام XCHG يمكننا التبديل بين محتوى الموقعين كالتالي:

```
MOV   AX ,   [20H]
MOV   BX ,   [30H]
MOV   [20H] , BX
MOV   [30H],  AX
```

٢– باستخدام XCHG يمكننا التبديل بين محتوى الموقعين كالتالي:

```
MOV AX, [20H]
XCHG AX, [30H]
```

– **التعليمة XLAT:** وصيغتها العامة هي XLAT بدون معامل

وتعني Translate a byte in AL أي ترجمة أو تحويل بايت بوساطة AL . تستبدل هذه التعليمة بايتا موجودا في المسجل AL ببايت آخر في جدول الفحص أو البحث Lookup table الموجود في الذاكرة.

وبالتالي يجب قبل تنفيذ هذه التعليمة وضع جدول البحث المحتوي على قيم الشفرة الجديدة في الذاكرة. ويجب أيضاً تحميل إزاحة عنوان البداية لهذا الجدول في المسجل BX. وبعد ذلك للدلالة على البايت المطلوب في الجدول المذكور، تجمع تعليمة XLAT البايت الموجود في AL مع إزاحة البداية للجدول في BX، ومن ثم تنسخ البايت من العنوان المشار إليه بوساطة (AL + BX) إلى المسجل AL. يمكن استخدام التعليمة XTLAT لتحويل أي شفرة 8 bit أو أقل إلى شيفرة بطول 8 bit أو أقل. وهي لا تؤثر بالموشرات.

– **التعليمة PUSH:** وصيغتها العامة هي PUSH Source

وتسمى تعليمة الدفع إلى المكدس Stack وتستخدم لتخزين المعطيات فيه لكي لا تتأثر هذه المعطيات من قبل إجراء ما، ولذلك تقوم هذه التعليمة مع كل تنفيذ لها بإنقاص مؤشر المكدس SP وبعد ذلك يمكن الولوج إلى هذه المعطيات بوساطة أحد

المسجلات عند الحاجة إليها .

يمكن أن يكون المصدر مسجلا ذا أغراض عامة أو مسجل قطعة أو ذاكرة ، وينبغي تهيئة مسجل القطعة و SP قبل استخدام التعليمة. لا تتأثر المؤشرات بهذه التعليمة.

فمثلا : PUSH BX ; Decrement SP by 2 ,Copy BX to Stack

خاطئة PUSH AL ; illegal, Must PUSH a word

- التعليمة **PUSHA**: وتعني Push All Register On Stack

التعليمة بدون معامل (معاملاتها ضمنية وهي كافة المسجلات AX, CX, DX, BX, SP, BP, S1, DI) وهي حالة خاصة لاستخدام عملية الدفع لدى المعالجين 80186/80188 فقط وتعني دفع كل المسجلات على المكدس، وتدفع المسجلات بالترتيب التالي : AX, CX, DX, BX, SP, BP, S1, DI.

- التعليمة **POP**: وصيغتها العامة هي POP Destination

وتسمى تعليمة الاستخراج وهي تعمل بعكس التعليمة PUSH حيث تنسخ كلمة من المكدس مشار لها بالمسجل SP إلى وجهة محددة في التعليمة ، ويمكن أن تكون الوجهة مسجلا ذا أغراض عامة أو مسجل قطعة أو موقعاً ذاكرياً . بعد أن يتم نسخ الكلمة إلى الوجهة المحددة سيزداد مؤشر المكدس بشكل آلي بمقدار 2 ليشير إلى الكلمة التالية في المكدس، وهي لاتؤثر بالمؤشرات

- التعليمة **POPA**: بدون معامل وتعني Pop all registers from Stack

وهي حالة خاصة لاستخدام تعليمة الاستخراج لدى المعالجين 80186/80188 فقط وتعني استخراج كل المسجلات من المكدس . تعد هذه التعليمة بعكس التعليمة PUSHA وهي تستخرج قيم كافة المسجلات التي خزنت بالتعليمة PUSHA . بعد نسخ القيم المخزنة من المكدس إلى المسجل المناسب تتم زيادة SP بمقدار 2 ويتم استرجاع محتويات المسجلات من المكدس بالترتيب التالي:

AX, CX, DX, BX, SP, BP, S1, D1 ولا تؤثر على المؤشرات.

II- تعليمات نقل العناوين الخاصة: وعددها ثلاث تعليمات، نعرض منها

- التعليمة **LEA:** وصيغتها العامة هي LEA register, source وتعني Load Effective Address أي تحميل العنوان الفعّال، حيث إن هذه التعليمة تحدد إزاحة المتحول أو الموقع الذاكري (المصدر)، وتضع هذه الإزاحة في المسجل ذي الطول 16 bit والمشار إليه كهدف. لا تغير التعليمة LEA المؤشرات

```
LEA  BX, Prices    ; Load BX with offset of prices in DS
LEA  CX, [BX] [DI] ; Load CX with EA = [BX] + [DI]
```

III- تعليمات النقل البسيطة الخاصة بمنافذ الدخل والخرج: وعددها تعليمتان هما

- تعليمة الإدخال **IN:** وصيغتها العامة هي IN Accumulator, Port

تنسخ تعليمة الإدخال IN المعطيات من المنفذ (البوابة Port) إلى المراكم. فإذا تمت قراءة منفذ بطول 8 bit فإن المعطيات ستذهب إلى المسجل AL . وإذا تمت قراءة منفذ بطول 16 bit فإن المعطيات ستذهب إلى المسجل AX ويوجد لها صيغتان هما:

- الإدخال من منفذ ثابت Fixed Port .
- الإدخال من منفذ متغير Variable Port .

فمن أجل المنفذ الثابت يكون عنوان المنفذ المؤلف من 8bits محدداً في التعليمة ، فمثلا:

```
IN AL , 0FAH        ; Input a byte from port 0FAH to AL
```

أما من أجل الإدخال من المنفذ المتغير، يكون عنوان المنفذ موجوداً في المسجل DX. وبما أن DX هو بطول 16bit فإن عنوان المنفذ يمكن أن يكون أي عدد ضمن المجال 0000H وحتى FFFFH. يحمل المسجل DX بعنوان المنفذ قبل تنفيذ تعليمة IN وبالتالي تكون صياغة هذه التعليمة في هذه الحالة على النحو التالي :

```
MOV DX , 0FABCH ; Initialize DX to point port with adr. 0FABCH
IN AL, DX ;Input a byte from 8bit port with address FABCH to AL
```

- تعليمة الإخراج **OUT**: وصيغتها العامة هي OUT Port, (AL or AX)

تنسخ هذه التعليمة Byte من المسجل AL أو Word من AX إلى منفذ محدد، وهي مثل تعليمة الإدخال تتميز بصيغتين ممكنتين هما:

- الإخراج إلى منفذ ثابت ، حيث يعين من أجل هذه الصيغة، عنوان المنفذ من 8bit مباشرةً في التعليمة ويمكن بهذه الصيغة عنونة أي منفذ من 256 منفذاً ممكناً.

```
OUT  5 AH  ,  AL   ; Copy the contents of AL to port 5AH
```

- الإخراج إلى منفذ متغير، وفيه سيتم نسخ محتويات AL أو AX إلى المنفذ عند العنوان الموجود في DX ، حيث يمكن هنا عنونة حتى 65536 منفذ في هذا النمط. ويجب دائماً تحميل المسجل DX بعنوان المنفذ المطلوب قبل استخدام التعليمة OUT وكمثال عن استخدام هذه الصيغة نورد المثال التالي:

```
MOV DX , 0ABC DH    ; Load desired port address in DX .
OUT  DX , AL        ; Copy contents of AL to port with
                    ; Address 0ABCDH
```

IV- تعليمات نقل المؤشرات: وتضم أربع تعليمات هي:

LAHE, SAHF, PUSHF, POPF ليس لأي من هذه التعليمات الأربعة أي معامل، حيث إن التعليمة

LAHE: تستخدم لنسخ البايت السفلي لمسجل المؤشرات بالمسجل AH والتعليمة

SAHF: تعمل بالعكس، أي نسخ المسجل AH إلى البايت السفلي لمسجل المؤشرات.

PUSHF: لنسخ الكلمة في مسجل المؤشرات إلى موقع ذاكراي مشار اليه بـ SP، أما

POPF: فتعمل بعكس التعليمة PUSHF.

١٦-٣ تعليمات معالجة البت

يبلغ عدد تعليمات هذه المجموعة لدى المعالج 8086 اثنتي عشرة تعليمة، مقسمة إلى ثلاث مجموعات جزئية هي:

أ- التعليمات المنطقية: وعددها خمس تعليمات، نعرض منها

- **التعليمة NOT**: وصيغتها العامة هي NOT destination

وتعني عكس كل بت في المعامل (الذي يمكن أن يكون Byte أو كلمة Word). تشكل هذه التعليمة المتمم الأحادي للوجهة التي يمكن أن تكون مسجلا أو موقعاً ذاكرياً، وهي لا تؤثر في المؤشرات. Ex. NOT BYTE PTR [BX]

; Complement memory byte at offset [BX] in DS

- **التعليمة AND**: وصيغتها العامة هي AND destination, source

وتعني AND corresponding bits of two operands أي تنفيذ عملية الضرب المنطقي AND للبتات المتقابلة، وهذا يعني تنفيذ الضرب لكل بت من بتات البايت أو الكلمة مع البت المقابل له في البايت أو الكلمة الأخرى. يتم بعد تنفيذ التعليمة وضع النتيجة في الوجهة المحّددة. ويمكن أن يكون معامل المصدر عدداً فورياً أو محتويات مسجل ما أو قد يكون محتويات موقع ذاكري. ويمكن للوجهة أن تكون مسجلا أو موقعاً ذاكرياً. تصبح المؤشرات CF و OF أصفاراً بعد تنفيذ تعليمة AND ويتم تعديل باقي المؤشرات. ويمكن بهذه التعليمة تقنيع أي بت بضربه منطقياً بالصفر .

AND BH , CL ; AND byte in CL with byte in BH, result in BH.

- **التعليمة OR**: وصيغتها العامة هي OR destination, source

وتمثل الجمع المنطقي للبتات المتقابلة لمعاملي التعليمة. وهي من حيث نوع المعاملات وأطوالها مماثلة تماماً لتعليمة AND باستثناء أنه عوضاً عن الضرب المنطقي بالتعليمة AND يتم الجمع المنطقي بالتعليمة OR.

مثال: استخدم عمليات AND و NOT لتحل محل التعليمة OR.

```
NOT  AL
NOT  BL
AND  AL, BL
NOT  AL
```

وهي تحاكي قانون دي مورغان: $(a + b)' = a'.b'$

- التعليمة **XOR**: وصيغتها العامة هي AND destination, source

وهي أيضاً مماثلة لتعليمتي AND و OR من حيث نوع المعاملات وطولها. إذا تمت عملية XOR لبت معين مع الواحد فإنه يتم نفي هذا البت، أما إذا جرى مع الصفر فسوف لن يتغير ذلك البت، بالتالي يمكن استخدام التعليمة XOR للعكس الاختياري أو عدم العكس لبيتات المعامل.

- التعليمة **TEST**: وصيغتها العامة هي TEST destination, source

وتستخدم هذه التعليمة لضرب المعاملات المنطقية لتعديل المؤشرات، أي تقوم بالضرب المنطقي لمحتويات بايت أو كلمة المصدر بمحتويات كلمة الوجهة المحددة، وتعدل المؤشرات ولكن لايتغير أي معامل.

تستخدم تعليمة TEST غالبا لرفع Set المؤشرات قبل تعليمة قفز شرطي. وهي مماثلة للتعليمة AND من حيث نوع وطول المعاملات. ويصبح المؤشران CF و OF صفرا بعد TEST. وتعدل المؤشرات ZF, SF, PF لإظهار نتائج الضرب المنطقي AND وسيكون المؤشر AF غير معرف. ويصبح للمؤشر PF معنى فقط من أجل البتات الثمانية السفلى للوجهة، فمثلا:

Examples (Coding)

```
TEST AL, BH  ; AND BH with AL, no result. Update PF, SF, ZF.
TEST BP, [BX] [DI] ; AND word at offset [BX] [DI] in data
                   ; segment with word in BP, no result stored.
                   ; Update PF, SF, and ZF
```

Examples (Numerical)

```
      ; AL = 0101 0001
TEST AL, 80H  ; AND immediate 80H with AL to test.
              ; AL = 0101 0001 Unchanged, PF = 0, SF = 0,
              ; ZF = 1 because ANDing produced 00
```

وبعد أن تعرفنا بعض التعليمات المعالج 8086 (من مجموعة التعليمات الحسابية والمنطقية ومجموعة تعليمات نقل المعطيات) وبعض التوجيهات يمكننا المباشرة بعرض أمثلة بسيطة باستخدام هذه التعليمات وفق التالي:

مثال: حول إيجاد متوسط درجتي حرارة.

سنحاول بهذا المثال البسيط إظهار الشكل العام للبرنامج المقابل بلغة الأسمبلي، بفرض أن المعطيات عن درجتي الحرارة الدنيا LO-temp والعليا HI-temp موجودة في الذاكرة. فأول ما يجب فعله هو تجهيز بنية المعطيات باستخدام التوجيهين DATA-Here Segment و DATA-Here ENDS حيث يمكن التصريح في القطعة المسماة Data-Here عن المتحول HI-Temp على أنه متحول من النوع Byte type ويهيأ بالقيمة 50 مثلا. وعن المتحول LO-Temp أيضاً ويهيأ بالقيمة 10 مثلا. ويحجز موقع ذاكري بالاسم AV-Temp للنتيجة ليخزن به البايت المقابل للمتوسط الحسابي بقيمة غير معروفة، فيكون البرنامج على النحو التالي:

```
DATA-Here Segment
        HI-temp   DB   50H          ; max temp storage
        LO-temp   DB   10H
        AV-temp   DB   ?            ; put average here
DATA-Here   ENDS
CODE-Here  Segment
                ASSUME CS: CODE-Here , DS: DATA-Here
                MOV  AX , DATA-Here   ; initialize data segment
                MOV  DS , AX
                MOV  AL , HI-temp          ; get first temperature
                ADD   AL , LO-temp
                MOV  AH , 00H
```

```
                ADC   AH , 00H     ; put carry in LSB of AH
                MOV   BL , 02H
                DIV   BL          ; divide AX by BL
                MOV   AV--temp , AL
CODE-Here       ENDS
                END
```

مثال: اكتب برنامجاً بلغة الأسمبلي لضرب عددين كل منهما بطول 16 Bits.

الجواب: بأسلوب مماثل للمثال السابق، يمكننا أن نبدأ بتجهيز مقطع المعطيات ونعطيه اسماً ما. ويمكننا إضافة أي شروحات توضح عمل البرنامج بعد الفاصلة المنقوطة.

ملاحظة: سنضطر في هذا المثال استخدام تعليمة جديدة لم نعرفها بعد، وهي التعليمة INT 3H (كما سنجد لاحقا) التي تجبر المعالج على التوقف عن تنفيذ التعليمات وانتظار المستخدم لكي يملي على المعالج خطوة العمل التالية.

```
DATA-Here  SEGMENT
           Multiplicand  DW   204AH  ; First Word Here
           Multiplier    DW   3B2AH  ; Second Word Here
           Product       DW   2  DUP ( ? ) ; Result Here
DATA-Here  ENDS
CODE-Here SEGMENT
           ASSUME  CS : CODE-Here , DS : DATA-Here
Start:     MOV   AX , DATA-Here   ; Initialize DS Register
           MOV   DS , AX
           MOV   AX , Multiplicand
           MUL   Multiplier
; the result (product) is 32 bit stored in registers AX and DX
```

```
        MOV   Product , AX
        MOV   Product + 2 , DX
        INT     3H
CODE-Here ENDS
        END     Start
```

كي يشحن البرنامج وينفذ يحتاج فقط لـ END directive followed by a CR

II- تعليمات الإزاحة: وعددها ثلاث تعليمات هي:

- **التعليمة SHL:** وصيغتها العامة SHL Destination, Count

وتعني Shift operand bits left, put zero in LSB أي إزاحة بتات المعامل إلى اليسار ووضع صفر في البت (البتات) LSB. يوجد رمز آخر للتعليمة هو SAL (يماثل SHL). الرمزان اختصار لنفس التعليمة، وهي تزيح كل بيت في الوجهة المحددة عددا من المواضع إلى اليسار . عندما يزاح البت من الموضع LSB سيتم وضع الصفر بهذا الموضع ويزاح البت MSB إلى مؤشر النقل CF. يمكن أن يكون معامل الوجهة بايتا أو كلمة، ويمكن أن يوجد في مسجل أو موقع ذاكري. إذا كان عدد الإزاحات المطلوبة واحداً، فإن ذلك يمكن تحديده بوضع 1 في موضع العد للتعليمة، وإذا كان عدد الإزاحات أكثر، فيحمّل العدد في المسجل CL. يمكن استخدام التعليمة SHL أو SAL أيضاً لضرب عدد ثنائي غير مؤشر بقوى 2.

- **التعليمة SHR:** وصيغتها العامة SHR Destination, Count

وتعني Shift operand bits right, put zero in MSB وهي مماثلة تماما للتعليمة SHL بخلاف واحد هو أن الإزاحة تكون نحو اليمين مع مراعاة المفاهيم المرتبطة بكل نوع من الإزاحة، كالقسمة مثلا بدل الضرب.

III- تعليمات التدوير: وعددها أربع تعليمات هي:

- **التعليمة ROL**: وصيغتها العامة ROL Destination, Count

وتعني Rotate all bits of operand left, MSB in LSB أي تدوير كل بتات المعامل إلى اليسار من MSB إلى LSB. تدور هذه التعليمة كل البتات في بايت أو كلمة معينة عددا من المواضع إلى اليسار، بحيث إن بت المعطيات الدائر من MSB يعود دائرا إلى LSB وينسخ أيضاً إلى مؤشر الحمل CF أثناء ROL. يمكن استخدام التعليمة ROL لمقايضة Swap النيبلات في بايت ، أو البايتات في كلمة. كما يمكن استخدام هذه التعليمة أيضاً لتدوير بت معين إلى مؤشر النقل لاختباره

- **التعليمة ROR**: وصيغتها العامة ROR Destination, Count

وتعني Rotate all bits of operand right, LSB to MSB أي تدوير كل بتات المعامل إلى اليمين من LSB إلى MSB وهي تدور كل البتات في بايت أو كلمة معينة عددا من المواضع إلى اليمين، وتوصف العملية بأنها دوران أكثر منها إزاحة، وذلك لان البت المتحرك من الموضع LSB يدور إلى الموضع MSB. حيث ينقل بت المعطيات الخارج من LSB إلى CF أثناء تعليمة ROR.

التعليمتان ROR, ROL من حيث المعاملات مماثلة تماما لمختلف تعليمات الدوران والإزاحة (وفق ما هو مبين بالنسبة للتعليمة SHL).

والتعليمتان RCL و RCR مماثلة للتعليمتين ROL و ROR بخلاف أن تدوير المعامل يتم عبر مؤشر النقل، إما لليمين وإما اليسار.

الفصل السابع عشر: توصيف تعليمات نقل البرنامج والتحكم بالمعالج 8086

بداية نشير إلى بعض الملاحظات المتفرقة حول البرمجة بلغة الأسمبلي التي نصادفها أثناء كتابة بعض البرامج ومنها:

1- الغموض وإزالته Removing ambiguity

نصادف بلغة Assembly بعض أنواع الغموض التي يجب تجاوزها، علماً أن بعض نسخ المجمّعات الحديثة تتجاوز مثل هذه الإشكالات. ومن أشكال الغموض

*- عند كتابة تعليمة بالشكل MOV AX , 26 يحدث غموض يتمثل بعدم معرفة نوع المصدر 26 هل هو عدد عشري أم ثماني أم ستة عشري؟، بالتالي لإزالة الغموض لابد من إضافة رمز D أوO أوH لتحديد نوع المصدر، كأن نكتب: MOV AX , 26 H

*- وعند كتابة تعليمة بالشكل MOV DL , AH يحدث غموض أيضاً ، يتمثل بعدم معرفة الـ Assembler نوع المصدر هنا وهو AH هل هو مسجل أم عدد ستة عشري؟ بالتالي لإزالة الغموض لابد من إضافة صفر أمام المعامل إذا كان عدداً بحيث يصبح مفهوم التعليمة MOV DL , 0AH انقل القيمة الستة عشرية A إلى المسجل DL . وأما MOV DL , AH فتعني انقل محتوى المسجل AH إلى المسجل DL.

2- تغيير محتويات المسجلات: ويقصد بذلك إن تعديل قيمة مسجل بطول 16 bit يؤثر في أطوال المسجلات المكونة له بطول 8bit وفق ورود استخدام هذه المسجلات بالبرنامج، فمثلا بالتعليمات:

```
MOV  DH , 98 H
MOV  DX ,23ABH
ADD  DH , 1
```

بعد تنفيذ آخر تعليمة، يكون محتوى المسجل DH = 24H وليس 99H ، لان قيمة DH من المرحلة قبل الاخيرة أصبح 23H عوضاً عن 98H .

3 - إذا كان المتحول Message يقابل العنوان 0200 : 102A (عنوان المقطع: عنوان الإزاحة) فان عبارة MOV BX , OFFSET Message ستجعل في المسجل BX مقدار الإزاحة فقط، وهو في هذا المثال 0200H وتضيفه إلى العنوان الموجود في DS

عند حساب العنوان الفيزيائي بعد إزاحته نحو اليسار.

4- العناوين وأنواعها: نشير إلى أن بعض التعليمات تتعامل مع عناوين، وأن لهذه العناوين خصائص محددة. تطلق ثلاث صفات للعناوين هي:

العناوين القصيرة SHORT: هي تلك العناوين التي يمكن الوصول إليها انطلاقا من الموقع الحالي بوساطة قيمة للإزاحة محدودة بالمجال من128- وحتى 127 بايت، بينما

العناوين القريبة NEAR: يقصد بها العناوين التي يمكن الوصول إليها بوساطة الإزاحة المحددة بالمجال من 32.768 - وحتى 32.767 بايتاً ضمن نفس المقطع الذي نوجد به. وأخيرا

العناوين البعيدة Far: يقصد بها العناوين التي يمكن الوصول إليها بوساطة الإزاحة المحددة بمجال خارج المجال الأخير، أي تلك العناوين التي نصل إليها من خلال عنواني الإزاحة والمقطع معا.

وهنا نشير أيضاً إلى أن معظم الحواسيب تفهم طريقة العنونة من التعليمة، حيث إن أغلب المجمعات تستعمل رموزاً مختلفة لتعريف طريقة العنونة. فمثلا

- الرمز #: يرمز للعنونة الثقافية immediate addressing وهو يشير إلى ثابت أو معامل (يعني وجود عددا مهما كان نظام تمثيله) كأن نكتب العدد الثنائي #1011B

- الرمز @: يرمز للعنونة غير المباشرة indirect addressing، وتستطيع استخدام العبارات المسبوقة بـ @ في العبارات التنفيذية كالتعليمة MOV AX, @DATA

5- إجراء التتبع Making traces

يقصد بعملية التتبع Tracing ملاحقة التغيرات التي تحدث أثناء تنفيذ البرنامج على محتوى المسجلات أو مواقع الذاكرة المستخدمة أو قيم المؤشرات الحسابية بعد تنفيذ كل تعليمة على حدة. تكون قيم المؤشرات أو القيم في المسجلات ومواقع الذاكرة المستخدمة، قبل المباشرة بإجراء تتبع قيم مجهولة (غير معروفة) ويرمز لها بالرمز " ؟ " (Unknown) وتتغير بالتدريج مع التنفيذ المتتابع للبرنامج. وفيما يلي نعرض مثال على

ذلك وليكن بالصيغة التالية: Make trace of arithmetic flags through excuted the sequence of the following assembly stetments.

بمعنى: أجري تتبع للمؤشرات الحسابية خلال تنفيذ سلسلة تعليمات الاسمبلي التالية:

Instruction	C	O	S	Z	P	Comment
	?	?	?	?	?	unknown
MOV BL,3H	?	?	?	?	?	MOV does not affect flag
ADD BL, 2H	0	0	0	0	1	Decimal 5=$(00000101)_2$
SUB BL, 5H	0	0	0	1	1	Since 3 + 2 − 5 = 0
SUB BL, 2H	0	0	1	0	0	Result(−2)is negative
MOV DH, 2H	0	0	1	0	0	MOV does not affect the flags
ADD DH,0FFH	1	0	0	0	0	Carry because 2+FFH=101H
MOV AL, 4H	1	0	0	0	0	MOV does not affect the flags
ADD AL, 7FH	1	1	1	0	0	Signed number overflow

مثال: اكتب برنامجاً يعمل على زيادة محتويات كل من مواقع الذاكرة الثلاثة من 200H وحتى 202H بقيمة واحد فقط. واجر عملية التتبع Tracing للمسجلات ومحتويات مواقع الذاكرة المستخدمة وفق هذا البرنامج، إذا علمت أن محتويات مواقع الذاكرة قبل إقلاع البرنامج كانت: Location: 200H 201H 202H

Contents: 2AH B4H 51H

وبالعودة إلى موضوع تعليمات نقل البرنامج والتحكم بالمعالج 8086، نشير إلى ان التتبع ينتقل خلال تنفيذ برنامج ما من موضع لآخر باستخدام تعليمات نقل البرنامج التي يبلغ عددها ثماني وعشرين تعليمة مقسمة إلى أربع مجموعات جزئية هي:

1- تعليمات النقل غير الشرطية .

2- تعليمات النقل الشرطية .

3- تعليمات المقاطعة .

4- تعليمات التحكم التكرارية .

١٧-١ تعليمات النقل غير الشرطية

وتضم هذه المجموعة ثلاث تعليمات فقط، نعرضها وفق التالي:

- **التعليمة JMP:** وصيغتها العامة هي JMP destination (or Adr.)

وتعني Unconditional jump to specified destination أي القفز غير الشرطي إلى وجهة محددة، وتؤدي إلى جلب التعليمة التالية من الموقع المحدد في التعليمة بدلا من الموقع التالي بعد التعليمة JMP . فمثلا من أجل تعليمة بالشكل: JMP adr، يلاحظ إذا كانت الوجهة destination موجودة في نفس قطعة الشفرة CS فإنه يتم تغيير مؤشر التعليمات IP فقط للحصول على موقع الوجهة. أما إذا كانت الوجهة لتعليمة القفز في قطعة ثانية ذات اسم يختلف عن القطعة التي تحتوي تعليمة JMP ، فإنه يتم تغيير محتويات كل من مؤشر التعليمات IP ومسجل قطعة الشفرة CS للحصول على موقع الوجهة.

- **التعليمة CALL:** وصيغتها العامة هي CALL destination (adr.)

وتعني استدعاء إجراء وتستخدم لنقل التنفيذ إلى برنامج فرعي، حيث يوجد نوعان من الاستدعاءات، هما " الاستدعاءات القريبة والبعيدة NEAR & FAR " تستدعي تعليمة CALL القريبة إجراء متواجد في نفس قطعة الشفرة CS وتستدعي تعليمة CALL البعيدة إجراء متواجد في قطعة شفرة ثانية .

عندما ينفذ المعالج 8086 تعليمة CALL القريبة ، فإنه ينقص مؤشر المكدس SP بمقدار 2 لينسخ إزاحة التعليمة التالية لـ CALL إلى المكدس وتستخدم هذه الإزاحة المخزنة في المكدس للاشارة إلى عنوان العودة Return Address . أما الاستدعاء البعيد فهو استدعاء لاجراء يكون في قطعة مختلفة عن القطعة التي تحتوي التعليمة CALL . عندما

ينفذ المعالج تعليمة من هذا النوع فإنه ينقص SP بمقدار 2 وينسخ محتويات CS إلى المكدس ومن ثم ينقص SP بمقدار 2 مرة ثانية (من جديد) وينسخ إزاحة التعليمة التالية للتعليمة CALL إلى المكدس .

- **التعليمة RET:** وصيغتها العامة هي RET بدون معامل

Return Execution from procedure to calling Program

وتعني عودة التنفيذ إلى البرنامج المستدعي. تعيد تعليمة RET التنفيذ من إجراء ما إلى التعليمة التالية لتعليمة CALL في البرنامج المستدعي . وهنا يجب التميز مابين الإجراء إن كان قريبا Near Procedure أو بعيدا Far Procedure لان ذلك ينعكس على طريقة العودة، إما باستبدال المؤشر IP بكلمة من قمة المكدس وزيادة SP بمقدار 2 عندما يستخرج عنوان العودة من المكدس. وإما باستبدال IP بالكلمة الموجودة عند قمة المكدس، وبعدئذ سيزداد SP بمقدار 2 ثم يستبدل بعد ذلك مسجل قطعة الشفرة بكلمة من القمة الجديدة للمكدس .

٢-١٧ تعليمات النقل الشرطية

تسمح تعليمات القفز المشروطة لعناصر العائلة 8086 أن تتخذ قراراً اعتماداً على المعلومات التي يعرضها مسجل المؤشرات وهذا يعني أن تعليمة Jump أياً كان نوعها يمكن أن تسمح بالاستمرار في تنفيذ البرنامج من تعليمة ما في مكان آخر أبعد بمسافة ما ضمن البرنامج . فمثلا: تعليمة من الشكل JZ xx (وتقرأ Jump zero) يمكنها أن تقطع التنفيذ التتابعي للبرنامج وأن تكمل التنفيذ من مكان آخر معنون باللافتة xx كالتالي: إذا كانت نتيجة التعليمة السابقة للتعليمة JZ xx صفرية اقفز إلى التعليمة باللافتة xx وأكمل التنفيذ من ذلك المكان الجديد في البرنامج.

بالطبع إذا لم يكن ZF = 1 فإن البرنامج يكمل من التعليمة التي تلي التعليمة JZ xx مباشرةً، وبالاطار العام كما يمكن القفز بناءً على قيمة مؤشر ما مثل ZF بالصيغة JZ يمكن القفز على نفي قيمة المؤشر أي ZF = 0 بالصيغة JNZ. ونميز بتعليمات القفز بين القيم المؤشرة وغير المؤشرة، حيث يصاحب بعض تعليمات القفز رمز من الرموز

التالية:

A– من Above وتعني أعلى أو B من Below وتعني أدنى، للاشارة إلى الأعداد الثنائية غير المؤشرة و

G – من Greater Than أكبر من أو L من Less Than أصغر من، للاشارة إلى الأعداد المؤشرة .

وبما أنه يوجد لدى المعالج 8086 ستة مؤشرات حسابية يمكن تشكيل عشر تعليمات بالاعتماد على خمسة من هذه المؤشرات، لان مؤشر النقل المساعد AF لا يقابل أي تعليمة، ويمكننا عرض تعليمات القفز المشروط وفق التالي:

– **التعليمتين JNZ – JZ:** وصيغتهما العامة هي JZ xx أو JNZ yy

وتقرأ Jump if Zero – Jump if Not Zero وتعني،

القفز إذا كان مؤشر الصفرية ZF = 1 أي النتيجة صفرية (JZ) أو القفز إذا كانت النتيجة بعكس الصفر أي ZF = 0 (JNZ).

أي عندما نقول أن القفز سيتم بناء على المؤشر ZF = 1 الذي يعني نتيجة صفرية ، كأننا نقول القفز عند التساوي مع الصفر أي Jump of equal اختصارا JE وبالتالي

– JE / JZ يمثلان اختصارا لنفس التعليمة .

– ويكون JNE – Jump if not equal مماثلة لـ JNZ أي JNE/ JNZ

تعليمات القفز المشروط لا تؤثر في المؤشرات.

– **التعليمتان JNC – JC:** وصيغتهما العامة هي JC xx أو JNC yy

وتعني Jump IF Carry – Jump if no Carry ، أي القفز إذا كان موجوداً نقل CF = 1 أو القفز إذا لم يكن موجوداً نقل CF = 0 .

والتعليمتان تختلفان عن بعضيهما البعض باختلاف قيمة مؤشر النقل، فمثلا:

```
ADD  BX , CX     ; Add two word
JC   Error–Fix   ; Jump to label Error–Fix if CF = 1
```

- التعليمتان JS – JNS: وصيغتهما العامة هي JS xx أو JNS yy

وتقرأ – (–) Jump if Signed أو (+) Jump if Not Singed

وهاتان التعليمتان حول القفز المشروط بوجود إشارة سالبة من عدم وجودها مماثلة لتعليمات القفز حول شرط النقل ، بمعنى أنه لا يوجد مكافئ لهما.

- التعليمتان JO – JNO: وصيغتهما العامة هي JO xx أو JNO yy

وتقرأ Jump if overflow – Jump if no overflow

وهي أيضاً مثل التعليمات الخاصة بالقفز على وجود مؤشر النقل أو الإشارة من عدم وجودهما، لا يقابلها أي من تعليمات القفز للقيم المؤشرة أو غير المؤشرة.

- التعليمتان JPE – JPO: وصيغتهما العامة هي JPE xx أو JPO yy

وتقرأ Jump if Parity Even – Jump if Parity Odd

وتعني القفز إذا كان التكافؤ زوجياً أي PF = 1 أو إذا كان التكافؤ فردياً PF = 0.

مجرد القول ان القفز سيتم بناء على التكافؤ، أي Jump if Parity اختصارا JP، يعني أن القفز سيتم على وجود تكافؤ زوجي وهذا ما يقابل الصيغة JPE لذلك قد يستخدم الاختصاران JP/JPE على أنهما متكافئتان للاشارة إلى نفس التعليمة. ونفس الشيء عند القول بان القفز سيتم في حال عدم وجود Parity ، اختصاراً JNP فهذا يعني أن القفز سيتم بناء على وجود تكافؤ فردي، أي JPO وبالتالي التعليمتان:

- JNP/JPO متماثلتان ويمثلان نفس التعليمة.

بعض هذا العرض المبسط لبعض تعليمات المعالج 8086 يمكننا المباشرة بكتابة بعض البرامج البسيطة. حيث سيتم التركيز على كتابة الـ Code دون عرض التركيبة القياسية للبرنامج بلغة الاسمبلي، أي بالمقاطع الاساسية للبرنامج

مثال: اكتب برنامج بلغة الأسمبلي يقارن بين محتوى المسجلين AX , BX، بحيث إذا كان كلا المسجلين يحتويان نفس القيمة، يضع 1 في المسجل DX وألا يضع 0 في DX، شرط أن لا يتغير محتوى أي من المسجلين AX, BX.

الجواب:

```
        MOV   CX , AX
        SUB   AX , BX
        JZ    xx
        MOV   DX , 0
        JMP   yy
xx:     MOV   DX , 1
yy:     MOV   AX , CX
```

مثال: اكتب برنامج يقوم بنقل 10 قيم بالذاكرة اعتباراً من الموقع 1000H كل منها بطول 1 Byte إلى قائمة ثانية في الذاكرة عنوان آخر موقع فيها 2000H على أن يتم قلب القائمة رأساً على عقب.

الجواب:

```
        MOV  CX, 0AH
        MOV  BX, 1000H
        MOV  DX, 2000H
xx:     MOV  AL, [BX]
        MOV  AH, [DX]
        XCHG AL, AH
        INC  BX
        DEC  DX
        JNZ  xx
```

نكمل عرض بقية تعليمات هذه المجموعة وفق ما هو مبين بالجدول 1-17.

الجدول 1-17 : تعليمات قفز المعالج 8086 المتكافئة

الرمز	معنى التعليمة	المكافئ
JA	JUMP IF ABOVE القفز إذا كان أعلى، ويتم القفز إذا تحقق الشرط التالي (CF or ZF = 0) وتكتب بالشكل JA / JNBE .	JNBE
JAE	JUMP IF ABOVE OR EQUAL القفز إذا كان أعلى أو يساوي ويتم القفز إذا تحقق الشرط التالي CF =0، وتكتب بالشكل JAE / JNB	JNB
JB	JUMP IF BELOW القفز إذا كان أدنى ويتم القفز إذا تحقق الشرط التالي CF = 1 وتكتب بالشكل JB / JNAE.	JNAE
JBE	JUMP IF BELOW or EQUAL القفز إذا كان أدنى أو يساوي ويتم القفز إذا تحقق الشرط CF or ZF = 1 وتكتب بالشكل JBE / JNA	JNA
JG	JUMP IF GREATER THAN القفز إذا كان أكبر من ويتم القفز إذا تحقق الشرط ZF = 0 and SF=OF وتكتب بالشكل JG/JNLE	JNLE
JGE	JUMP IF GREATER THAN or EQUAL القفز إذا كان أكبر من أو يساوي JGE = JNL	JNL
JL	JUMP IF LESS THAN القفز إذا كان أقل من ويتم القفز إذا تحقق SF≠OF	JNGE
JLE	JUMP IF LESS THAN or EQUAL القفز إذا كان أقل من أو يساوي ويتم القفز إذا كان ZF = 1 or SF ≠OF ويكتب بالشكل JLE/ JNG	JNG

مثال: اكتب برنامجاً يقوم بعّد الأحداث (أو القيم) الستة عشرية المعطاة التي تساوي القيمة 3A7H بالمواضع 24FH --200 ضمنا واترك الناتج بالمسجل DX .

الجواب: بداية ; 24F – 200 = 4FH , 4F + 1 = 50H, Thus

```
    MOV   CX , 50H
    MOV   DX , 0H
    MOV   BX , 200H
yy: MOV   AX , [BX]
    SUB   AX , 3A7H
    JNZ   xx
    INC   DX
xx: INC   BX
    INC   BX
    DEC   CX
    JNZ   yy
```

مثال: نحاول في هذا المثال كتابة برنامج من نوع آخر يتعلق بآلية توليد تأخير زمني بمقدار دقيقة مثلا باستخدام المعالج 8086 إذا كان المعالج يعمل بـ 10MHz. الجواب: بشكل عام في التطبيقات التي تعمل بالزمن الحقيقي كالتحكم بالإشارات الضوئية أو الساعة الرقمية وغيرها، من الضروري متابعة مسار الزمن، حيث يوجد تقنيتان برمجيتان لتحقيق التأخير الزمني الذي يلزم في بعض التطبيقات، هما إما باستخدام التعليمة NOP واما باستخدام العدادات. فبالنسبة لتقنية العدادات، تستطيع عملية العّد أن تولد التأخير الزمني المطلوب انطلاقا من أن زمن تنفيذ التعليمات المستخدمة بإجرائية العّد يكون معروفاً. وبالتالي الأساس بهذه التقنية هو معرفة القيمة الأولية التي تعطى للعداد لتحديد مقدار التأخير الزمني وهذه القيمة عادة تحسب وفق خوارزمية محددة. فبالنسبة للمعطيات المعطاة في هذه المسألة، يكون البرنامج على الشكل التالي:

```
        MOV  BX, Multiplier Count
REPE:   MOV  CX, Count  ; 4
BACK:   DEC  CX         ; 2
        JNZ  BACK       ; 16/4
        DEC  BX
        JNZ  REPE
```

ويصبح السهم هو معرفة القيمة Multiplier Count التي توضع بالمسجل BX وقيمة العَد Count التي توضع بالمسجل CX، لأنه بمعرفتهما يمكن تحديد مقدار التأخير الزمني الذي يولده البرنامج.

في البرنامج أعلاه معروف عدد الدورات الساعية اللازمة لتنفيذ كل تعليمة على حده. إذ إن التعليمتان DEC CX و JNZ BACK ستنفذ عدد من المرات يساوي قيمة الـ Count المخزن بـ CX ويمكن حساب هذه القيمة وفق التالي:

يلزم التعليمة JNZ للتنفيذ في حال عدم تحقق الشرط 4 دورات ساعية وفي حال تحقيقه يلزم 16 دورة ساعية، بينما التعليمة MOV CX, Count التي ستنفذ لمرة واحدة يلزمها 4 دورات ساعية. إذا يوجد بالحلقة أعلاه (Count – 1) مرور حتى يتحقق الشرط ويعود التحكم للتعليمة DEC CX، وبالتالي يصبح عدد الدورات الساعية عبر المسجل CX طالما أن قيمته مغايرة للصفر يساوي (Count – 1)x(16 + 2) وفي آخر مرور بالحلقة يلزم (4 + 2) دورة ساعية ويصبح إجمالي الدورات الساعية المطلوب لتنفيذ البرنامج كالتالي:

MOV CX, Count (للتعليمة 4) + [(Count – 1)x(16 + 2)] (للحلقة) +
(لآخر مرور بالحلقة)(4 + 2)

وانطلاقا من فرضية أن المعالج يعمل على تواتر 10 MHz فإن الزمن المطلوب للدورة الساعية الواحدة هو 1/10 MHz = 0.1 µs وهو الـ Multiplier Count. بالتالي لحساب التأخير المولد بالحلقة الداخلية في هذا المثال من أجل العَد الأعظمي وقدره (65536 = FFFFH) يكون

$0.1x \{4 + [(65536 - 1)x(16 + 2)] + (2 + 4)\} = 118.14$ ms

ولحساب Multiplier Count للحصول على تأخير زمني بمقدار دقيقة يكون

Multiplier Count = Required Delay / Delay provided by inner loop =

1 x 60 s/118.14ms = 509 ~ 1FDH

أي أن القيمة 1FDH هي التي يجب تخزينها بالمسجل BX للحصول على المطلوب.

٣-١٧ تعليمات المقاطعة

وتضم هذه المجموعة من التعليمات ثلاث تعليمات نعرضها وفق التالي:

- **التعليمة INT** : وصيغتها العامة هي INT Type (Number)

تسمح معظم المعالجات أن يقاطع تنفيذ البرنامج الذي تؤديه عن طريق بعض الإشارات الخارجية أو عن طريق تعليمة خاصة في البرنامج. عندما يقاطع المعالج يوقف تنفيذ برنامجه الحالي ويستدعي CALL الإجراء الذي يقوم بخدمة المقاطعة.

- **التعليمة INTO**: وصيغتها العامة هي INTO بدون معامل

وتعني Interrupt on Overflow أي المقاطعة على الطفح أو الفيضان

التعليمة IRET: وصيغتها العامة هي IRET بدون معامل

وتعني Interrupt Return . وتستخدم التعليمة IRET عند نهاية إجراء خدمة مقاطعة وذلك لعودة التنفيذ إلى البرنامج المقاطع. عندما يستجيب المعالج 8086 لإشارة مقاطعة أو لتعليمة مقاطعة، فإنه يدفع المؤشرات، وقيم CS و IP الحالية على المكدس، ومن ثم يحمّل CS و IP بعنوان البداية للإجراء الذي نكتبه من أجل الاستجابة لتلك المقاطعة.

تشبه التعليمة IRET التعليمة RET ولكن يجب التميز بينهما إذ لا ينبغي مثلا استخدام التعليمة RET للعودة من إجراء المقاطعة لأنها لا تسترجع الأعلام من المكدس إلى مسجل الأعلام. وبالنسبة للتعليمة RET يمكن أن تكون متبوعة بعدد مثل RET6 وفي هذه الحالة سيزداد SP بمقدار ستة عناوين بعد أن يستخرج CS, IP من المكدس.

عندما ينفذ المعالج تعليمة INT Type يجري الأعمال التالية

- ينقص المؤشر SP بمقدار 2 ويدفع PUSH المؤشرات على المكدس
- ينقص المؤشر SP بمقدار 2 ويدفع PUSH محتويات CS على المكدس
- ينقص المؤشر SP بمقدار 2 ويدفع PUSH الإزاحة Offset للتعليمة التالية بعد التعليمة INT Type على المكدس.
- يحصل على قيمة جديدة للمسجل IP من عنوان ذاكري مطلق وقيمة هذا العنوان مساوية إلى أربعة أضعاف قيمة النوع Type المحدّد في التعليمة. فمثلا من أجل INT 8 تتم قراءة قيمة IP الجديدة من العنوان 00020H .
- يحصل على قيمة جديدة للمسجل CS من عنوان ذاكري مطلق قيمته مساوية إلى أربعة أضعاف النوع المحدّد في التعليمة زائد 2. فمثلا من أجل INT 8 ستقرأ قيمة CS الجديدة من العنوان 00022H.
- يصفر المؤشران IF و TF بينما لا تتأثر بقية المؤشرات الأخرى.

إن مصطلح Type في التعليمة INT يشير إلى عدد مابين 0 و 255 لتعريف نوع المقاطعة. يمكننا التميز بين نوعين من المقاطعة: مقاطعة DOS ومقاطعات BIOS وكل مقاطعة تملك أكثر من برنامج خدمة يؤدي كل منها غرضاً معيناً. نتعرف فيما يلي على المقاطعة 21H من مقاطعات DOS والمقاطعة 10 H من مقاطعات BIOS.

١٧-٣-١٠ المقاطعة 21H

تتعلق المقاطعة 21H بوظائف DOS للإدخال من لوحة المفاتيح وإظهار الخرج على الشاشة، وهي تستخدم المسجل AH لتحديد نوع الخدمة المطلوبة منها. نستعرض فيما يلي من خلال الجدول 2-17، أهم وظائف DOS المتاحة تحت 21H.

فمثلا: جملة التعليمات التالية:

```
MOV   DL , 41H     ; ASCII Code for the letter A
MOV   AH , 2H      ; Specifies function number 2
INT   21H          ; Do it          تظهر الحرف A على الشاشة
```

الجدول2-17: مفهوم بعض الوظائف المتاحة تحت المقاطعة 21H

Function	Description	What it does
1	Keyboard input	تنتظر إلى حين ضغط حرف على لوحة المفاتيح لتضع ASCII Code للحرف المضغوط في المسجل AL حصراً.
2	Output on the Display Screen	تطبع على شاشة الإظهار الحرف الذي كوده بالـ ASCII موجود في المسجل DL
5	Print Character	الحرف الذي كوده بالـ ASCII موجود في المسجل DL يرسل إلى الطابعة
9	Display String	تطبع كل سلسلة الأحرف المخزنة بالذاكرة، ابتداء بواحد من العناوين المعطاة بـ DX (نسبة لـ DS) وتقف عند ASCII Code للإشارة $
4C		تنهي تنفيذ البرنامج وتعيد عملية القيادة إلى DOS. يمكن أن يوضع مستوى الخطأ في AL ورقم العودة 4C في AH.

مثال: اكتب برنامجاً يحاكي عمل المفتاح Caps Lock

الجواب: بما أن الفرق بين كل كود ASCII لحرف صغير ومقابله كحرف كبير هو 20H يكون البرنامج المطلوب وفق التالي:

```
Next:  MOV   AH , 1H        ; Keyboard input
       INT   21H            ; Read a character into AL
       SUB   AL , 20H       ; Convert small letter into capital
       MOV   DL , AL
       MOV   AH , 2H
       INT   21H            ; Display the converted character
       JMP   Next
```

١٧-٣-٢ المقاطعة 10H

تستخدم المقاطعة 10H لمعالجة مختلف الأمور المتعلقة بالشاشة وهي من مقاطعات النظام BIOS. بالتالي لمعرفة آلية عمل هذه المقاطعة لابد من الإشارة إلى أن الشاشة تمثل بشبكة من المواقع القابلة للعنونة. تستطيع المشيرة التوضع في أي موقع فيها. يبين الجدول أدناه مجموعة لمواقع المشيرة Cursor.

Hexadecimal Format		Decimal Format		Location
Column	Row	Column	Row	Description
00	00	00	00	زاوية أعلى اليسار
4F	00	79	00	زاوية أعلى اليمين
27 / 28	0C	39 / 40	12	مركز الشاشة
00	18	00	24	زاوية أدنى اليسار
4F	18	79	24	زاوية أدنى اليمين

تتألف وحدة المراقبة المرئية النموذجية من 25 سطراً (24 ÷ 0) ومن 80 عموداً (79 ÷ 0) ويمثل ذلك بشبكة من المواقع التي تنتقل ضمنها المشيرة إلى أي موقع. يجهز الحاسوب بذاكرة وسيطة Buffer للشاشة تقع ضمن حيز مخصص من الذاكرة وتبدأ ذاكرة الشاشة أحادية اللون Monochrome في منطقة BIOS عند الموقع (العنوان) B0000H . وكانت في البدايات تشغل حيزاً من الذاكرة بحجم 6 KB، أربعة كيلو منها متوفرة لتمثيل المحارف و2 KB الأخرى لوضع مواصفات Attribute المحارف من وميض Blinking وحدة Intensity وتسطير Underlining وغيرها. أما شاشات الإظهار الملونة أو إظهار الرسوم البيانية Graphics فكانت في البدايات تشغل حيز 16 KB تبدأ عند الموقع B8000H وبالإمكان استخدامها بنمطين هما: النمط النصي Text لإظهار المحا رف أو في نمط الرسوم البيانية Graphics. يمكن بشكل تقني في البرامج نقل المعطيات مباشرة إلى ذاكرة الشاشة الوسيطية لإظهارها، لكن الطريقة الموثوقة هي استخدام

تعليمات المقاطعة المناسبة. حيث تعليمة المقاطعة INT 10H مثلا تسمح بإظهار المعطيات ونقل الCursor إلى أي موقع وتنفيذ عملية Clear بإخلاء الشاشة من الحروف ويلزم إضافة لذلك بالنسبة للشاشات الملونة وضع رقم الصفحة Page في المسجل BH وهو عادة الرقم 0. وهنا تجدر الإشارة إلى ان ذاكرة الإظهار للشاشة تقسم عند استخدامها في النمط النصي إلى أربع صفحات ترقم من 0 وحتى 3 في الشاشة الموافقة من 40 عمود أو من 0 إلى 7 في الشاشات الموافقة من 80 عمود وفق الشكل:
B8000 : | page0|page1|page2|page3| وتمثل الصفحة الأولى ذات الرقم 0 صفحة العمل الحالية، علماً أننا نستطيع تجهيز وإظهار أية صفحة من الصفحات الأربع التي تشكل كل منها 4KB و 2KB . يتم تحديد رقم السطر في المسجل DH ورقم العمود في DL وتبين التعليمات التالية، كيفية وضع المشيرة في السطر 5 والعمود 12 (علماً أنه بالإمكان وضع القيمتين بتعليمة واحدة من الشكل: MOV DX , 050CH)

```
MOV  AH , 02H    ; Request Set Cursor
MOV  BH , 00H    ; Page number 0
MOV  DH , 05H    ; Row 5
MOV  DL , 12H    ; Interrupt-exit to BIOS
```

ولمسح الشاشة Clearing the Screen ، نعلم بأن كافة المحثات Prompts والأوامر والمعطيات تبقى على الشاشة إلى حين الكتابة فوقها أو إزاحتها نحو الأعلى.
وبما أننا مع بداية تنفيذ أي برنامج نرغب بمسح الشاشة يمكننا ذلك من خلال مقاطعة BIOS بالتعليمة INT 10H التي تنفذ المسح بدءاً من أي موقع نريد وانتهاء عند أي موقع نريد ونستطيع إزاحة معطيات الشاشة نحو الأعلى Scrolling up.
لتحقيق عملية المسح، يجب استخدام رقم خدمة المسح 06H ووضعها في AH. ويمكن تعريف نافذة من خلال تحديد رقم سطرها العلوي في المسجل CH وسطرها السفلي في المسجل DH ورقم عمودها اليساري في المسجل CL وعمودها اليميني في المسجل DL. ونحتاج بالإضافة لذلك إلى وضع عدد الأسطر التي سوف تزاح نحو الأعلى ضمن هذه النافذة في AL. يمكن بالقيمة 00H مسح كامل النافذة التي يتم تحديدها بوساطة المسجلين

CX و DX. تضع الخدمة 6H من المقاطعة 10H محارف الفراغ Blank في السطر السفلي الذي تزاح معطياته نحو الأعلى، وتؤخذ صفة محارف الفراغ الجديدة من خلال المسجل BH. نكتفي بهذا الحد بالنسبة للمقاطعة 10H.

نركز بالأمثلة القادمة على استخدام المقاطعة 21H فقط، حيث يمكننا من خلالها إجراء طباعة سطر جديد والعودة إلى DOS.

فعملية طباعة سطر جديد مثلا (أي مفتاح Enter) تحاكي ما كان يتم في الالات الكاتبة القديمة ، إذ كان يتم إعادة الجزء المتحرك من آلة الكاتبة، أي Carriage التي تمسك الورق إلى بداية السطر وتسمى هذه العملية Carrage Return اختصارا CR وهو رمز موجود بجدول ASCII بالقيمة 0DH. وبعد ذلك يتم تحريك الورقة إلى الأعلى بسطر واحد وتسمى العملية Line Feed اختصارا LF ولها بجدول ASCII القيمة 0A وبالتالي لإجراء سطر جديد تتم أولا عملية CR يليها عملية LF.

مثال: Display on the screen "NO" followed by a new line

if "NO" Entered at a) keyboard

b) Memory c) Register

a) Keyboard:

```
MOV AH, 1H
INT 21H
MOV DL, AL
MOV AH, 2H
INT 21H
MOV AH, 1H
MOV DL, AL
MOV AH, 2H
INT 21H
```

b) Memory

```
. DATA
   Lab  DB  'NO$'
. CODE
   Begin Proc Far
   MOV AX, DATA
   MOV DS, AX
   MOV DX, offset lab
   MOV AH, 9H
   INT 21H
```

c) Register

```
. CODE
   Begin Proc Far
   MOV DL, 'N'
   MOV AH, 2H
   INT 21H
   MOV DL, 'O'
   INT 21H
```

وكل شكل من الأشكال الثلاثة المبينة أعلاه ينهي بجملة التعليمات التي تشكل سطراً جديداً وهي:

```
MOV DL, 0DH
INT 21H
MOV DL, 0AH
INT 21H
```

مثال: اظهر عدد مؤشر من رقم واحد من المجال (+9 , -9) على شاشة الاظهار.

```
   MOV   BL , N
   CMP   BL , 0H
   JL      Label1
   MOV   DL , ' + '
   MOV   AH , 2H
   INT     21H
   JMP     Label2
```

```
Label1: MOV   DL , ' - '
        MOV   AH , 2H
        INT     21H
        NEG     BL

Label2: MOV DL, BL
        ADD   DL , 30H
        MOV   AH , 2H
        INT     21H
```

مثال: اكتب برنامجاً يعمل على إدخال رقم ما من المجال 9 ÷ 0 من لوحة المفاتيح وإظهاره على الشاشة، وإظهار مربعه على السطر التالي:

```
MOV   AH , 1   INT    21H
MOV   DL , AL
MOV   AH , 2H
INT    21H
MOV    DL , 0D    ; New Line
INT    21H
MOV    DL , 0AH
INT    21H
SUB    AL , 30H
MUL    AL
MOV    CL , 10D
DIV    CL
MOV    DL, AL
ADD    DL, 30H
MOV    BH, AH
MOV    AH, 2H
INT     21H
MOV    DL, BH
ADD    DL, 30H
INT     21H
MOV    AX, 4C00H
INT     21H
```

وتضم هذه المجموعة أربع تعليمات، جميعها لاتؤثر بالمؤشرات، نعرضها وفق التالي:

- التعليمة **LOOP**: وصيغتها العامة هي: LOOP Label

وتعني LOOP To Specified Label Until CX = 0 أي الدوران حول اللافتة Label حتى يصبح محتوى المسجل CX = 0. تستخدم هذه التعليمة لتكرار سلسلة من التعليمات عدداً من المرات ، بحيث إن عدد المرات المطلوب فيها تكرار السلسلة يحّمل في المسجل CX. يتم إنقاص CX في كل مرة تنفذ فيها السلسلة بشكل آلي بمقدار 1. وعندما يصبح CX = 0 بعد التناقص الالي، فإن التنفيذ سيذهب ببساطة إلى التعليمة التالية لتعليمة LOOP .

- التعليمة **LOOPE/LOOPZ** : وصيغتها LOOPE/LOOPZ Label وهنا يستخدم الرمز "/" للدلالة على أن الاختصارين يمثلان نفس التعليمة، وتعني هاتان التعليمتان LOOP While CX ≠ 0 and ZF = 0 ، أي الدوران في الحلقة طالما أن CX ≠ 0 و ZF = 1 . بالطبع عدد المرات المطلوب فيها تكرار السلسلة يكون محّمل في مسجل العّد CX وفي كل مرة تنفذ فيها التعليمة LOOP يتناقص CX بشكل آلي بمقدار 1. فإذا لم يكن CX = 0 وكان ZF = 1، فإن التنفيذ سيقفز إلى الـ Label المحدد في التعليمة . أما إذا أصبح CX = 0 أو ZF = 0، فإن التنفيذ ينتهي ويذهب إلى التعليمة التالية للتعليمة LOOPE/LOOPZ.

- التعليمة **LOOPNE / LOOPNZ**: وصيغتها LOOPNE/LOOPNZ Label وتعني LOOP While CX ≠ 0 and ZF = 1، أي الدوران في الحلقة طالما أن CX ≠ 0 و ZF = 0 . أي أنها مماثلة تماماً للتعليمتين السابقتين.

- التعليمة **JCXZ**: وصيغتها العامة هي JCXZ Label

وتعني JUMP if the CX Register is Zero، أي القفز إذا كان محتوى المسجل CX صفراً . تسبب هذه التعليمة قفزاً إلى Label معطاة في التعليمة إذا كان CX = 0. وإذا لم يحتوي CX على أصفار فإن التنفيذ يتقدم إلى التعليمة التالية .

تستخدم LOOP للحلقات التي تحدد بشكل مسبق Predetermined عدد التكرارات كما هو الحال مع حلقة FOR.

مثال: اكتب برنامج يعمل على تحويل كافة الأحرف المخزنة بالذاكرة ابتداء من الموقع بالاسم Letters والتي عددها 128H حرف من الحروف الكبيرة إلى الأحرف الصغيرة.

```
        MOV   BX , offset Letters
        MOV   CX , 128H
Next:   MOV   AL , [BX]
        ADD   AL , 20H
        MOV   [BX] , AL
        INC    BX
        LOOP  Next
```

أي كان بالإمكان في هذا المثال تعويض أخر تعليمة بالتعليمتان: `DEC CX` و `JNZ Next`

وتقوم التعليمة LOOP بإنقاص المسجل CX في كل مرة تنفذ فيها السلسلة بمقدار 1 حتى ولو لم يذكر ذلك إلى حين أن يصبح CX = 0.

أما بخصوص التعليمة LOOPZ فهي تعمل عمل التعليمة LOOP بإنقاص قيمة المسجل CX طالما CX ≠ 0 و ZF = 1، بينما التعليمة LOOPNZ تعمل أيضاً عمل التعليمة LOOPZ بإنقاص قيمة CX طالما CX ≠ 0 و ZF = 0.

ملاحظة: بالنسبة للتعليمة LOOP يجب أن يكون عنوان الوجهة بالمجال Byte +127 ÷ -128 اعتبارا من التعليمة التالية لـ LOOP.

نوضح فيما يلي، استخدام التعليمتان من خلال المثال السابق بعد إضافة شرط ما اضافي للإنهاء وهو التالي:

مثال: بفرض أن النص بطول 128H يحوي بعض الأحرف الكبيرة Xs (واحد على الأقل) ونريد فقط تحويل الـ Xs من الأحرف الكبيرة إلى أحرف صغيرة، على أن يترك الباقي على حاله بدون تحويل. تستخدم بهذه الحالة التعليمة LOOPZ وفق التالي:

```
MOV   BX , offset Letters
MOV   CX , 128H
      MOV   AL , [BX]
Next: ADD   AL , 20H
      MOV   [BX] , AL
      INC    BX
      MOV   AL , [BX]
      CMP   AL , ' X '
      LOOPZ  Next   ; if an ' X ' then repeat provided CX ≠ 0
                    ; Loop continue While CX ≠ 0 and ZF = 1
```

وتعليمة LOOPNZ تعمل بشكل مماثل، لكن شرط الإنهاء هو CX≠ 0 و ZF= 0.

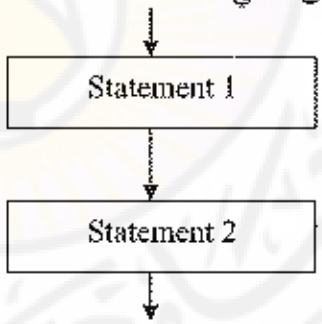
ساعدت الأدوات البرمجية القياسية على اكتشاف أن أي عملية برمجية سوف تمثل بثلاثة أنواع أساسية من العمليات. أولى هذه العمليات هو التتابع Sequence أي تسلسل تنفيذ مجموعة من الأعمال. وثانيها هو القرار Decision or Selection أو الاختيار مابين أمرين بديلين، أما النوع الثالث فهو التكرار Repetition or Iteration أو الإعادة لسلسة من الأعمال حتى يتحقق أو لا يتحقق شرط معين.

واعتمادا على ذلك تم اقتراح: على أن يستخدم كل المبرمجين مجموعة مؤلفة من ثلاث إلى سبع بنى قياسية لتمثيل كل العمليات في البرنامج، بحيث يمكن للبنية الواحدة أن تمتلك بنى أخرى من نفس النوع أو من أنواع أخرى وكل بنية تملك مدخلا وحيداً Entry ومخرجاً وحيداً Exit فقط . نقسم فيما يلي البنى البرمجية إلى مجموعتين هما:

الأولى: وتضم البنى WHILE – DO, IF – THEN – ELSE SEQUENCE

الثانية: وهي مجموعة بنى مشتقة من بنى المجموعة الأولى وتستخدم لجعل البرنامج أكثر وضوحاً. ويمكن استخدام الشفرة الزائفة كبديل لرموز المخطط الانسيابي وفق التالي:

1– Simple Sequence flowchart

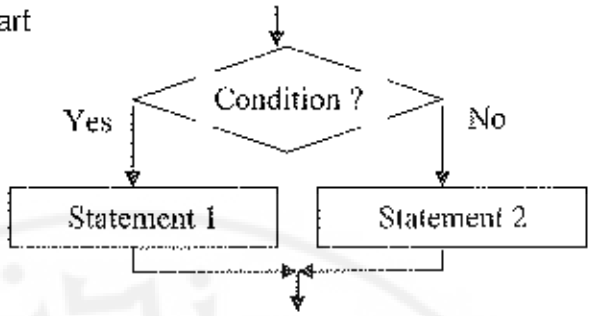


Pseudo code: stat 1
stat 2
.....

ex.) Get data sample
ADD 10
Store in memory location

وتكتب في هذه البنية الأفعال (أو الجمل) ببساطة بالترتيب المطلوب.

2- IF THEN ELSE Flowchart



Pseudo code:

If condition then Statement1

else Statement2

Ex.) if Hungry then

get food

else make sport

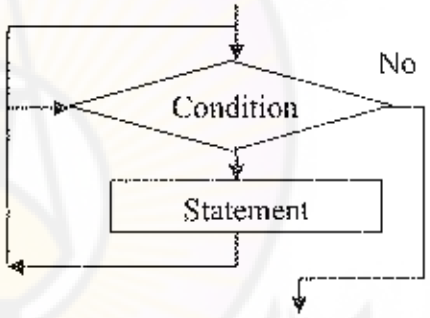
وتستخدم هذه البنى للتوجيه العملية لأحد عمليتين مختلفتين اعتماداً على أحد الشروط

3- WHILE – DO LOOP Flowchart

Pseudo code:

While condition Do

Statement(s)



تمثل هذه البنية إحدى صيغ التكرار المستخدمة للإشارة إلى أننا نريد تنفيذ أحد الأفعال (أو سلسلة من الأفعال) طالما أن أحد الشروط محقق

4- REPEAT UNTIL Flowchart

Pseudo code:

Repeat Statement(s)

Until condition

تعدّ البنية Repeat–Until مشتقة من البنية While–Do وتستخدم للإشارة، فيما إن كنا نريد من البرنامج أن يعيد أحد الأفعال (أو سلسة من الأفعال) حتى يتحقق شرط معين، أي أن هذه البنية تنفذ العبارة (أو العبارات) أولا ومن ثم تنفذ الشرط، بمعنى أن هذه البنية تنفذ العمل أو سلسلة الأعمال مرة واحدة على الأقل.

ex.) Repeat

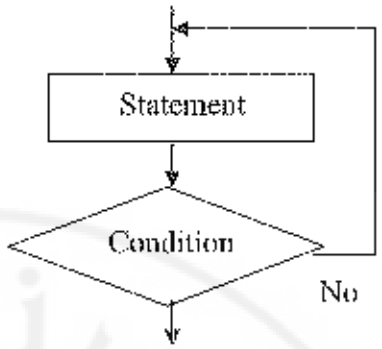
get data sample

ADD 7

Store Result in memory

Wait one Hour

Until 24 Samples taken



بما أن البنية Repeat– Until مشتقة من البنية While–Do فإن أي مشكلة تمثل باستخدام While – Do ، يمكن تمثيلها بالبنية Repeat-- Until.

تحتوي البنيتان Repeat–Until , While – Do في داخلهما عملية القرار البسيطة IF–THEN–ELSE وهكذا يعدّ القرار جزءاً ضمنياً في هذه البنى.

يوجد صيغة أخرى لعملية التكرار يمكن مصادفتها في برامج اللغات العليا هي حلقة FOR–DO ولها الصيغة التالية:

For count = 1 To n Do

stat 1

stat 2

وينفذ هذا النوع من العمليات عادة بلغة الأسمبلي بالبنية Repeat–Until.

5– CASE Flowchart

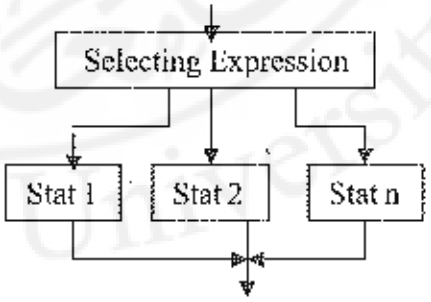
تعدّ هذه البنية مجرد طريقة مضغوطة لتمثيل سركبة معقدة لـ IF–THEN– ELSE وهي تمثل طريقة مضغوطة لتمثيل الخيارات بين عدة أفعال بديلة.

Pseudo code: Case Expression Of

1. statement 1
2. statement 2

.....

n. statement n



ومن التراكيب المتبقية:

6- IF THEN Structure

Pseudo code:

```
If condition then
Statement(s)
```

7-- CASE Expressed as Multiple IF – THEN – ELSE Structure

Pseudo code:

```
IF Monday THEN
Make Logic
ELSE IF Tuesday  THEN
Make Microprocessor
ELSE IF Wednesday THEN
Make Computer .....
```

١٨-١ تنفيذ بعض البنى البرمجية الأساسية بلغة الأسمبلي

يعدّ البرنامج المكتوب بالطريقة البنيوية أسهل في التنقيح Debugging ومفهوم أكثر من قبل المبرمجين الاخريين الذين سيتعاملون معه، كما يكون سهل التطبيق في لغة الأسمبلي وغيرها من اللغات العالية المستوى مثل " C ".

نظهر فيما يلي آلية تنفيذ بعض البنى البرمجية بلغة الأسمبلي بتمثيل كل بنية بلغة عليا مثل الـ Pascal ، وبعد ذلك نظهرها بلغة الأسمبلي لتتسنى عملية المقارنة.

– البنية Repeat ... Until: بلغة الـ Pascal يمكن تمثيلها بالشكل التالي:

```
Repeat      Action 1
            .....
            Action n
Until       Condition
```

يمكن تنفيذ هذه الصيغة (البنية) باستخدام الـ Pseudo operation التالية:

```
Repeat: Implement   of Action 1
        ..........
        Implement   of Action n
        If Condition not True    Jump to Repeat
```

فإذا سمحنا باستخدام أسماء بلغة الأسمبلي لاسماء متغيرات بلغة الباسكال، يمكن التعبير عن ذلك بالشكل:

```
Repeat: AX := AX – 1
        BX := BX + 1
        CX := AX – BX
Until   AX = 0
```

ويمكن صياغة ذلك بلغة الأسمبلي على النحو التالي:

```
Repeat:  DEC   AX
         INC    BX
         MOV   CX , 0
         ADD   CX , AX
         SUB    CX , BX
         CMP    AX , 0
         JNZ     Repeat
```

– بنية While ... DO: يمكن أن تنفذ بأسلوب مماثل لما سبق وفق التالي:

```
WHILE Condition   DO    BEGIN
                          Action 1
                          Action n
                        END
```

وهي تكافئ الصياغة التالية:

```
Start:  If  Condition  Not True,  JUMP to Stop
            Action 1 .....
            Action n
            JMP  Start  Stop:
```

ويمكن صياغتها بكل من لغتي الباسكال والأسمبلي بالشكل التالي:

بلغة الـ Pascal:

```
Wile AX < 100D  DO
Begin
  AX := AX + BX
                BX := BX + 1
                End
```

بلغة الـ Assembly:

```
Start:  CMP  AX , 100D
        JNB  Stop
                    ADD  AX , BX
                    INC   BX
                    JMP   Start
               Stop:
```

- بنية FOR .... DO: الشكل العام لهذه البنية هو

```
FOR  I := Initial Value  TO  Final Value  DO
        Begin          Action 1
                       .........
                       Action n
        END
```

وهي تقريباً على نفس النمط للتركيبة التالية بلغة الأسمبلي التي فيها الـ Byte في الذاكرة يشار له بوساطة الاسم Loop-counter الذي يستخدم لتسجيل القيمة من الحلقة FOR

والعَد I كالتالي:

```
        MOV  Loop-counter, Initial Value
Start:  JA   Stop
        Action 1
        ..........
        Action n
        INC  Loop-counter
        JMP  Start
Stop:
```

برنامج بلغة الباسكال:

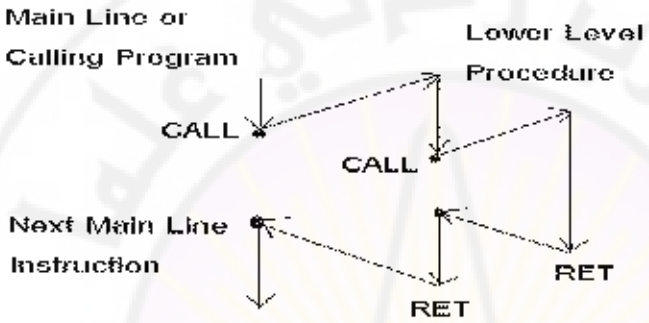
```
FOR I := 3 TO  25D  DO
        Begin
        AX := AX + BX
        BX := BX + 1
        END
```

يمكن تنفيذه بلغة الأسمبلي وفق التالي:

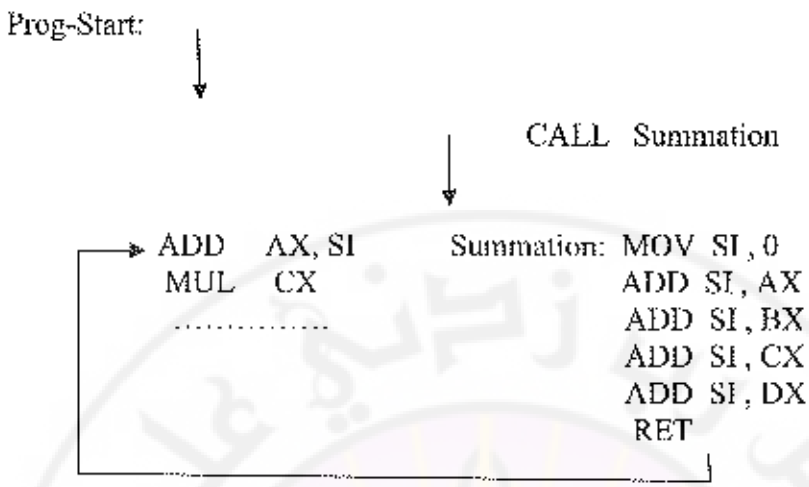
```
          MOV  Loop-counter , 3
  Start: CMP  Loop-counter , 25D
         JA     STOP
         ADD  AX , BX
         DEC    BX
         INC    Loop-counter
         JMP    Start      Stop:
```

١٨-٢ الإجراءات (البرامج الفرعية) بلغة الأسمبلي

من المعلوم بأنه كلما كان لدينا سلسلة من التعليمات التي نريد تنفيذها مرات عديدة في البرنامج، نكتب هذه السلسلة من التعليمات كبرنامج فرعي Procedure، وبعد ذلك يمكن استدعاء CALL هذا البرنامج الفرعي المنفصل في كل مرة نريد فيها تنفيذ تلك السلسلة من التعليمات. نوضح فيما يلي بشكل تخطيطي كيف يذهب تنفيذ البرنامج من الخط الرئيسي إلى إجراء ما، ثم يعود إلى الخط الرئيسي مرة أخرى .



تحّمل التعليمة CALL في الخط الرئيسي للبرنامج مسجل مؤشر التعليمات IP وفي بعض الحالات تحّمل أيضاً مسجل قطعة الشفرة بعنوان البداية للإجراء وستكون التعليمة التالية لتعليمة CALL بعدئذ هي التعليمة الأولى للإجراء. وعند نهاية الإجراء تعيد تعليمة العودة RET التنفيذ إلى التعليمة التالية لتعليمة CALL في برنامج الخط الرئيسي، حيث تقوم التعليمة RET بتنفيذ ذلك بتحميل مسجل مؤشر التعليمات بعنوان التعليمة التالية للتعليمة CALL. بالطبع يمكن لاجراء ما كما هو مبين بالمخطط أعلاه أن يستدعي إجراء آخر وهذا ما يدعى بتعشيش الإجراءات (تداخلها) Nesting Procedures. لتنجيز عمليتي CALL و RET يلزم مقطع خاص من الذاكرة لحفظ عناوين الانتقال والعودة، يدعى بالمكدس Stack. بفرض أننا نحتاج إلى جزء من برنامج يترك مجموع أعداد غير موشرة بطول 16 bits موجودة في المسجلات AX, BX, CX, DX بالمسجل SI ولنجز ذلك بداخل إجرائية تدعى Summation، كما هو موضح بالمخطط أدناه.



تظهر الملاحظات في جزء البرنامج أدناه سلسلة القيم التي تؤخذ بوساطة SI

```
MOV   AX , 0
MOV   BX , 0
MOV   CX , 0
MOV   DX , 1
CALL  Summation
MOV   AX , 2
CALL  Summation
MOV   BX , 3
MOV   CX , 4
CALL  Summation
```

سيكون لهذا البرنامج بعد أول CALL :
; SI has now been set to 1
وسيكون بعد ثاني CALL :
; SI has now been set to 3 (1 + 2 = 3)
وسيكون بعد ثالث CALL
; SI has now been set to
(0 + 2 + 3 + 4 + 1 = 10 = 0AH)
لأنه كل مرة تنفذ الإجرائية المبينة بالمخطط أعلاه.

١٨-٢-١ المكدس في المعالج 8086

يعرف المكدس بأنه جزء من الذاكرة يحجز خصيصا لتخزين عناوين العودة، ويستخدم أيضاً لحفظ محتويات المسجلات من أجل برنامج الخط الرئيسي عندما ينفذ الإجراء. والاستخدام الثالث للمكدس هو حفظ المعطيات أو العناوين التي سيتعامل معها الإجراء يسمح المعالج 8086 بحجز أو تخصيص قطعة كاملة من الذاكرة بحجم 64 KB على انها مكدس. ويحتوي المعالج 8086 على مسجل قطعة المكدس SS وعلى مسجل مؤشر المكدس SP وكل منهما بطول 16 bit (وقد سبق وأشرنا بالجزء الثاني إلى آلية حساب العنوان الفيزيائي لموقع مكدسي). وقد نحتاج في كل برنامج إلى تعريف المكدس (أي تهيئة كل من مسجل قطعة المكدس ومسجل مؤشر المكدس). نبين فيما يلي الأجزاء المطلوب إضافتها إلى البرنامج للتصريح عن المكدس

```
STACK-HERE      SEGMENT STACK
          DW    40    DUP(0)
STACK-TOP       LABEL       WORD
STACK-HERE      ENDS
CODE-HERE       SEGMENT
                ASSUME CS: CODE-HERE,
                       SS: STACK-HERE
                MOV   AX, STACK-HERE ; initialize stack
                MOV   SS, AX
                LEA   SP, STACK-TOP ; initialize stack
                                    ; pointer
                .....
CODE-HERE       ENDS
                END
```

٣-١٨ الماكرو Macro

توجد طريقتان لتفادي كتابة مجموعة من التعليمات في البرنامج بشكل متكرر وذلك في كل مرة نريد استخدامها فيها.

تتمثل إحدى هذه الطرق بكتابة هذه المجموعة من التعليمات كإجراء منفصل Separate Procedure، كما أشرنا لذلك أعلاه، وبعدئذ نستطيع أن نستدعي (CALL) الإجراء كلما احتجنا لتنفيذ تلك المجموعة من التعليمات. إن الميزة الكبيرة لاستخدام الإجراء، أن شيفرات الآلة لهذه المجموعة من التعليمات في الإجراء، يجب أن توضع في الذاكرة لمرة واحدة فقط. أما سيئات الإجراء فهي الحاجة للمكدس والزمن الزائد اللازم لاستدعائه والعودة إلى البرنامج المستدعي.

الطريقة الثانية وتستخدم عندما تكون المجموعة المكررة من التعليمات قصيرة جداً أو غير مناسبة لكي تكتب كإجراء، حيث يستخدم الماكرو الذي يمثل مجموعة من التعليمات التي نحيطها بأقواس ونعطيها اسماً ما عند بداية البرنامج.

يقوم المجمع في كل مرة " يستدعي" فيها الماكرو في البرنامج، بغرس Insert المجموعة المعرفة من التعليمات في مكان الاستدعاء CALL. بمعنى أن استدعاء الماكرو هو تعبير مختزل يخبر المجمع بما يلي: " في كل مرة ترى فيها اسم ماكرو في البرنامج، استبدله بمجموعة التعليمات المعرفة لذلك الماكرو الموجود عند بداية البرنامج". والنقطة المهمة هنا أن المجمع يولد شيفرات الآلة لمجموعة التعليمات في كل مرة يستدعى فيها الماكرو. إذ يدعى استبدال الماكرو بالتعليمات التي تمثله عموما " بتوسع" Expanding الماكرو. وبما أن شيفرات الآلة المولدة تكون صحيحة وموجودة " في الأسطر" مع بقية البرنامج فلا ينبغي على المعالج أن يذهب إلى الإجراء ويعود.

ولذلك فإن استخدام الماكرو يجنبنا الزمن الزائد في عملية الاستدعاء والعودة من الإجراء، لكن السيئة في توليد شيفرات الأسطر في كل مرة يستدعى فيها الماكرو. إن هذه الطريقة تجعل البرنامج يستهلك ذاكرة أكبر.

بفرض أننا أردنا كتابة إجراء في برنامج للمعالج 8086، هذا يعني أننا قد نحتاج عند بداية الإجراء إلى تخزين المؤشرات وجميع المسجلات بدفعها push على المكدس، وبالتالي

نحتاج عند نهاية الإجراء استعادة المؤشرات والمسجلات بالتعليمات pop من المكدس. أي أن الإجراء بشكل طبيعي سيحتوي على سلسلة طويلة من تعليمات push في البداية وعلى سلسلة طويلة من تعليمات pop في النهاية . وهذا عمل ممل وعرضه للأخطاء. ولذلك ينبغي كتابة إجراء يقوم بالتخزين، وآخر يقوم بالاستعادة. أي أن هذه الاضافات قد تعقد البرنامج، وبالتالي لاتكون مناسبة له.

وباستخدام قواعد Syntax مجمّع الماكرو MASM (Macro ASMmbler) يمكننا حل مثل هذه المشكلة باستخدام اثنين من المايكروات، فمثلا المايكرو لتخزين المسجلات يمكن أن يكون من الشكل التالي:

```
PUSH   ALL   MACRO
PUSHF
PUSH  AX
PUSH  BX
ENDM
```

تعرّف العبارة PUSH ALL MACRO بداية الماكرو وتعطيه اسما، وتعرف ENDM نهاية الماكرو. لاستدعاء الماكرو في أحد إجراءاتنا نضع اسم الماكرو، وعندما يترجم المجمع هذا المقطع من البرنامج فإنه يستبدل الاسم PUSH ALL بالتعليمات التي تمثله ويغرس شيفرة الآلة للتعليمات في نسخة الشيفرة الغرضية للبرنامج (لوائح المجمع تخبرنا بالأسطر التي غرست بوساطة استدعاء الماكرو وذلك بوضع إشارة + عند كل سطر مغروس في البرنامج).

إن استخدام الماكرو يجعل البرنامج المصدري source program أكثر قابلية للقراءة لعدم احتوائه على تلك السلسلة الطويلة من تعليمات PUSH.

٤-١٨ تعليمات التحكم بالمعالج 8086

وتضم هذه المجموعة من تعليمات المعالج الصغري 8086 التي تسمى Processor Control Instruction اثنتي عشرة تعليمة مقسمة إلى ثلاث مجموعات جزئية هي: – تعليمات عدم التشغيل

– تعليمات رفع/ تصفير الأعلام أو المؤشرات

– تعليمات تزامن المكونات الصلبة الخارجية

جميع تعليمات هذه المجموعة هي بدون معاملات ولا تؤثر بالمؤشرات وهي:

I– تعليمات عدم التشغيل: حيث يوجد تعليمة واحدة ، هي:

– **التعليمة NOP** : وصيغتها العامة هي NOP بدون أي معامل

وتعني عدم تنفيذ أي عملية No Operation تستخدم التعليمة NOP لزيادة تأخير حلقة التأخير Delay Loop وتستخدم للاحتفاظ بعدة أمكنة في البرنامج من أجل أية تعليمة يراد إضافتها لاحقاً.

II– تعليمات رفع/ تصفير الأعلام أو المؤشرات:

وتضم هذه المجموعة سبع تعليمات نعرضها وفق التالي:

– **التعليمة STC** : وصيغتها العامة هي STC بدون أي معامل

تعمل هذه التعليمة على رفع مؤشر الحمل ، أي جعل CF = 1 .

وتعني STC Instruction – Set The carry flag to a one

– **التعليمة CLC**: وصيغتها العامة هي CLC بدون أي معامل

تعمل هذه التعليمة على تصفير مؤشر الحمل CF ، أي جعل CF = 0 وتعني CLC Instruction – Clear The carry flag.

– التعليمة CMC : وصيغتها العامة هي CMC بدون أي معامل

تعمل هذه التعليمة على اتمام حالة مؤشر الحمل CF ، وتعني CMC Instruction – Complement The carry بمعنى إذا كان الحمل CF صفرا

قبل هذه التعليمة ، فإنه سيرفع بواحد بعد التعليمة والعكس صحيح.

- **التعليمة STD** : وصيغتها العامة هي STD بدون أي معامل

وتعمل هذه التعليمة على رفع علم الاتجاه DF بواحد، وتعني STD Instruction Set the direction flag to one. تستخدم لرفع علم الاتجاه بواحد، بحيث يتناقص SI و/أو DI بشكل آلي ليشير إلى عنصر السلسة التالي عندما تنفذ إحدى تعليمات السلسلة.

- **التعليمة CLD** : وصيغتها العامة هي CLD بدون أي معامل

وتعمل على تصفير علم الاتجاه DF ، وتعني CLD – Clear Direction flag. تعيد هذه التعليمة علم الاتجاه إلى الصفر. فإذا تم تصفير علم الاتجاه سيزداد SI و DI آليا وذلك عندما يتم تنفيذ إحدى تعليمات السلاسل.

- **التعليمة STI** : وصيغتها العامة هي STI بدون أي معامل

وتعمل على رفع مؤشر المقاطعة IF بواحد (تمكين المدخل INTR)، وتعني STI Instruction – Set Interrupt Flag

إن رفع علم المقاطعة بواحد يؤدي إلى تمكين مدخل المقاطعة INTR في المعالج 8086 بعد التعليمة التالية لتعليمة STI حيث تؤدي إشارة المقاطعة على هذا المدخل إلى مقاطعة تنفيذ البرنامج الرئيسي وإلى دفع عنوان العودة والأعلام على المكدس ومن ثم تنفيذ إجراء خدمة المقاطعة. وستعيد تعليمة IRET عند نهاية إجراء خدمة المقاطعة، كل المؤشرات التي كانت مخزنة على المكدس، وتعيد التنفيذ إلى البرنامج المقاطع.

- **التعليمة CLI** : وصيغتها العامة هي CLI بدون أي معامل

وتعمل هذه التعليمة على تصفير علم المقاطعة (عدم تمكين المدخل INTR)، وتعني CLI Instruction – Clear Interrupt Flag. لاتؤثر هذه التعليمة في مدخل المقاطعة غير المقنعة NMI.

III- تعليمات تزامن المكونات الصلبة الخارجية: وتضم هذه المجموعة أربع تعليمات:

- **تعليمة التوقف HLT:** وصيغتها العامة هي HLT بدون أي معامل

وهي تعليمة من نوع خاص تعني التوقف عن العمل حتى تتم المقاطعة أو التصفير، وتسبب إيقاف المعالج عن جلب التعليمات وتنفيذها، وبعد تنفيذ هذه التعليمة يوجد ثلاث طرائق لوضع المعالج خارج حالة التوقف وهي: وجود إشارة مقاطعة على مدخليه INTR و NMI أو وجود إشارة تصفير على مدخله RESET.

- تعليمة **WAIT:** وهي تعليمة من نوع خاص تعني الانتظار حتى تصبح الإشارة على المأخذ Test منخفضة وتعني الانتظار من أجل إشارة فحص أو إشارة مقاطعة. تستخدم التعليمة WAIT لمزامنة Synchronize المعالج مع المكونات الصلبة الخارجية مثل المعالج الحسابي المساعد 8087 وهي لاتؤثر في الأعلام. عند تنفيذ التعليمة WAIT يدخل المعالج في حالة توقف عن العمل Idle Condition أو انتظار حتى تأتي إشارة مقاطعة على أحد المدخلين INTR, NMI. وإذا حدثت مقاطعة قانونية valid على احد مداخل المقاطعة بينما يكون المعالج في حالة انتظار، فإن المعالج سيعود إلى حالة الانتظار هذه بعد تنفيذ إجراء خدمة المقاطعة، لأن عنوان تعليمة WAIT يكون هو العنوان المدفوع pushed على المكدس عندما يستجيب المعالج لطلب المقاطعة.

- تعليمة **ESC:** وهي تعليمة من نوع خاص تعني الهروب إلى معالج مساعد خارجي. وتستخدم هذه التعليمة لتمرير التعليمات إلى معالج مساعد Coprocessor مثل المعالج الحسابي المساعد 8087 الذي يتشارك بممر العناوين والمعطيات مع المعالج 8086 تمثل تعليمات المعالج المساعد بوساطة شيفرة مكونة من 6 bits مضمنة في التعليمة Escape. ومثلما يجلب المعالج 8086 بايتات التعليمة، فإن المعالج المساعد أيضاً يلتقط هذه البايتات من ممر المعطيات ويضعها في صفه الخاص Queue. كل من المعالجين 8086 و 8087 يتعامل مع تعليمات الآخر على أنها NOP. عندما يجلب المعالج 8086 تعليمة ESC فإن المعالج المساعد يفكك شيفرة التعليمة وينجز العمل المحدد بوساطة الشيفرة 6 bits المعينة في التعليمة.

- تعليمة **LOCK**: وهي تعليمة من نوع خاص تعني بادئة تعليمة، تمنع معالج آخر من السيطرة على الممر عندما تكون التعليمة المجاورة قيد التنفيذ.

وتعني LOCK INSTRUCTION – Assert Bus Lock Signal أي تمكين إشارة قفل الممر. كما هو معلوم تحتوي العديد من أنظمة الحواسيب الصغيرة على عدة معالجات صغرية، ويمتلك كل معالج ذاكرته وممره المحلي الخاص به. وتكون هذه المعالجات موصولة مع بعضها بوساطة ممر النظام System bus بحيث يمكن لكل معالج الولوج إلى موارد النظام Resources. يتحكم كل معالج صغري بممر النظام فقط عندما يريد الولوج إلى أحد موارد النظام. وعند تنفيذ تعليمة لها البادئة Lock، سيمكن المعالج 8086 إشارة قفل الممر على مخرجه. وتصل هذه الإشارة إلى جهاز متحكم الممر الخارجي الذي يمنع بدوره أي معالج آخر من السيطرة على ممر النظام والتحكم به.

١٨ – ٥ أدوات تطوير برامج لغة الأسمبلي

يتطلب العمل البرمجي لتطوير البرامج على المنظومات الصغرية إلى أحد منظومات تطوير الحواسيب الصغرية Microcomputer Development Systems وإلى أدوات تطوير البرنامج Program Development Tools التي تجعل العمل أكثر سهولة. نتعرف فيما يلي عدة أدوات عامة لتطوير البرامج. تمثل أدوات تطوير برامج لغة التجميع برامج تنفذ لتحقيق إحدى وظائف البرنامج الذي نقوم بكتابته. وفيما يلي أهم هذه الأدوات:

1- المحرر Editor

يمثل المحرر برنامج يسمح بتشكيل ملف يحتوي على عبارات لغة الأسمبلي الخاصة بالبرنامج الذي نكتبه. عند تدوين المحارف والأعداد في برنامج ما، فإن المحرر يخزن شفرات ASCII لهذه المحارف والأعداد في مواقع ذاكريه متتالية بالذاكرة RAM. وإذا تم ارتكاب خطأ أثناء تدوين تعليمات البرنامج، فإن المحرر يسمح بالعودة إلى الخطأ وتصحيحه. عند الانتهاء من تدوين كامل ملف البرنامج يخزن على القرص الصلب مثلاً ويسمى بالملف المصدري وستكون الخطوة التالية هي معالجة هذا الملف بوساطة المجمع، وينبغي إعطاء اسم الملف امتداد مثل ASM إذا تم استخدام مجمع مثل MASM .

2- المجمّع Assembler

يستخدم برنامج " المجمّع " لترجمة مختصرات Mnemonics لغة الأسمبلي إلى الشفرات الثنائية المقابلة لها (عوضاً عن التشفير اليدوي). عند تنفيذ المجمّع يتم قراءة الملف المصدري المخزن على القرص بعد عملية التحرير Editing. يحدّد المجمّع عند مروره الأول First Pass عبر البرنامج المصدري، قيم الابتعادات لبنود المعطيات المسماة وإزاحات Offset اللافتات، .....

ثم يضع هذه المعلومات في جدول خاص يدعى " جدول الرموز Symbol Table ". ينتج المجمّع في المرور الثاني له عبر البرنامج المصدري الشفرة الثنائية لكل تعليمة ويغرس الإزاحات وغيرها التي حسبها أثناء المرور الأول.

يولد المجمع ملفين على القرص، يدعى الملف الأول بالملف الغرضي Object File ويعطي له الأمتداد OBJ. وهو يحتوي على الشفرات الثنائية للتعليمات وعلى معلومات عن عناوين هذه التعليمات. يولد الملف الثاني بوساطة المجمع ويدعى " ملف لائحة المجمع " Assembler List File ويعطى الأمتداد LST.

يحتوي ملف اللائحة على عبارات لغة الأسمبلي وعلى الشفرات الثنائية لكل تعليمة وعلى إزاحة كل تعليمة أيضاً. وتشير لائحة المجمع أيضاً إلى الأخطاء القواعدية التي تم ارتكابها أثناء كتابة البرنامج المصدري.

يستخدم المحرر لتصحيح الأخطاء المشار إليها على اللائحة وبعد ذلك تتم إعادة تجميع Reassemble البرنامج المصدري المصحح، أي أن حلقة " التحرير – التجميع" تأخذ عدة دورات قبل أن تعالج كل الأخطاء القواعدية لبرنامج المصدر.

3- محدّد المواقع Locator

يعدّ محدّد المواقع برنامج يستخدم لتعيين العناوين المحددة، حيث ينبغي تحميل قطعة الشفرة الغرضية Object Code في الذاكرة (يسمى برنامج محدد المواقع على الحاسوب الشخصية IBM PC مع نظام التشغيل DOS بالاسم EXE2BIN وهو يحول الملف EXE. إلى الملف BIN.)

4- الرابط Linker

تقسم البرامج الضخمة عند كتابتها إلى وحدات مستقلة أصغر، بحيث يمكن اختبار وتنقيح كل وحدة مستقلة بشكل أفرادي، وبعد أن تعمل كل هذه الوحدات بشكل سليم يمكن ربط هذه الوحدات الغرضية Object Modules مع بعضها لتشكيل برنامج وظيفي ضخم، إذا الرابط يمثل برنامج يستخدم لربط عدة ملفات غرضيه Object بحيث تؤول كلها إلى ملف غرضي واحد ضخم.

يشغّل برنامج Link على الملف OBJ. وينتج الرابط Linker ملف الربط Link file الذي يحتوي على الشفرات الثنائية لكل الوحدات التي جرى ضمها، كما ينتج الرابط أيضاً ملف مخطط الربط الذي يحتوي على معلومات العناوين عن الملفات المربوطة Linked، لكن الرابط لا يعيّن العناوين المطلقة للبرنامج، إنما يعيّن العناوين النسبية المبتدئة من الصفر.

5- المنقح Debugger

يمثل المنقح برنامجاً يسمح بالتوقف عن التنفيذ بعد كل تعليمة، ويسمح بتحميل برنامج الشفرة الغرضية Object Code Program في ذاكرة النظام وبتنفيذه، ثم تنقيحه واكتشاف الأخطاء فيه، حيث يسمح بالنظر إلى محتويات المسجلات والمواقع الذاكرية بعد تنفيذ البرنامج وتبديل محتوياتها وإعادة التنفيذ من جديد.

يأتي المنقح الأساسي مع نظام DOS في حواسيب IBM PC، لكن المنقحات الأكثر قوة مثل المنقح Turbo Debugger من شركة Borland والمنقح Code View من Microsoft تجعل عملية التنقيح أكثر سهولة، لأنها تسمح مباشرة بمراقبة محتويات المسجلات والمواقع الذاكرية عند تغيّرها أثناء تنفيذ البرنامج. يسمى المنقح أيضاً بالبرنامج المراقب أو المشرف Monitor Program لأنه يسمح بمراقبة البرنامج بكل فعالية. ويسمح بوضع نقاط التوقف Break Points عند أي نقطة في البرنامج، وهذا يعني أنه إذا غرست نقطة توقف بعد تعليمة معينة مثلاً فإن المنقح سينفذ البرنامج حتى التعليمة التي وضعت بعدها نقطة التوقف مباشرة، ثم يوقف التنفيذ لفحص محتويات المسجلات والذاكرة إلى عند هذه النقطة.

6- المحاكي الحقيقي Emulator

يمثل المحاكي Emulator مزيجاً من المكونات الصلبة واللينة لفحص وتنقيح المكونات الصلبة واللينة لنظام خارجي. إن جزءاً من المكونات الصلبة للمحاكي هي عبارة عن كابل متعدد الأسلاك Multi wire cable مهمته توصيل النظام المضيف Host إلى المنظومة المراد تطويرها. يغرس القابس Plug الموجود عند نهاية الكبل في مكان المعالج الصغري لمنظومة النموذج الأولي Prototype ، ومن خلال هذا الوصل تسمح المكونات اللينة للمحاكي بتحميل الشفرة الغرضية للبرنامج ضمن الذاكرة RAM في المنظومة المراد فحصها ، ومن ثم يتم تنفيذ هذه الشفرة. يسمح المحاكي بشكل مشابه للمنقح بتحميل البرامج وتنفيذها وفحص محتويات المسجلات وتبديلها وفحص محتويات المواقع الذاكرية وتبديلها وغرس نقاط التوقف في البرنامج. ويستطيع المحاكي أيضاً أن يأخذ مستخلصاً Snapshot لمحتويات المسجلات ويبيّن الفعاليات الموجودة على ممر العناوين والمعطيات، وكذلك يبيّن حالة المؤشرات عند تنفيذ كل تعليمة. ويوجد ميزة قوية للمحاكي تتمثل بقدرته على استخدام ذاكرة النظام أو الذاكرة الموجودة على منظومة النموذج الأولي من أجل البرنامج الذي يتم تنقيحه. إذا المحاكي يمثل طريقة فعّالة لتنفيذ البرنامج وفحص المكونات الصلبة واللينة.

١٨-٦ معالجة المعطيات

للتعامل مع المعطيات وبخاصة الحرفية منها لابد من تعرّف آخر مجموعة من مجموعات تعليمات المعالج 8086 التي لم يتم التطرق لها بعد وهي تعليمات السلاسل.

١٨-٦-١ تعليمات السلاسل

وتضم مجموعة تعليمات السلاسل:

- البادئة REP, REPE, REPZ, REPNE, REPNZ تكرار تعليمة سلسلة حتى توجد شروط محددة، حيث تكتب البادئة REP قبل إحدى تعليمات السلسة، وتسبب هذه البادئة إنقاص المسجل CX وتكرار تعليمة السلسلة حتى يصبح CX = 0. وتعدّ الرموز REPE, REPZ و REPNE, REPNZ اختصارات لنفس البادئة، وترمز إلى تكرار إذا

كان يساوي" و "تكرار إذا كان صفرا" وهكذا.
غالبا ما تستخدم هذه البوادئ مع تعليمة مقارنة السلسلة بالصيغة REP COMPSB أو مع تعليمة مسح السلسلة بالصيغة REP SCASB ويصبح شرط تكرار السلسلة مع بعض البوادئ مزدوجا، بمعنى أن بادئة مثل REPE COMPSB تعني تكرار تعليمة السلسلة طالما أن الكلمات المقارنة متساوية أي ZF = 1 و CX لم يصل بعد إلى 0.

- التعليمات **CMPS/CMPSB/CMPSW**: هي لمقارنة بايتات أو كلمات السلسلة، حيث يستخدم SI لحفظ إزاحة البايت او الكلمة في سلسلة المصدر، ويستخدم DI لحفظ إزاحة البايت او الكلمة في السلسلة الأخرى، وتحدث المقارنة بطرح ما هو مشار إليه بـ DI مما مشار إليه في SI. تتأثر المؤشرات بعملية المقارنة إلا أن المعاملات لاتتأثر. وبعد تنفيذ عملية المقارنة يتزايد أو يتناقص محتوى SI و DI بشكل آلي ليشيرا إلى العناصر التالية في السلسلتين. يجب أن توجد السلسلة المشار إليها بوساطة DI في القطعة الإضافية ES ويفترض أن تكون السلسلة المشار إليها بوساطة SI في الـ DS.

- التعليمات **SCAS/SCASB/SCASW**: وهي لمسح بايت سلسلة أو كلمة سلسلة Scan string byte or string word حيث تقارن التعليمة SCAS بايت سلسلة مع بايت موجود في AL، أو تقارن كلمة سلسلة مع كلمة موجودة في AX. تؤثر هذه التعليمة بالمؤشرات ولكنها لاتغير المعامل في (AX) AL ولا المعامل في السلسلة. ويجب أن توجد السلسلة المطلوب مسحها في القطعة الإضافية، كما يجب أن يحتوي DI على إزاحة البايت أو الكلمة المطلوب مقارنتها.
بعد تنفيذ التعليمة SCAS سيتزايد DI أو يتناقص بشكل آلي (حيث من أجل سلاسل البايتات، يتزايد DI أو يتناقص بمقدار 1، أما من أجل سلاسل الكلمات فسوف يتزايد أو يتناقص بمقدار 2). إذا كان مؤشر الاتجاه صفرا ، فإن DI سيزداد بعد التعليمة SCAS (وبالعكس، عندما DF = 1 فان DI سيتناقص). تستخدم التعليمة SCAS غالبا مع بادئة التكرار REP وذلك لإيجاد الحدوث الأول لبايت محدد أو لكلمة محددة في السلسلة.

ملاحظة: يتكرر المسح Scanning طالما أن البايتات غير متساوية ولم تنته السلسلة بعد. وإذا تم العثور على الشفرة 0DH لمفتاح الإدخال، فإن ZF يرفع بواحد وسيشير DI إلى البايت التالي في السلسلة بعد هذه الشيفرة. وإذا لم توجد شيفرة مفتاح الإدخال يكون CX = 0 و ZF = 0 .

يستخدم المجمع اسم السلسلة ليحدد ما إذا كانت السلسلة من نوع Byte أو WORD.

ويمكن بدلا من استخدام اسم السلسلة إعلام المجمع مباشرة بنوع السلسلة باستخدام الاختصار SCASB من أجل سلسلة البايت و SCASW من أجل سلسلة الكلمات.

ويمكن بأسلوب مماثل دراسة تعليمات: – النقل MOVS/MOVSB/MOVSW

– الإدخال INS/INSB/INSW

– الإخراج OUTS/OUTSB/OUTSW

– التحميل LODS/LODSB/LODSW

– التخزين STOS/STOSB/STOSW

مثال: أكتب البرنامج الذي يجد العلامة الأعلى لطالب من بين خمس علامات وخزنها بالمسجل DH.

```
DATA DB 51, 44, 99, 88, 80 ; 13h, 2ch,63h,58h,50h
        MOV CX,5 ;set up loop counter
        MOV BX, OFFSET DATA ;BX points to GRADE data
        SUB AL,AL ;AL holds highest grade found so far
AGAIN:  CMP AL,[BX] ;compare next grade to highest
        JA NEXT ;jump if AL still highest
        MOV AL,[BX] ;else AL holds new highest
NEXT:   INC BX ;point to next grade
        LOOP AGAIN ;continue search
        MOV DH, AL
```

١٨-٦-٢ التحويل بين أشكال مختلفة للمعطيات

من المعلوم أن الحاسوب يستطيع إنجاز العمليات الحسابية بفعالية أكبر إذا كانت المعطيات بالصيغة الثنائية Binary Format، ولكن في العديد من الأحيان لابد من إدخال معطيات من لوحة المفاتيح Keyboard على شكل محارف ASCII أو إظهارها على الشاشة بصيغة ASCII أيضاً.

وبما أن أساس الصيغة الثنائية للأعداد هو 2 وأساس صيغة ASCII هو 10 فإن المعالجة الحسابية لمعطيات بالـ ASCII تختلف عن معالجتها بالـ Binary. فمثلا العدد 23 يمكن كتابته بالصيغة الثنائية على الشكل $_2$(0001 0111) أو 17H في حين أن كل محرف ASCII يحتاج إلى بايت كامل، ولذلك يكتب العدد 23 بصيغة الـ ASCII بالشكل 3233 عوضا عن 17H. أي لابد من الإلمام بعملية تحويل الأعداد من صيغة ASCII إلى صيغة Binary وبالعكس.

تحتل كافة المعطيات التي يتم إدخالها من لوحة المفاتيح مواقعها في الذاكرة وفق تمثيل ASCII مثل المحارف SAM تخزن على النحو التالي 53414DH، بينما العدد 1234 يخزن على النحو 31323334H.

من طرف آخر تبقى في معظم الأحيان المعطيات الأبجدية المدخلة كاسم شخص ما دون تغير في البرنامج، في حين أن العمليات الحسابية على المعطيات العددية تحتاج لمعاملة خاصة لتبقى في صيغة ASCII صحيحة.

تساعد التعليمات AAA, AAD, AAM, AAS في إنجاز العمليات الحسابية على معطيات ASCII العددية. تعمل هذه التعليمات بشكل آلي على تعديل قيمة المسجل AX، حيث إن المعالج يقوم بعملياته الحسابية على الأعداد بالـ ASCII كما لو أنها ذات الأساس 2 في حين ان شيفرة ASCII الثنائية لاي رقم هي عبارة عن عدد محدد من فئة ذات الأساس 10 غير المضغوطة Unpaked. ولتوضيح هذا المفهوم الجديد، نشير إلى أن الأعداد العشرية تمثل ثنائيا بطريقتين:

- الأولى: من خلال تمثيل كل رقم عشري بأربع خانات ثنائية مثل تمثيل العدد 513 بالشكل 0011 0001 0101 وتسمى عندئذ أعداد عشرية مضغوطة Packed.

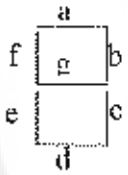
– الثانية: من خلال تمثيل كل رقم عشري ببايت واحد تحجز خاناته الأربع الدنيا لتمثيل الرقم وتترك الخانات الأربع العليا مساوية للصفر، فالعدد السابق يمثل بالعدد الثنائي 0000 0101 0000 0001 0000 0011 وتسمى عندئذ أعداد عشرية غير مضغوطة Unpacked.

واعتمادا على ذلك يمكن اعتبار شيفرة ASCII من 30H وحتى 39H أعداد عشرية يمكن تمثيلها ثنائيا بأعداد عشرية مضغوطة مثل 35H الذي يقابل 0101 0011.

ننتقل إلى مفهوم تحويل المعطيات من ASCII إلى الصيغة الثنائية، حيث إن معظم العمليات الحسابية على معطيات بشيفرة ASCII أو BCD تبقى مناسبة فقط عندما تكون اطوالها صغيرة. ومن المناسب تحويلها إلى الثنائي عندما تكون كبيرة.

مثال: اكتب برنامج 8086 ALP لتحويل أي رقم بالـ BCD إلى 7's Segment.

بداية نشير إلى الـ 7's Segment، رمز كل مقطع والكود المقابل لكل رقم وفق ما هو مبين أدناه، بحيث إن عنصر المقاطع السبعة يستخدم دخل بالمستوى 1 (High) لإضاءة المقطع.



0	g	f	e	d	c	b	a

9	8	7	6	5	4	3	2	1	0
6F	7F	07	7D	6D	66	4F	5B	06	3F

يخزن الجدول المبين بالشكل أعلاه بذاكرة المعطيات (Data Segment) ويكون البرنامج المطلوب وفق التالي:

```
.MODEL   SMALL
.DATA
Table  DB  3F, 06, 5B, 4F, 66, 6D, 7D, 07, 7F, 6FH ; 0...9
```

```
.CODE
START:    MOV  AX, @DATA  ; initialize
          MOV  DS, AX  ; Data Segment
          MOV  AL, 08H  ; load AL with any BCD digit, for
          example 8, to be converted to seven segment code
          MOV  BX, OFFSET  Table ; load BX with the
                    Offset of starting address of lookup table
          XLAT  Table  ; copy byte from address pointed by
                    ; [BX + AL] back into AL
          MOV  AH, 4CH  ; exit to DOS
          INT  21H
          END  START    END
```

مثال: Display on the screen two digit number $(85 \in (10 \div 99))$

```
;The Principe: (85)10 ~ 85/10 → result = 8 & reminder = 5
        MOV   AL , 85H
        CBW
        MOV   CL , 10D
        DIV     CL
        MOV   DL , AL
        ADD   DL , 30H
        MOV   CH , AH
        MOV   AH , 2H
        INT     21H
        MOV   DL , CH
```

```
MOV    DL , 30H
MOV    AH , 2H
INT    21H
```

واذا طلب على سبيل المثال إظهار عدد مكون من ثلاثة أرقام (أو أكثر) أي ضمن المجال (100 ÷ 999) مثل العدد 528 مثلاً فإن المبدأ نفسه، أي نقوم بالمرحلة الأولى بتقسيم العدد المعطى على 100 وبالمرحلة الثانية تقسيم الناتج على 10.

مثال: أكتب برنامج 8086 ALP لترتيب مجموعة أعداد بشكل تصاعدي.

ALGORITHM:

- Move the content to CX, decrement the count by one
- Move start address to BX
- Move CX to DI
- Move BX pointed data to AX
- Compare AX with next data
- If borrow is set, go to step 9
- Exchange AX and BX+2 value
- Move accumulator value to BX
- Add 2 to BX
- Decrement DI
- If DI not equal to zero go to step 4
- Decrement CX. If CX not equal to zero go to step 2

PROGRAM:

```
CODE SEGMENT
        ASSUME  CS: CODE, DS: CODE
        ORG  1000H
```

```
        MOV  CX, 0005H
        DEC CX
L3      MOV  BX, 1100H
        MOV  DI, CX
L2      MOV AX, [BX]
        CMP  AX, [BX + 2]
        JB  L1
        XCHG AL, [BX + 2]
        MOV [BX], AX
L1      ADD BX, 0002
        DEC  DI
        JNZ  L2
        LOOP  L3
        HLT
CODE ENDS
        END
```

مثال: اكتب برنامج بلغة الأسمبلي لجمع مصفوفتين كل منهما 3x3# بفرض أن كل من المصفوفتين مخزن بصيغة قائمة List فيكون البرنامج كالتالي:

```
; Title Addition of 3x3 Matrix
.MODEL    Small
. DATA
        M1  DB  10H, 20H, 30H, 40H, 50H, 60H, 70H, 80H, 90H
        M1  DB  10H, 20H, 30H, 40H, 50H, 60H, 70H, 80H, 90H
        Result    DW   9  DUP(0)
```

```
. CODE
Atart:  MOV AX, @DATA ; loads the address of data segment
        MOV  DS, AX       ; in [DS]
        MOV  CX, 9H   ; initialize the counter
        MOV  DI, OFFSET m1 ; initialize the pointer to matrix1
        MOV  BX, OFFSET m2 ; initialize the pointer to matrix2
Back:   MOV  AH, 00H  ; make MSB of result zero
        MOV  AL, [DI]  ; get the number from matrix1
        ADD  AL, [BX]  ; get the number from matrix2 and add
                       ; it in corresponding number of matrix1
        ADC  AH, 00H  ; save the carry of addition in MSB

        MOV  [SI], AX  ; store the result in corresponding position
                       ; of reseltant matrix
        INC  DI  ; increment pointer to matrix1
        INC  BX  ; increment pointer to matrix2
        INC  SI  ; [ increment pointer to resultant matrix]
        INC  SI
        LOOP  Back  ; repeat process for all matrix element
        MOV  AH, 4CH  ; [exit to DOS]
        INT  21H
        END  Start
             END
```

١٨-٧ أسئلة متعددة الاجابات

اختبر معلوماتك، باختيار الجواب الصحيح لكل من التساؤلات التالية:

١- أي من الصيغ التالية غير صحيحة؟

a. ADD [BX] , AX
b. ADD AL , [200H]
c. ADD [220H] , AX
d. ADD [BX], [2000H]

٢- يمكن تنفيذ عملية الضرب على أعداد مؤشرة بواسطة

a. Repeated integer addition
b. MUL instruction
c. IMUL instruction
d. None of the above

٣- الـ CBW هي التعليمة التي يمكن بواسطتها اجراء

a. convert the word stored in register Ax and store the result in AL
b. the 8-bits signed no. in AL is converted to 16-bits signed in AX
c. convert unsigned byte in AL to unsigned word to AX
d. convert the word stored in register AL and store the result in AX

٤- الـ NEG هي التعليمة التي يرد معها المعامل التالي

a. register
b. memory
c. number
d. a and b

٥- المؤشر الذي لايتأثر بالتعليمة "INC operand" هو التالي:

a. carry flag
b. zero flag
c. overflow flag
d. sign flag

٦- تنفيذ التعليمة SBB AH,AL يؤدي إلى

a- AH –AL –CARRY
b- AL --AH
c- AL – AH –CARRY
d- AH-AL

٧- عند اجراء عملية قسمة بتعليمة من الشكل IDIV BL يكون:

a. BH الناتج في BL والباقي في
b. BL الناتج في BH والباقي في
a. AH الناتج في AL والباقي في
b. AL الناتج في AH والباقي في

٨- الرقم من "2" وظائف DOS المتاحة تحت المقاطعة "21H" يعني:

a. keyboard input
b. output on the display screen
c. display string
d. return to DOS

٩- أرقام كافة الوظائف المتاحة تحت المقاطعة "21H" تخزن في:

a- AH
b. DL
c. DX
d. AL

١٠- شيفرة الرمز الذي يرسل بالوظيفة "5" تحت المقاطعة "21H" للطباعة يخزن بالمسجل

a. DL
b. AH
c. AL
d. DX

الجزء الرابع: الأنظمة المدمجة

ويضم فصلان

الفصل التاسع عشر: المتحكمات الصغرية

الفصل العشرون: برمجة المتحكمات

Damascus University

وقل رب زدني علما
جامعة دمشق
Damascus University

المنظومة المدمجة هي منظومة تعتمد متحكماً ميكروياً Microcontroller، ومقادة بالبرمجيات، وذات وثوقية، وهي منظومة للتحكم في الزمن الحقيقي، تعمل في أوساط مختلفة وتباع بأسعار اقتصادية منافسة، وظيفتها الأساسية ليست حسابية بل متحكم فيها باستخدام حاسوب مضمَّن ضمنها.

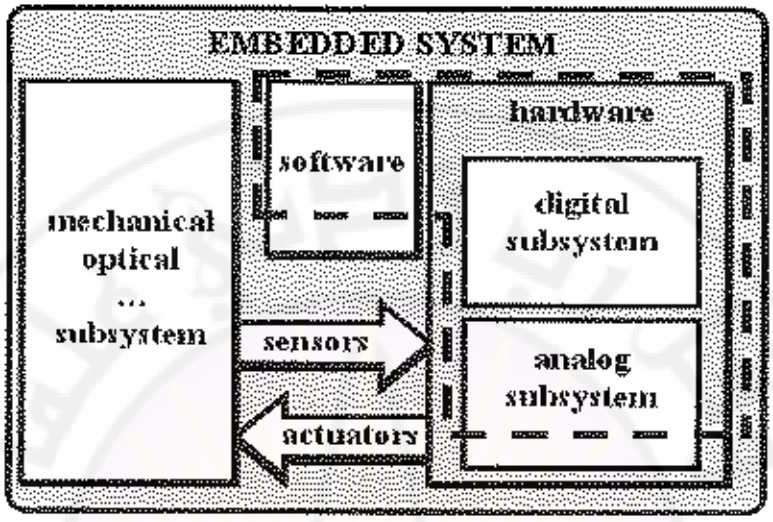
وتعني كلمة مضمَّن أنها موجودة ضمن المنظومة مشكلة جزءاً متكاملاً من المنظومة الأكبر. بالتالي يمكن النظر إلى كل نظام مدمج على أنه كل نظام يخدم نظاماً أكبر منه عن طريق تقديم بعض الوظائف.

تسمى الأنظمة المدمجة (أو المضمّنة) أيضاً بالأنظمة المدفونة أو الأنظمة المخبأة، وكلها تعابير تشير إلى نفس المعنى الذي يقابل المصطلح Embedded Systems. ومن الناحية العلمية يمكن تعريف الأنظمة المدمجة بأنها أنظمة الكترونية تحتوي على دوائر الكترونية وبرمجيات تعمل من خلالها على تنفيذ التطبيق المطلوب كمعالجة البيانات واتخاذ القرار المناسب في الوقت المطلوب.

توجد الأنظمة المدمجة في العديد من الأجهزة، وذلك لسرعتها ودقة لأدائها وسهولة تنفيذها نسبيا بالمقارنة بالأجهزة الالكترونية التقليدية، فنصادفها بكثير من التجهيزات الطبية والأجهزة الصناعية وأجهزة الملاحة والتحكم بالطائرات والسفن والسيارات وآلات الصرف الآلي ATM في المصارف، وأنظمة قيادة الصواريخ والطابعات الحاسوبية والراسمات ومعظم الأجهزة المنزلية من تلفزيون وفيديو وحتى الغسالات والثلاجات. إن وجود البرمجيات بالأنظمة المدمجة يجعلها أكثر مرونة وملائمة للتطبيق وبالتالي أكثر فاعلية وتوفيراً للطاقة، وأسهل في تحديد الأعطال والصيانة. يبين الشكل IV-1 التركيبة النموذجية للمنظومة المدمجة التي تضم جملة من الوحدات المبيتة بشريحة واحدة أو كرت واحد وهي:

- Processor or Microcontroller
- Memory
- Input/output interface chip
- I/O devices including Sensors and Actuators
- A/D and D/A Converters

- Software as Operating System
- Application Software



الشكل I-IV: التركيبة العامة للمنظومة المدمجة

ومثلما يمثل المعالج الصغري Microprocessor العنصر الأساسي بتركيبة الحاسوب الصغري Microcomputers، فإن المتحكم الصغري Microcontroller يمثل العنصر الأساسي بتركيبة النظام المدمج Embedded System.

كانت أول منظومة مضمّنة هي حاسوب التحكم في قيادة مركبة الفضاء الأمريكية أبولو، وقد تضمنت كل رحلة إلى القمر اثنين منها. تطورت المنظومات المضمّنة منذ ذلك التاريخ وحتى اليوم وتزايدت قدرتها على المعالجة، وانخفضت كلفة إنتاجها.

كان المعالج 4004 الذي أنتجته شركة إنتل Intel أول المعالجات الصغرية الذي أخذ طريقه إلى استخدامات عديدة في الآلات الحاسبة ومنظومات صغيرة أخرى، وكان يتطلب بعض المكونات الأخرى كالذاكرة ووحدات الإدخال والإخراج. ومع نهاية الثمانينيات وُضع المعالج وجميع المكونات الخارجية الضرورية في دارة متكاملة واحدة مما أدى إلى ظهورر وتطور ما سُمي بالمتحكم المايكروي الذي شاع استخدامه في المنظومات المضمّنة.

مقومات المنظومة المضمّنة

تتكون المنظومات المضمّنة مثل كل المنظومات الحاسوبية، من عتاد وبرمجيات، وحتى وقت قريب كان تصميم العتاد الذي يتركز على المعالج الصغري وربطه بوحدة الذاكرة والبوابات، يأخذ الوقت والجهد الأكبر من التصميم. وكان يتم بعد الانتهاء من تصميم المنظومة الصغرية كتابة برنامج بسيط باستخدام لغة الآلة يتناسب مع حجم الذاكرة ووسائل التطوير المتاحة. وبعد التطور الكبير الذي حصل في تقانات الدارات المتكاملة صار يوضع على كسرة واحدة كامل المنظومة السابقة في صيغة متحكم مايكروي. وهكذا صار تصميم العتاد بسيطاً، وتوجّه التركيز إلى كيفية استخدام لغات البرمجة المتقدمة العالية المستوى وأدواتها في كتابة برمجيات تخدم تطبيقات أكثر تعقيداً.

- الدقة الزمنية

يجب أن يستجيب المتحكم المايكروي بسرعة ليبقى تنفيذ العملية في منطقة الأمان، وهذا يُعدّ من أهم سمات العمل في الزمن الحقيقي Real time، إذ يستجيب المتحكم لتغير الدخل ويحقق الاستجابة اللازمة ضمن الإطار الزمني المحدد. وتختلف طريقة العمل هذه عن عمل الحواسيب الشخصية كثيراً، إذ يمكن الانتظار على الحاسوب الشخصي إذا تأخر في تنفيذ عمل معين كتحديث ملف ماء ولكن لا يمكن التأخر في استجابة منظومة مكابح السيارة عند إعطاء الأمر بذلك. وهكذا فإن بعض النظم المضمّنة تعمل ضمن شروط عمل قاسية من حيث الزمن، وتحقق جميعها الدقة الزمنية.

- ترابطات المنظومة

قد تتطلب بعض المنظومات أكثر من متحكم واحد لتنفذ عملها على نحو صحيح وسريع، ويُكلَّف كل متحكم مضمّن التحكم بمنظومة جزئية. لهذا يجب أن يكون ممكناً ربط النظام المضمّن بمنظومات أخرى. ويتم تبادل المعلومات المشتركة بين هذه المتحكمات المضمّنة عبر شبكة بسيطة لتبادل المعلومات تصمم لتناسب متطلبات النظام المتحكم فيه. أصبح هذا المنهج أكثر شيوعاً لأن كلفة عدد من المتحكمات وشبكة ترابطها أقل بكثير من حاسوب وحيد.

- الوثوقية

يجب أن تكون البرمجيات المطوّرة للنظم المضمّنة ذات وثوقية عالية، وألا تحوي أي أخطاء تصميمية. وعلى عكس البرمجيات التي تُطورها الشركات للحواسيب الشخصية، فإنه لا يمكن أن تحوي النسخة الأولى أي أخطاء على أمل تصحيحها في النسخ اللاحقة، إذ إن أي خطأ قد يؤدي إلى كوارث لا تُحمد عقباها، ويعقد ذلك عمليات التصميم والاختبار والتصحيح لكل من العتاد والبرمجيات الخاصة بالنظم المضمّنة.

مكوّنات المتحكم المضمّن

يتألف كل متحكم مضمّن من المكونات الأساسية التالية:

- وحدة المعالجة المركزية

وهي عادة معالج صغري مناسب للتطبيق، فقد يكون ذا 8 ،16 أو 32 bit . كما يُزود عادة بمجموعة تعليمات مناسبة للتطبيق. وقد يكون هذا المعالج من المعالجات ذات مجموعة التعليمات المعقدة CISC (أو RISC)، وله إمكانات عدونة الذاكرة والبوابات.

- وحدة الذاكرة

هي تجهيزه إلكترونية لها دارات قراءة وكتابة، ويمكن حفظ كمية معلومات كبيرة فيها ضمن حيز فيزيائي صغير، وتحتفظ بالمعلومات من دون تغيير، وهي ذات استهلاك منخفض نسبياً للطاقة الكهربائية، وذات وثوقية واستمرارية عالية. وتصنّف وفق إمكانية القراءة والكتابة في ذواكر قراءة فقط (ROM)، أو ذواكر قراءة وكتابة (RWM). قد تكون الذواكر ذات النفاذ العشوائي من نوع ديناميكي Dynamic RAM، أو من النوع الساكن Static RAM ويجب على المصمم عند إنجاز المنظومة المضمّنة أن يقرر مكان توضع الذاكرة، هل يُضمّنها على رقاقة المعالج أم يضعها خارجها.

- وحدات الإدخال والإخراج

وهي وحدات مهمتها ربط المعالج بالوسط الخارجي؛ مهمتها إدخال المُعطيات المطلوب معالجتها إلى المعالج وإخراج نتائج المعالجة من المعالج إلى الوحدات المحيطية الأخرى. قد تكون المعطيات المُدخَلة إلى المعالج رقمية فتُستخدم وحدات إدخال رقمية لهذا الغرض.

أما إذا كانت المعطيات المُدخلة تمثل بإشارة تمثيلية فلابد من استخدام وحدات إدخال تمثيلية. يمكن قول الشيء ذاته عن وحدات الإخراج أيضاً.

عند استعمال وحدات تمثيلية لإدخال إشارات تمثيلية ومعالجتها في المعالج، لا بد من تحويل الإشارة إلى ما يكافئها رقمياً قبل إدخالها إلى المعالج، ويستخدم لهذا الغرض مبدل تمثيلي رقمي (ADC). أما عند إخراج نتائج المعالجة الرقمية من المعالج إلى وحدة تمثيلية فلا بد من استخدام مبدل (DAC).

وأخيراً قد تكون وحدات الإدخال والإخراج تسلسلية Serial أو تفرعيه Parallel، ومتزامنة Synchronous أو غير متزامنة Asynchronous. ولكل طريقة معايير وبروتوكولات طُوّرت لمثل هذه التطبيقات.

برمجيات المنظومات المضمّنة

يستخدم مصممو المنظومات المضمّنة برمجيات عديدة كالمترجمات Compilers والمنقحات Debuggers لتطوير البرمجيات التي تتحكم في المنظومات المضمّنة. كما قد يستخدمون الأدوات الآتية:

- مقلد في الدارة in-circuit emulator: هو تجهيزه عتادية – برمجية تقوم مقام المعالج وتوفر التسهيلات لتحميل الرماز التجريبي للبرنامج التطبيقي وتنقيحه.
- تسهيلات لزيادة وثوقية المنظومة بإضافة إحدى طرق الترميز coding مثل اختبار المجموع أو CRC للبرنامج، بحيث تستطيع المنظومة تبيان ما إذا كان رماز البرنامج صحيحاً أم قد ترافق بخطأ ما.
- قد تستخدم مترجمات compilers وواصلات linkers لتحسين صياغة البرنامج وتحريره لتطبيق محدّد.
- برمجيات الإقلاع: تستخدم معظم المنظومات المضمّنة برمجيات محددة لإقلاعها. تقوم هذه البرمجيات باختبار المكونات الالكترونية وتنجز اختباراً ذاتياً self-test قبل إقلاع البرنامج التطبيقي.

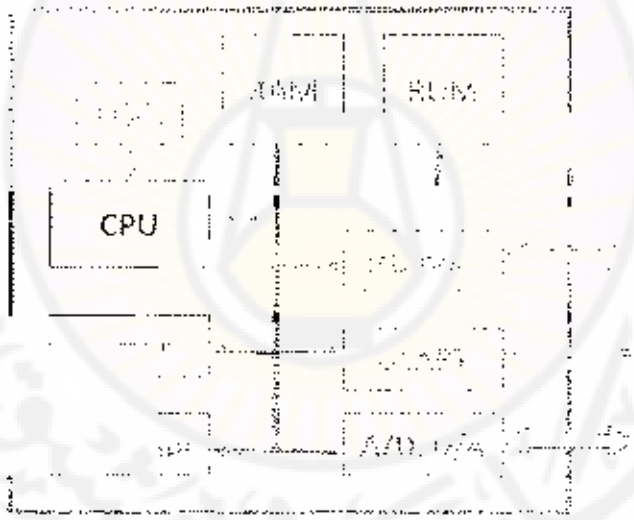
متطلبات التغذية الكهربائية:

تتطلب المتحكمات المضمّنة تغذية كهربائية تُمكّنها من القيام بمهماتها. ويكون عادة الخيار بين استخدام مصدر التغذية العامة أو البطاريات أو مصادر متجددة. ويجب على مصمم النظام أولا تقدير قيمة الطاقة اللازمة وتحديد مصدرها وكيفية توصيلها إلى النظام لضمان الاستمرارية والأمن والسلامة وعدم إعاقتها قيام المنظومة بما حُدّد لها؛ مثل إمكانية كونها ثقالة في بعض الأحيان. فإذا كان الخيار استخدام بطاريات، فعلى المصمم تحديد نوعها وسعتها وسعرها ومعدل إعادة شحنها.

أما إذا كان الخيار استخدام الشبكة العامة فيتوجب استخدام منظم جهد ومكيف الاستطاعة للحصول على تغذية نظيفة وتحديد طريقة التغذية لضمان الاستخدام الأمن لها واستمرارية التغذية.

يمثل المتحكم الصغري بحد ذاته منظومة مدمجة يجمع مابين الـ Hardware والـ Software في نظام متكامل، يمكن برمجته بحسب الطلب لأداء عمليات رياضية بسيطة، مراقبة المخارج، وإظهار المداخل الرقمية.

تتميز المتحكمات الصغرية(التي تدعى بالمتحكمات المايكروية أو المتحكمات المخصصة) عن المعالجات الصغرية العادية بأنها تمتلك بالإضافة لوحدة المعالجة المركزية CPU، ذاكرة (بنوعيها ROM و RAM)، وبوابات إدخال وإخراج، وبعض الدارات الأخرى مثل Timer/Counter ودارات للنقل التسلسلي وأخرى لتحويل الإشارة ومنظم للمقاطعة وغيرها على نفس الشريحة. يظهر الشكل 1-19 المخطط الصندوقي للتركيبة العامة للمتحكم الصغري.



الشكل1-19: التركيبة العامة للمتحكم الصغري

يستطيع المتحكم الصغري مثل المعالج جلب إشارات الحساسات والمفاتيح عن طريق المداخل ويقوم بمعالجتها عن طريق البرنامج المخزن في الذاكرة ROM، ومن ثم يقوم بإخراج الأوامر على المخارج التي تكون موصولة إلى المشغلات التي نريد التحكم بها.

يمثل المتحكم الصغري Microcontroller كعنصر رئيسي في المنظومة المدمجة ميكروكمبيوتر مصغراً بدارة واحدة.

صمم أول متحكم صغري بالعام 1971 من قبل Smithsonian institution بالاسم TMS1000 وبعد ذلك أنتجت شركة Intel المتحكم 8048 في العام 1976 ثم تبع هذه الشركات العديد من مصنعي المتحكمات، لدرجة أنه أصبح يوجد حاليا العديد من المتحكمات الحديثة التي تمتاز بسرعات عالية وإمكانيات متعددة. ومن أهم الفوارق الأساسية بين كل من المتحكم والمعالج الصغري نذكر :

١- يتعامل المتحكم مع الأنظمة بالزمن الحقيقي على عكس المعالج بالحاسوب الذي ليس بالضرورة أن يتعامل في معظم الأحيان مع مثل هذه الأنظمة.

٢- مصدر الأوامر أو مكان وجود التعليمات.

ففي المعالج تتواجد التعليمات والأوامر في ذاكرة خارج شريحة المعالج، وبالتالي يمكن أن تكون التعليمات طويلة ومعقدة جداً بالمقارنة مع المتحكم التي توجد تعليماته ضمن ذاكرة خاصة تختلف باختلاف نوع المتحكم، موجودة داخل الشريحة. يمكن أن تكون التعليمات في المتحكم أقل وأبسط لان المتحكم محدد الوظيفة، ومن هنا أتت تسميته بمتحكم، حيث لا يحتاج إلى أوامر توجيه خارجية أثناء العمل، بعكس المعالج الذي سمي بهذا الاسم، لأنه يحتاج لأوامر خارجية لكي يقوم بعمله. ولاختيار المتحكم المناسب لتطبيق ما، يوجد عشر مراحل يجب إتباعها هي:

المرحلة الأولى: تشكيل قائمة بالمتطلبات التقنية للربط

Step1: Make a list of required hardware interfaces

حيث يحتاج المتحكم للربط مع الوسط الخارجي بعض المكونات التي يمكن تصنيفها بمجموعتين هما: وصلات التبادل للمعطيات ونوع الربط.

الثانية: فحص التركيبة البرمجية Examine the software architecture

الثالثة: اختيار المعمارية Select the architecture

الرابعة: تحديد متطلبات الذاكرة Identify Memory Needs

الخامسة: البحث عن المتحكمات Start searching for microcontrollers

السادسة: تخمين التكلفة وشروط التغذية — Examine Costs and Power Constraints

السابعة: فحص قسم من الاتاحية — Check part availability

الثامنة: اختيار عدة أو لوحة التطوير — Select a development kit

التاسعة: التحقيق في المترجم والأدوات — Investigate compilers and tools

العاشرة: البدء في التجريب — Start Experimenting

يوجد العديد من عائلات المتحكمات التي يمكن الاختيار منها، ومن أشهرها منتجات: Intel, PIC, AVR Family، يظهر الجدول أدناه بعض الفوارق الجوهرية بين متحكمات 8 bit لدى أكثر ثلاث شركات شهيرة على صعيد المتحكمات الصغرية.

الجدول 1-19: الفوارق الجوهرية بين متحكمات Intel, AVR, PIC من رتبة 8 bit

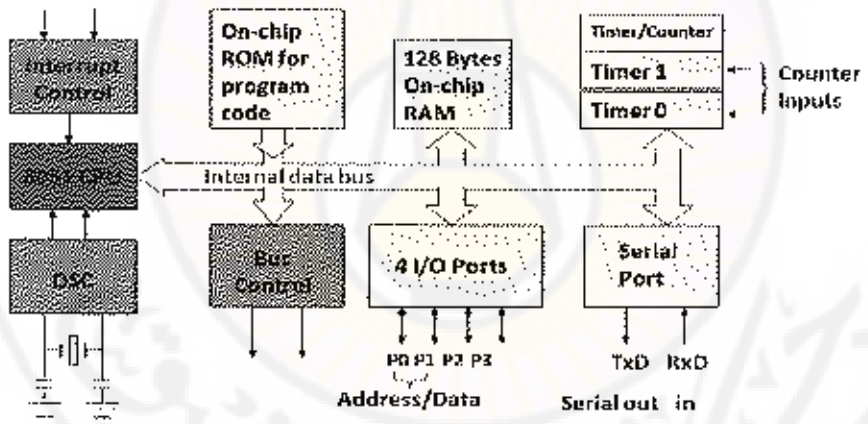
	Intel	PIC	AVR
Speed	Slow	Moderate	Fast
Memory	Small	Large	Large
Architecture	CISC	RISC	RISC
ADC	Not Present	Inbuilt	Inbuilt
Timers	Inbuilt	Inbuilt	Inbuilt
PWM Channels	Not Present	Inbuilt	Inbuilt

سنتعرف في الفقرات التالية بشيء من التفصيل على متحكم واحد بطول 8 bit من كل من هذه العوائل الثلاثة المذكورة كعينة عن بقية المتحكمات الأخرى المعروفة، حيث إن فهم تركيبة وآلية عمل أي متحكم صغري من أي منتج (شركة) لأي من التطبيقات البسيطة، يساعد على فهم بقية عمل المتحكمات من نفس الشركة أو غيرها.

١٩-١ المتحكم Intel 8051

أطلقت شركة Intel المتحكم الصغري 8051 بالعام 1980 من القرن الماضي(ويسمى (The Intel 8051 Embedded Controller Family بمعمارية Harvard وبجملة تعليمات CISC كمتحكم من شريحة واحدة (μC) للاستخدام في الأنظمة المدمجة. ينتمي المتحكم 8051 إلى فئة المتحكمات بطول كلمة 8 bits وبجملة تعليمات يصل إلى 250 تعليمة كل منها تأخذ من 1 وحتى 4 حلقات ساعية للتنفيذ وتصل سرعته إلى مليون تعليمة بالثانية.

يستخدم الرمز Intel MCS-51 للمتحكم 8051 مع دارات الدعم الخاصة به، وقد طورت العائلة MCS-51 الأساسية باستخدام التقانة NMOS ، ولكن بوقت متأخر ظهر المعالج 80C51 بالتقانة CMOS الذي يستهلك مقداراً أقل من الطاقة. يبين الشكل 19-2 المخطط الصندوقي الداخلي للمتحكم 8051 .



الشكل19-2: التركيبة الداخلية للمتحكم Intel 8051

ومن أهم ميزات المتحكم 8051 نذكر :

- 8 bit ALU
- 16 bit PC and DPTR
- 8 bit Stack Pointer and 8 bit PSW
- 4 KB Internal ROM
- 128 Bytes of Internal RAM

- 32 bit Arranged as four, 8 bit ports P0–P3
- Two 16 bit Timers/ Counters T0 & T1
- Full Duplex Serial Port
- Control Registers: TCON, TMOD, SCON, PCON, IP, IE
- Two External and Three Internal Interrupt Sources
- 0 –12 MHz Clock
- 40 Pin DIP Package

تربط بالممر الداخلية للمتحكم كل أجهزة الدعم لوحدة المعالجة المركزية لـ 8051 من ذاكرة البرنامج، البوابات، ذاكرة المعطيات، وحدة الربط التسلسلي، متحكم المقاطعة، والمؤقتات، وهي تتكون من 8 bit لممر المعطيات و16 bit للعناوين وممر لبقية إشارات التحكم مثل الإشارتين RxD, TxD التي توصل لـ Port 3. يمكن باختصار شرح كل من المكونات (الكتل) الأساسية للمتحكم 8051 المبينة بالشكل أعلاه كالتالي:

١ – ذاكرة 4 KB ROM

يملك المتحكم 8051 ذاكرة من نوع ROM بسعة 4 KB لتخزين البرنامج. ويمكننا وصل ذاكرة من هذا النوع بشكل خارجي تصل سعتها حتى 64 KB إذا كان التطبيق يحتاج لذلك. ويمكن حساب مجال عداد البرنامج PC انطلاقاً من أن السعة 4 KB التي تقابل 2^{12} Byte كما يلي.



أي أن مجمل المواضع أو مجال الـ PC هو من العنوان 0000H وحتى 0FFFH يخصص لتخزين البرامج.

٢- ذاكرة 128 Bytes RAM

يملك المتحكم 8051 ذاكرة من نوع RAM بسعة 128 Bytes لتخزين المعطيات. تحتوي RAM على بنوك المسجلات، المكدس للتخزين المؤقت للبيانات. كما أنها تحتوي على بعض المسجلات الوظيفية الخاصة Special Function Register اختصارا SFR التي تستخدم لبعض الأهداف الخاصة كمؤقت، بوابات دخل / خرج وغيرها. يملك المتحكم بشكل عادي 256 Bytes من ذاكرة RAM منها 128 Byte تستخدم لفضاء المستخدم والتي هي عادة مسجلات، بنوك ومكدس Stack والـ 128 Bytes الأخرى تخصص لـ SFRs. وبما أن 128 Bytes تقابل 2^7 Byte فإن مجال العنونة لهذه الذاكرة المخصصة لتخزين البيانات مبين بالشكل التالي:

	7	6	5	4	3	2	1	0	
If initial address is	0	0	0	0	0	0	0	0	(00H)
then final or last address is	0	1	1	1	1	1	1	1	(7FH)
		7			F				

أي أن مجمل الموضع هو من العنوان 00H وحتى 7FH ، وتسمى هذه الإجرائية بالحساب ' تخطيط الذاكرة Memory mapping '

٣- المؤقتات والعدادات Timers and Counters

المؤقت يعني أداة تستطيع إعطاء تأخير زمني محدد بين بعض الأحداث، مثل عملية إشعال ضوء أو إطفائه بعد كل S 2. هذا التأخير يمكن إجراؤه عبر برنامج بلغة التجميع، لكن المتحكم 8051 يملك تقنيا مأخذين لتوليد هذا التأخير. ويمكن استخدام هذين المأخذين لعّد بعض الأحداث الخارجية، كعدد مرات تكرار عدد ما معطى في جدول. هذان المأخذان بالمتحكم 8051 هما T0, T1 اللذان يعطيان التأخير فيما إذا تم استخدمهما بنمط أو بمود المؤقت Timer mode.

المؤقت المتاح هو بطول 16 bits وبالتالي يمكن توليد تأخير بين 0000H و FFFFH. والمسجلين الوظيفيين الخاصين المتاحين مبينين بالشكل أدناه



بمعنى إذا أردنا أن نملىء T0 بـ 16 bit يمكننا بشكل منفصل شحن الـ 8 bit السفلي بالقسم TL0 والـ 8 bit العليا بالقسم TH0. ونفس الشيء بالنسبة لـ T1. والمسجلان المستخدمان لمراقبة عملية المؤقت هما TMOD و TCON.

٤- البوابة التسلسلية Serial Port

يوجد لدى المتحكم 8051 مأخذان متاحان للنقل التسلسلي للمعطيات هما RxD و TxD. حيث إن TxD تستخدم بشكل عادي لإرسال المعطيات التسلسلية الموجودة في المسجل SBUF، بينما RxD يستخدم لاستقبال المعطيات التسلسلية. ويستخدم المسجل SCON لمراقبة العملية. حيث يوجد أربعة أنماط للنقل التسلسلي.

٥ – المذبذب Oscillator

يستخدم المذبذب لتأمين الساعة اللازمة لعمل المتحكم 8051 التي تحدد سرعة الإرسال للمتحكم الصغري. يمكن استخدام كريستال بتواتر متفاوت من4 MHz وحتى 30 MHz، ولكن بشكل عادي يتم استخدام كريستال بتواتر 11.0592 MHz.

٦- المقاطعات Interrupts

تعرّف المقاطعات على أنها طلبات من المتحكم لقطع تنفيذ برنامجه الأساسي والذهاب لتنفيذ برنامج آخر يخدم المقاطعة، ويمكن رفض طلب أي مقاطعة (أي تحجيبها) فيما إن لم تكن مستخدمة. يتم إتباع جملة من الأحداث أو الإجراءات لمعالجة المقاطعات. تعرّف هذه الإجراءات الخاصة بخدمة المقاطعة Interrupt Service Routines اختصارا ISR. ويملك المتحكم 8051 المأخذين INT0, INT1 للمقاطعات الخارجية.

٧- بوابات الدخل/ خرج Input output ports

يمتلك المتحكم 8051 أربعة بوابات للدخل والخرج هي P0, P1, P2, P3 كل منها بعرض 8 bit وله مسجل وظيفي خاص يعنون على مستوى البيت، بمعنى كل بيت يمكن

أن يوضع بحالة Set أو Reset بالتعليمات البتية SETB للوضع بالواحد وCLR للتصفير بشكل مستقل.

البوابة P0 : يمكن أن تقوم بعملين، إما لنقل المعطيات وإما لحمل القسم السفلي من ممر المعطيات والعناوين المركبة $AD_0 - AD_7$.

البوابة P1: بوابة نقل عامة، يمكن استخدامها لنقل بيانات بعرض 8 bit.

البوابة P2: يمكن استخدامها لنقل المعطيات كا I/O Ports ولنقل القسم العلوي من ممر العناوين $A_8 - A_{15}$.

البوابة P3: تستخدم أيضاً بوظيفتين الأولى كبوابة I/O والثانية لبعض الوظائف الخاصة، بحيث كل بيت منها يعني كما يلي:

P3.0 – RxD: for Serial I/P for Asynchronous communication,
or for Serial I/P for Asynchronous communication

p3.1 – TxD: Serial data transmit

P3.2 – INT0: External Interrupt 0

P3.3 – INT1: External Interrupt 1

P3.4 – T0: Clock input for counter 0.

P3.5 – T1: Clock input for counter 1.

P3.6 – WR: Signal for writing to external memory.

P3.7 – RD: Signal for reading from external memory

عندما تستخدم (توصل) الذاكرة الخارجية بالمتحكم 8051 لايمكن استخدام البوابتين P0 و P2 للعمل كبوابات دخل خرج، بل تعمل كممر للعناوين والمعطيات.

ينتج المتحكم 8051 الآن من شركات عدة بنسخ متعددة، وأصبح يعّد من أكثر المتحكمات شعبية.

طورت العائلة MCS-51 بالعام 1996 إلى 8 bit محسنة بالاسم MCS-151 والعائلة MCS-251 8/16/32 bit التي تمثل متحكمات متوافقة.

يشكل المتحكم 8051 من شركة إنتل مع متحكمات أخرى من نفس الشركة، مثل المتحكمات 8032, 8031, 8751, 8052 مجموعة مشتركة ببعض المواصفات، يبين الجدول 2-19 أهم الفوارق بينها:

الجدول2-19: المواصفات الأساسية لبعض المتحكمات الصغرية من إنتل

Device	Internal Memory		Timers/ Counters	Interrupts
	Program	Data		
8052	8Kx8 ROM	256x8 RAM	3x16 bit	6
8051	4Kx8 ROM	128x8 RAM	2x16 bit	5
8032	none	256x8 RAM	3x16 bit	6
8031	none	128x8 RAM	2x16 bit	5
8751	4Kx8 EPROM	128x8 RAM	2x16 bit	5

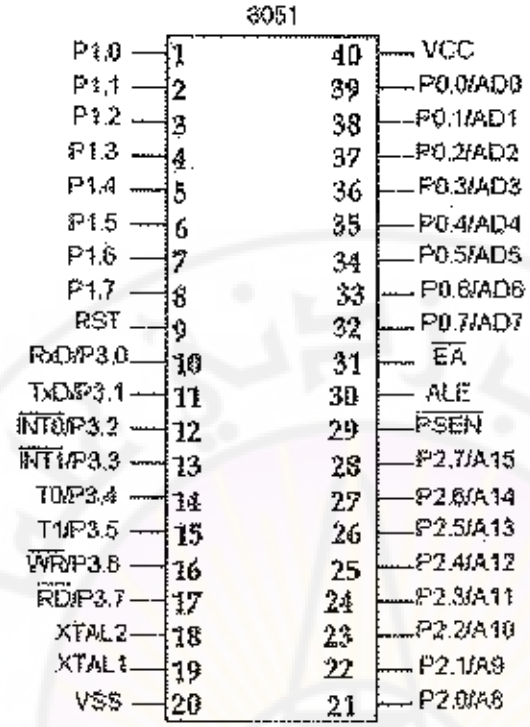
توصيف المتحكم 8051 يغطي كافة أفراد المجموعة المبينة أعلاه، مع فوارق بسيطة بين متحكم وآخر. يصنع المتحكم 8051 بالتغليف DIL-40 والشكل 3-19 يبين توزع إشاراته على كل من مآخذه. تشغل بوابات المتحكم 8051 الأربعة 32 مأخذاً، بالتالي يبقى ثمانية مآخذ تحمل إشارات مختلفة هي:

- مأخذين للتغذية Vcc (بقيمة 5V+) على المأخذ pin 40 والأرضي GND على المأخذ pin 20

- مأخذان XTAL1 and XTAL2 تقابل المآخذ pins 19,18 لتأمين الساعة اللازمة للمتحكم، حيث يوجد شكلان أو أسلوبان لتأمين الـ Clock للمتحكم هما:

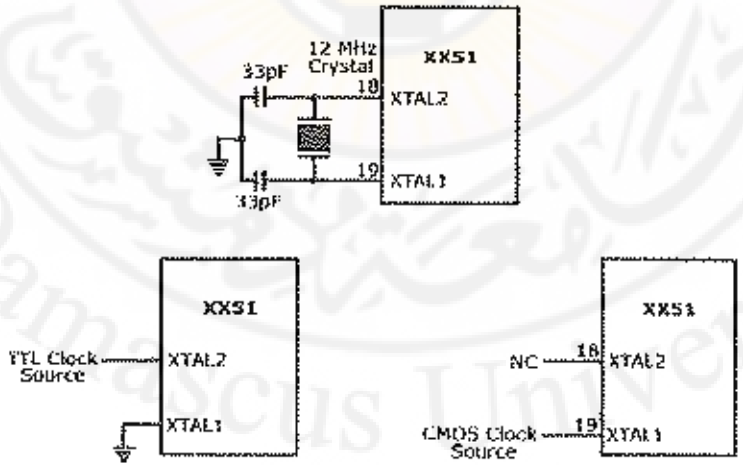
Way 1: using a quartz crystal oscillator

Way 2: using a TTL or CMOS oscillator



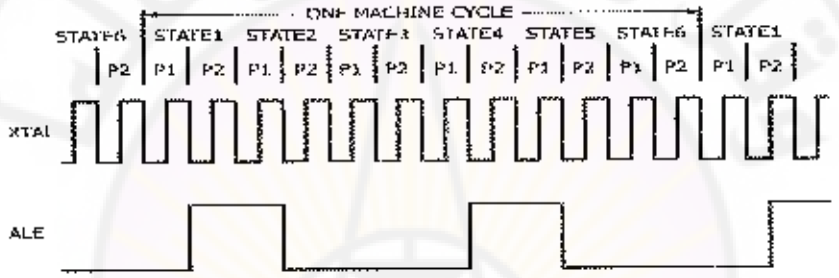
الشكل3-19: توزع مآخذ المتحكم Intel 8051 (8031)

يظهر الشكل 4-19 التوصيلات الممكنة لتأمين الساعة اللازمة، حيث يستخدم بالحالة العامة توصيل الكريستال لتأمين الساعة.



شكل4-19: الأساليب الممكنة لتأمين الساعة اللازمة للمتحكم 8051

يستخدم في بعض الحالات الخاصة منبع خارجي للساعة، بحيث يوصل المنبع إلى XTAL2 إذا كان من نوع TTL ويهذه الحالة يبقى XTAL1 غير موصول NC. وإذا كان المنبع CMOS يتم وصله بالعكس، أي يوصل المنبع إلى XTAL1 ويبقى XTAL2 غير موصول. حدود تواتر الساعة يمكن أن يتغير من جهاز Device لآخر. التردد القياسي هو 12 MHz أما إذا تم استخدام النقل التسلسلي يفضل استخدام التواتر 11.0592MHz. يبين الشكل أدناه ساعة النظام، أي موجة حلقة الآلة، بحيث إن تذبذب واحد كامل لمنبع الساعة يدعى نبضة Pulse. وكل نبضتين تشكل حالة State وكل ست حالات تشكل حلقة آلة.

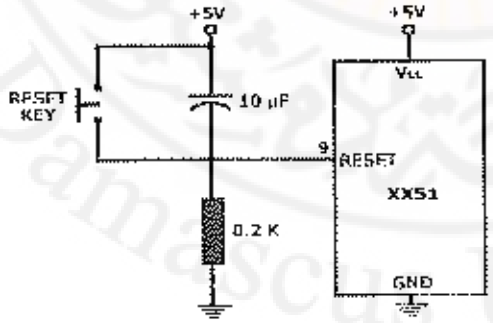


ويظهر بالشكل أعلاه نبضتان لـ ALE ضمن الحلقة الساعية الواحدة.

- مأخذ للتصفير Reset على الـ Pin9 وهو فعّال بالواحد، ويجب أن يدوم حلقتا آلة حتى يتم التصفير. يظهر الشكل 5-19 شكلين للتصفير هما:

الأول: عند وصل التغذية ويتم تلقائيا

الثاني: تصفير يدوي ويتم بحسب الطلب



الشكل5-19: دارة تصفير المتحكم 8051

التصفير لا يؤثر في محتوى الذاكرة الداخلية، بل يصفر محتوى عداد التعليمات ليبدأ من العنوان 00H، حيث يتم بشكل أساسي انتقاء بنك المسجلات صفر. ويضع مؤشر المكدس على العنوان 07H ويكتب بكل البوابات القيمة FFH.

– مأخذ للولوج الخارجي EA'/Vpp – External Access على الـ Pin31. بما أن كلاً من المتحكمين 8032 ,8031 لا يملك ذاكرة ROM على الشريحة فإن هذا المأخذ يفعّل بالوصل إلى الأرضي GND للإشارة بأن الشيفرة مخزنة خارجيا.

– مأخذ اتمكين ماسك العناوين 'ALE/Prog على الـ pin 30، بما أن البوابة صفر port 0 لدى المتحكم 8051 تعرض العناوين والمعطيات معا، فإن هذه الإشارة تستخدم لفكهما عن بعض de-multiplexing، عندما تكون ALE/Prog' = 1 .

– مأخذ اتمكين البرنامج المخزن'PSEN (Program Store Enable) على الـ pin 29 وهي إشارة خرج توصل بإشارة السماحية OE للذاكرة ROM. تستخدم الإشارة 'PSEN مع ALE مع ذاكرة ROM الخارجية.

أما بخصوص البوابات، فإن البوابات الأربعة تنقل البيانات بالاتجاهين

البوابة الأولى P0 : بمأخذها الثمانية P0.7~P0.0 على المآخذ pins 32–39 بالترتيب تحمل خطوط المعطيات $D_7 \sim D_0$ والثمانية خطوط الأولى من العناوين $A_7 \sim A_0$ وتسمى معا $AD_7 \sim AD_0$.

البوابة الثانية P1: بمأخذها الثمانية P1.7~P1.0 موزعة على المآخذ pins 1–8 غير محملة بأي نوع آخر من الإشارات

البوابة الثالثة P2: بمأخذها الثمانية P2.7~P2.0 على المآخذ pins 21–28 ومأخذ هذه البوابة تحمل بقية خطوط العنونة $A_8 \sim A_{15}$.

البوابة الرابعة P3: بمأخذها الثمانية P3.7~P3.0 موزعة على المآخذ pins 10–17 وهي تحمل مجموعة من إشارات التحكم.

يمكن للمتحكم 8051 برمجة ذاكرة بحجم 64 KB، وبالتالي إذا لم يتطلب التطبيق أي ذاكرة غير تلك المضمنة على الشريحة، فإن كافة الأجزاء الأربع الأساسية للمتحكم من

(Priority Interrupt Controllers, Programmable Timers,ROM&RAM) تكون متاحة للاستخدام كمنافذ خرج أو دخل.

أما إذا لزم التطبيق ذاكرة إضافية فيمكن برمجة المنفذ P0 والمنفذ P2 ليعملا كممر مضاعف للمعطيات والعناوين. عندما يستخدم هذا الممر مع ذاكرة خارجية، فإن خطين من المنفذ P3 سيستخدمان لتوليد الإشارتين 'RD و 'WR.

وبما أن المتحكم 8051 يحتوي على جملة من الدارات المخصصة لربط المعطيات التسلسلية ، فإنه عندما يستخدم للنقل التسلسلي للمعطيات، يقوم باستخدام مأخذين من البوابة P3 هما الخطان RxD, TxD لإرسال المعطيات التسلسلية واستقبالها. ويخصص من البوابة P3 خطان للمقاطعة الخارجية (p3.2, p3.3) وخطان للمؤقت.

١٩-١-١ معمارية الذاكرة في المتحكم 8051

تملك التركيبة MCS-51 أربعة أنواع مختلفة من الذاكرة هي:

Internal RAM, Special function registers, Program memory,
and External data memory.

بعض إصدارات المتحكم 8051 تستخدم ضمن الشريحة ذاكرة Flash. المفهوم العام لكل من ذواكر المتحكم الأربعة والمجال العنوني لكل منها يمكن اختصاره بالتالي:

- الذاكرة RAM الداخلية (Internal RAM اختصارا IRAM):

تتوضع هذه الذاكرة من العنوان 0 وحتى 0xFF، بحيث IRAM من العنوان 0x00 وحتى 0x7F يمكن الولوج لها بشكل مباشر. أما IRAM من العنوان 0x80 وحتى 0xFF) يجب الولوج لها بشكل غير مباشر، باستخدام المسجلات R0, R1 القوا عدية بعنوان للولوج معباً بالمسجلين (والـ 128 bits في IRAM بالمواضع 0x20 وحتى 0x2F تخصص لـ bit Addressable).

- المسجلات ذات الوظائف الخاصة (Special Function Registers):

تتوضع مسجلات SFR بنفس المجال العنوني مثل IRAM، من 0x80 وحتى 0xFF، ويتم الولوج لها مباشرة باستخدام نفس التعليمات لمثل النصف السفلي من IRAM، ولا يمكن الولوج إليها بشكل غير مباشر عبر R0, R1.

-- ذاكرة البرنامج (Program memory اختصارا PMEM):

تبدأ هذه الذاكرة التي هي من نوع ROM وبسعة 4 KB بالعنوان 0H بمجال عنوني منفصل. يمكن أن تكون ضمن الشريحة أو خارجها بالارتباط مع النموذج المحدد قيد الاستخدام.

- ذاكرة المعطيات الخارجية (External data memory اختصارا XRAM):

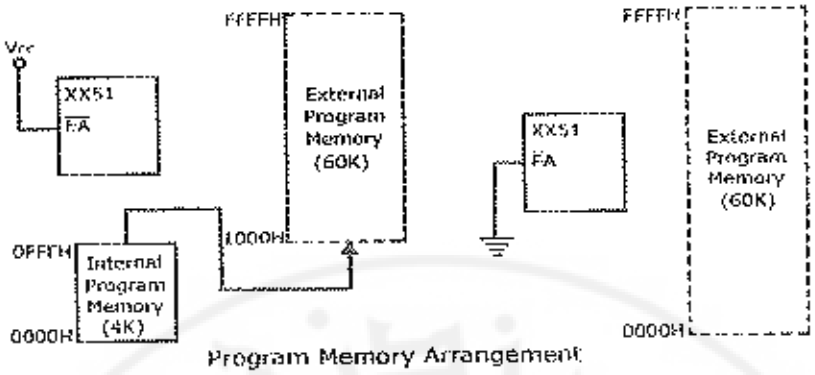
تمثل هذه الذاكرة الفراغ العنوني الثالث، وتبدأ أيضاً من العنوان 0H ويمكن أن تكون أيضاً ضمن الشريحة أو خارجها. بعض إصدارات المتحكم 8051 تتضمن 256 Byte من الذاكرة IRAM إضافة إلى بعض الكيلو بايت من XRAM على الشريحة نفسها.

١٩-١-٢ تنظيم الذاكرة في المتحكم 8051

يلاحظ أن للمتحكم 8051 نوعين منفصلين من الذاكرة هما: ذاكرة للبرنامج وذاكرة للمعطيات. يتعامل المتحكم مع كل منهما على انفراد وكأنها كينونة منفصلة.

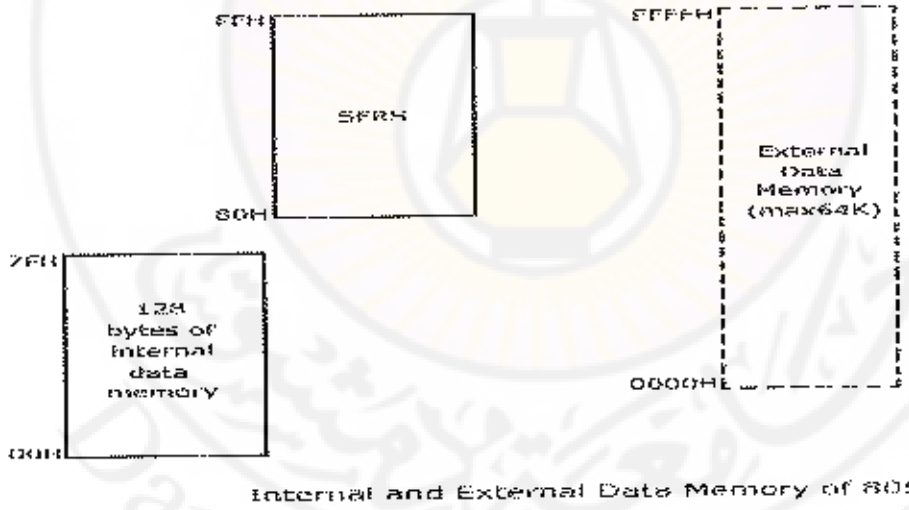
تنظيم ذاكرة البرنامج Program Memory Organization

ترتيب ذاكرة البرنامج لدى المتحكم 8051 يسمح له من أن يلج إلى برنامج داخلي بحجم 4 KB بشكل مباشر. وإذا احتاج المتحكم لذاكرة خارجية، يمكن إضافة ذاكرة بحجم60 KB على الأكثر وهذا يتحدد بقيمة الإشارة 'EA، بحيث: إذا كان EA' = Ucc يتم جلب التعليمات من الذاكرة الداخلية بحجم 4 KB. وإذا تم تجاوز أول 4 KB فان التحكم ينتقل تلقائيا إلى الذاكرة الخارجية لجلب بقية التعليمات. إذا أراد المبرمج أن يجلب التعليمات من الذاكرة الخارجية فقط (أي بتجاوز الذاكرة الداخلية)، يجب وصل EA' = GND كما هو مبين بالشكل 6-19.



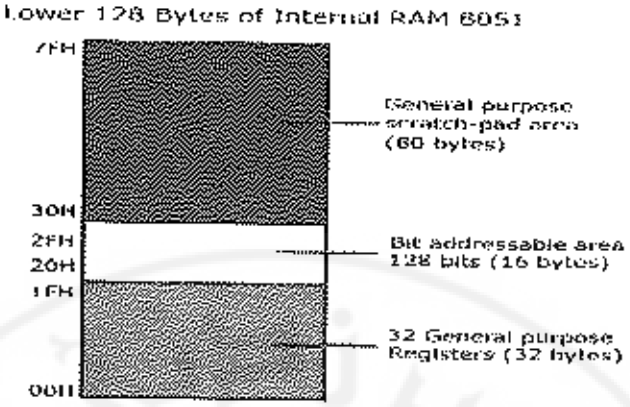
الشكل6–19: تنظيم ذاكرة البرنامج لدى المتحكم 8051

يملك المتحكم 8051 ذاكرة داخلية للمعطيات بسعة 128 Bytes ويسمح بوصل ذاكرة خارجية بسعة عظمى تصل إلى 64 KB. وبالتالي الحجم الكلي لذاكرة المعطيات لدى المتحكم 8051 يمكن أن يكون حتى 64 KB (خارجي) + 128 Bytes (داخلي). يظهر الشكل 7–19 ذاكرة المعطيات الداخلية والخارجية لدى المتحكم الصغري 8051.



الشكل7–19: تنظيم ذاكرة المعطيات Data Memory Organization

كما ويبين الشكل 8–19 توزع أدنى 128 Bytes من الذاكرة الداخلية للمتحكم 8051



الشكل8–19 : الأجزاء الأساسية لذاكرة المعطيات الداخلية (128 Bytes) للمتحكم الصغري 8051

وكما يبدو من الشكل أعلاه يوجد ثلاثة أجزاء لذاكرة المعطيات هي:

- بنوك المسجلات Registers Banks
- منطقة قابلة للعنونة Bit Addressable Area
- منطقة دفتر مسودة Scratch pad Area

بنوك المسجل من الـ 32 Bytes السفلى من الذاكرة الداخلية عددها أربعة هي: Bank0 – Bank3 كما هو مبين أدناه.

RS1	RS0	Reg. bank	Address
0	0	0	00H – 07H
0	0	0	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

كل بنك يملك 8 مسجلات يرمز لها من R0 – R7. بلحظة ما يكون منتقى بنك واحد للعمليات، والمسجلات ضمن البنك المنتقى يتم الولوج إليها باستخدام الاختصارات R0, R1, ...

تستخدم المسجلات لتخزين البيانات والمعاملات خلال التنفيذ. وبشكل تلقائي بنك المسجل Bank0 هو الذي يتم اختياره بعد عملية التصفير.

مناطق العنونة البتية the bit addressable areas تستخدم عادة لتخزين بيتات المتحولات ويمكن إيجاد هذه المنطقة في الـ 16 Bytes التالية لبنوك المسجلات. وهي مصممة من العنوان 20H إلى 2FH (المجموع 128 Bits). يمكن الولوج إلى كل بيت من العنوان 00H إلى 7FH ضمن الـ 128 Bits.

تستخدم هذه المنطقة لتخزين متغيرات بطول 1Bit من البرنامج (معلومة بطول 1 bit). المنطقة الثالثة The Scratch pad area تمثل الـ 80 Bytes العليا التي تستخدم لتخزين الغرض العام. وهي من 30H إلى 7FH وتتضمن المكدس Stack أيضاً.

مسجلات المتحكم 8051 CPU Registers

يوجد لدى المتحكم الصغري 8051 ثماني مسجلات للاستخدامات العامة، هي المسجلات R0 – R7 مخططة ضمن IRAM بين العناوين 0x1F and 0x00.

يستخدم فقط 8 Bytes من هذا المجال بأي وقت، وتحدد ببيت انتقاء البنك في PSW.

يوجد 256 bits معنونة بشكل مباشر. وهي الـ 16 موقع من IRAM بالمواضع من 0x20 – 0x2F و16 مسجل للوظائف الخاصة 0x80, 0x88, 0x90,...0xF8 يستخدم المتحكم 8051 مع تعليمات المجمع Assembler مجموعة من المسجلات هي:

A (Accumulator), PSW (Program Status Word), SP (Stack Pointer), PC (Program Counter), DPTR (Data Pointer)

المسجل الوحيد غير الموجود بمخطط الذاكرة لدى المتحكم 8051 هو المسجل PC بطول 16 bit الذي يحدد عنوان التعليمة التالية للتنفيذ.

تعليمات التفريع تجهز الإزاحة المؤشرة بطول 8 bit لتضاف إلى الـ PC.

المسجلات التالية بمخطط الذاكرة تخصص لوظائف محددة ضمن المجال وهي:

- (0x81) Stack pointer SP. This is an 8-bit register used by subroutine call and return instructions.
- (0x82–83) Data pointer DP. This is a 16-bit register that is used for accessing PMEM and XRAM.
- (0xE0) Accumulator A. it uses by most instructions.
- (0xF0) B register. This is used as an extension to the accumulator for multiply and divide instructions.
- (0xD0) Program status word PSW. This contains important status flags

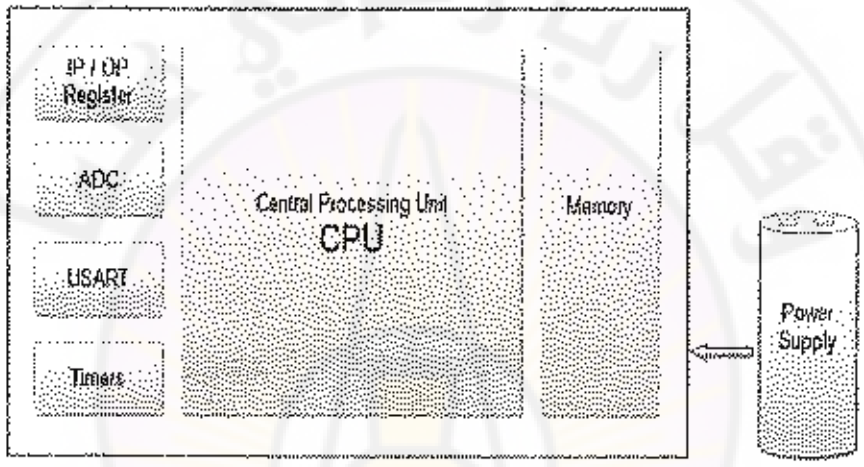
وبخصوص مسجل المؤشرات لدى المتحكم 8051 فله التوزيع التالي:

CY	AC	F0	RS1	RS0	OV	UD	P

- PSW.0: P Parity. Gives the parity.
  Parity flag Set/Reset odd/even parity
- PSW.1: UD User Defined.
- PSW.2: OV Overflow flag.
- PSW.3: RS0 Register select 0. The low-order bit of the register bank. Set when banks at 0x08 or 0x18 are in use.
- PSW.4: RS1 Register select 1. The high-order bit of the register bank. Set when banks at 0x10 or 0x18 are in use.
- PSW.5: F0 Flag 0. For general software use
- PSW.6: AC auxiliary carry.
- PSW.7: C Carry bit.

٢-١٩ متحكمات AVR

تم تطوير العائلة AVR من المتحكمات في العام 1996 بوساطة شركة Atmel. وتعرف أيضاً بـ Advanced Virtual RISC وقد اشتق المصطلح AVR من أسماء مطوريه أي من Alf-Egil Bogen Vegard Wollan RISC. يعدّ المتحكم AT90S8515 المتحكم الأول الذي بني على المعمارية AVR بالعام 1997. يظهر الشكل 9-19 معمارية المتحكم AVR.



الشكل9-19: الشكل العام لمعمارية متحكمات AVR

تصنف متحكمات AVR المتاحة بثلاثة أصناف هي:

- TinyAVR: وتمتاز بحجم صغير وحجم ذاكرة قليل وهي تناسب التطبيقات البسيطة.
- MegaAVR: وهي الأكثر شعبية أو انتشار من بين متحكمات AVR، تحتوي على كمية جيدة من الذواكر (حتى 256 KB) وعلى عدد عالٍ من المحيطيات المكاملة داخليا وهي تناسب التطبيقات المتوسطة وحتى المعقدة.
- XmegaAVR: وهي تستخدم تجاريا للتطبيقات المعقدة التي تتطلب ذاكرة برنامج كبيرة، وسرعة تنفيذ عالية. يظهر الجدول 3-19 مقارنة بين المواصفات التقنية لهذه السلاسل الثلاثة المشار إليها من متحكمات AVR.

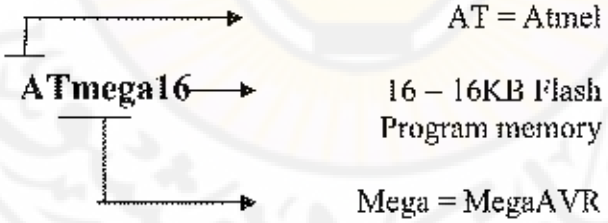
الجدول3-19: المواصفات الفنية الأساسية لمجمل سلاسل متحكمات AVR

Series Name	Pins	FlashMemory	Special Feature
TinyAVR	6 -32	0.5 – 8 KB	Small in size
MegaAVR	28 – 100	4 – 256 KB	Extended Peripherals
XmegaAVR	44 – 100	16 – 384 KB	DMA, Event System included

متحكم AVR بـ 8 bit مرتبط بالعائلة RISC، حيث بالمعمارية RISC جملة تعليمات الحاسوب ليست فقط أقل بالعدد وإنما أيضاً أبسط وأسرع في العمليات. وبشكل عام تتميز مجمل متحكمات AVR بأنها متحكمات سريعة، تنفذ معظم التعليمات بحلقة تنفيذ واحدة، وهي تقريبا بأربع مرات أسرع من متحكمات PIC ، وتستهلك طاقة أقل، ويمكن أن تعمل بأنماط مختلفة لحفظ الطاقة.

١٩-٢-١ المتحكم ATmega 16

سنتعرف فيما يلي إحدى متحكمات السلسلة الأوسع انتشاراً من متحكمات AVR وهي السلسلة MegaAVR وتحديدا المتحكم ATmega 16 الذي يعني التالي:



يمثل المتحكم الصغري At mega 16 من إنتاج شركة Atmel دارة متكاملة من رتبة VLSI (Very Large Scale Integration) مصنع بالتقنية CMOS وله القدرة على معالجة بيانات بعرض 8 bit. يؤدي هذا المتحكم وظائف تحكمية بحسب الضبط والتهيئة التي تجرى عليه، وبحسب البرنامج المخزن فيه.

بني المتحكم Atmega16 على أساس بنية AVR المتطورة التي تقدم قاعدة غنية بالتعليمات إضافة إلى 32 مسجلاً للاستخدامات العامة مرتبطة بوحدة الـ ALU. يحتوي المتحكم ATmega16 ثلاثة أنواع مختلفة من الذواكر هي:

- Flash EEPROM وتستخدم لتخزين البرنامج بوساطة المستخدم ضمن المتحكم. المتحكم ATmega16 مزود بـ 16 KB من ذاكرة فلاش Flash EEPROM
- ذاكرة EPROM قابلة للعنونة على مستوى البايت بسعة 512 Bytes.
- ذاكرة SRAM استاتيكية وهي على عكس الذواكر السابقة لأنها من النوع المتطاير. المتحكم ATmega16 مزود بـ 1KB من ذاكرة SRAM الداخلية. قسم صغير من الذاكرة SRAM يوضع جانبا للمسجلات عامية الهدف (مسجلات الأغراض العامة) المستخدمة من قبل الوحدة CPU وبعض الأنظمة الجزئية المحيطية بالمتحكم.

يوجد ضمن السلسلة MegaAVR متحكمات أخرى مثل المتحكمان ATmega8 و ATmega32 ، وهما مشابهان من حيث المعمارية مع المتحكم ATmega16. يظهر الجدول 4-19 مقارنة بين عناصر مختلفة من العائلة MegaAVR Family

الجدول4-19: لمواصفات الأساسية لمتحكمات العائلة MegaAVR Family

Part name	ROM	RAM	EPROM	I/O pins	Operating Frequency	Package
Atmega8	8 KB	1 KB	512 B	23	0-16 MHz	28
Atmega8L	8 KB	1 KB	512 B	23	0 – 8 MHz	28
Atmega16	16 KB	1 KB	512 B	32	0-16 MHz	40
Atmega16L	16 KB	1 KB	512 B	32	0 – 8 MHz	40
Atmega32	32 KB	2 KB	1 KB	32	0-16 MHz	40
Atmega32L	32 KB	2 KB	1 KB	32	0 – 8 MHz	40

علما أن عدد المؤقتات بكل من هذه المتحكمات المبينة بالجدول أعلاه ثابت وهو 3 وعدد المقاطعات لكل منها هو 21 باستثناء المتحكمين Atmega8 و Atmega8L ولكل منهما 19 مقاطعة، كما أن لكل متحكم من المتحكمات بدون الحرف L تغذية بحدود 4.5 – 5.5V . أما المتحكم الذي يرد الحرف L باسمه، مثل Atmega8L, Atmega16L, Atmega32L تكون تغذيته 2.7 – 5.5V. يعمل Atmega 16 من AVR بحلقة ساعية 1 MHz وتصل سرعته إلى 12 مليون تعليمة بالثانية. يصنع على شكل دارة متكاملة بـ 40 Pins وهو ينتمي إلى صنف megaAVR من العائلة AVRfamily، ويمتلك بحدود 140 تعليمة جميعها مبنية على حلقة تعليمة واحدة.

١٩-٢-١-١ ميزات المتحكم Atmega16

أهم ميزات المتحكم Atmega16 يمكن إجمالها بالنقاط التالية:

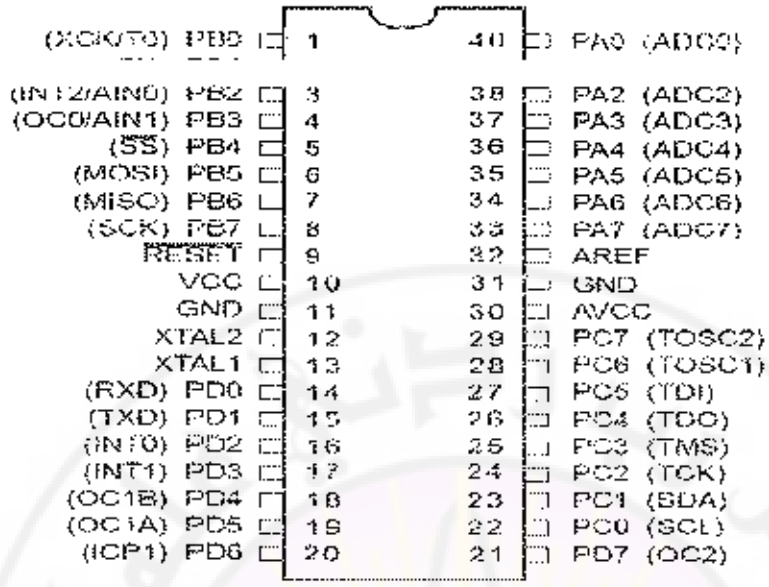
- أنه متحكم عالي الأداء، يمتلك خطوط نقل داخليه ومداخل ومخارج ومسجلات بعرض 8 bit ويعمل بجهد تغذية 2.7 – 5.5V للمتحكم ATmega16A
- Advanced RISC Architecture
  - بـ 131 تعليمة فعّالة (قوية) أغلبها ينفذ بحلقة ساعية واحدة.
  - يحوي 8 x 32 مسجلاً عمل للأغراض العامة
  - Up to 16 MIPS Throuputput at 16 MHz
- مقاطع ذاكريه عالية التحمل High Endurance memory segments
  - 16K ذاكرة فلاش Flash Memory
  - 1 KB ذاكرة RAM من النوع الاستاتيكي SRAM
  - 512 Bytes من ذاكرة EEPROM
- ميزات المحيطيات Peripherals Features
  - مؤقت/عداد 8 bit Timer/Counter with Separate Prescalers عدد /٢/
  - مؤقت/عداد 16bit Timer/Counter with Separate Prescalers عدد/١/
  - عداد بالزمن الحقيقي Real Time Counter with Separate Oscillator

- قنوات تعديل PWM (Pulse Width Modulation) Channels عدد /4/
- محول إشارة 8-channel, 10-bit ADC
- مقارن تمثيلي على الشريحة on-chip Analog Comparator
- دارة للنقل التسلسلي USART مبرمجة
- Master/Slave SPI Interface
- Programmable Watchdog timer with separate On-chip Oscillator
- 32 خط I/O مبرمجة
- واجهة ربط من نوع JTAG Interface
- ميزات خاصة للمتحكم Special Microcontroller Features
  - مصادر داخلية وخارجية للمقاطعة
  - ستة أنماط للنوم Six Sleep Modes
- سرعة المعالجة Speed Grades من0 وحتى 16 MHz
- تغليف PDIP 40, 44-lead TQFP

١٩-٢-١-٢ تركيبة المتحكم AT mega16

يبين الشكل 10-19 توزع إشارات المتحكم الصغري AT mega16 على مآخذه الأربعين، حيث يملك بالإضافة للبوابات الأربعة بـ 32 مأخذ على ثمانية إشارات تحكمية أخرى هي:

- مأخذ للتغذية Vcc = +5V (Pin 10)
- ومأخذان للأرضي GND (Pin 11, 31)
- مأخذان لوصل الكريستال XTAL1, XTAL2 (Pin 12, 13)
- مأخذ واحد للمدخل RESET' (Pin 9)
- مأخذ واحد لـ AVCC (Pin 30) الذي يمثل مأخذ جهد تغذية للبوابة PA وللمحول A/D Converter، بحيث إنه يجب وصله خارجيا للجهد VCC ، حتى ولو لم يكن المحول ADC مستخدما. أما إذا كان ADC مستخدما فيجب وصله إلى VCC عبر فلتر تمرير منخفض.



شكل10-19: توزع الإشارات للمتحكم الصغري Al mega 16

– مأخذ واحد لـ AREF (Pin 32) وهو يمثل مرجع تمثيلي Analog Reference للمحول ADC.

أما بخصوص الـ 32 I/O Pins للنقل بالاتجاهين فهي مجمعة بأربع بوابات:

PortA , PortB , PortC , PortD للدخل والخرج، وهذه المآخذ محملة بوظائف إضافية منها:

– ADC (AD0 – AD7 on port A)

– SPI (MISO, MSOI, SCK on port B)

– UART (Rx, Tx, on port D)

– External Int. (INT0 – INT2) و Timers (OC0 – OC2)

1 – البوابة PA (PA0 ÷ PA7) وتشغل المآخذ (Pin40 – Pin33) على الترتيب، بمعنى أن PA0 توجد على الـ Pin 40 و PA7 تتواجد على الـ Pin33. تخدم هذه البوابة كمداخل تمثيلية للمحول A/D بحيث تركب إشارات المحول ADC0 ÷ ADC7 بالترتيب، بمعنى: أن الـ Pin40 يقابل PA0/ADC0 وهكذا حتى آخر مأخذ من مآخذ البوابة.

وتعمل كبوابة نقل I/O بالاتجاهين إذا لم يكن المحول A/D مستخدما. مآخذ البوابة PA تكون بالحالة الثالثة tri-states عندما يكون التصفير فعالا.

2- البوابة PB (PB0 ÷ PB7) وتمثل 8 bit I/O Port للنقل بالاتجاهين. وتخدم هذه البوابة وظائف الميزات الخاصة المختلفة للمتحكم ATmega16 الموزعة على مآخذ البوابة وفق ما هو مبين بالجدول 5-19.

الجدول 5-19: الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة B

Pin	Signal	Pin	Signal
1	PB0/ (XCK/T0)	5	PB4/ SS'
2	PB1/ T1	6	PB5/MOSI
3	PB2/ (INT2/AIN0)	7	PB6/MISO
4	PB3/ (OC0/AIN2)	8	PB7/SCK

3- البوابة PC (PC0 ÷ PC7) وتمثل 8 bit I/O Port للنقل بالاتجاهين. وتخدم هذه البوابة وظائف حد أو واجهة الربط JTAG interface وبعض الميزات الخاصة للمتحكم ATmega16 الموزعة على مآخذ البوابة وفق ما هو مبين بالجدول 6-19.

الجدول 6-19: الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة C

Pin	Signal	Pin	Signal
22	PC0/SCL	26	PC4/TDO
23	PC1/SDA	27	PC5/TDI
24	PC2/TCK	28	PC6/TOSCI
25	PC3/TMS	29	PC7/TOSC2

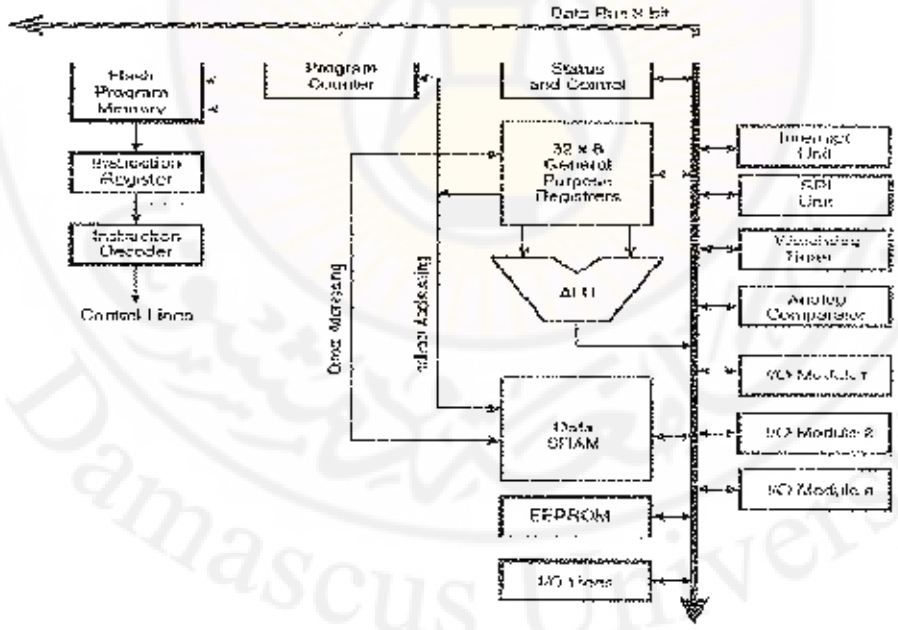
4- البوابة (PD0 ÷ PD7) PD وتمثل 8 bit I/O Port للنقل بالاتجاهين. وتخدم هذه البوابة وظائف الميزات الخاصة المختلفة للمتحكم ATmega16 الموزعة على مآخذ البوابة وفق ما هو مبين بالجدول 7-19.

الجدول 7-19: الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة D

Pin	Signal	Pin	Signal
14	PD0/RxD	18	PD4/OC1B
15	PD1/TxD	19	PD5/OC1A
16	PD2/INT0	20	PD6/ICP
17	PD3/INT1	21	PD7/OS2

AVR CPU

يبين الشكل 11-19 صميم المعمارية (AVR Core Architecture).



الشكل 11-19: صميم معمارية AVR

الوظيفة الرئيسية لصميم CPU هي تأمين التنفيذ الصحيح للبرنامج. ولذلك يجب أن تكون CPU قادرة على الولوج إلى الذواكر، إجراء الحسابات، والتعامل مع المقاطعات، مراقبة المحيطيات. ولتحقيق أقصى قدر من الأداء والتوازي استخدمت AVR معمارية هارفارد التي تستخدم ممرات وذواكر منفصلة لكل من البيانات والبرامج معتمدة مستوى منفرداً من الـ Pipelining. ستة من المسجلات الـ 32 يمكن استخدامها كثلاثة مسجلات كل منها بطول 16 bit كمسجلات مؤشرة لعنونة المعطيات بشكل غير مباشر.

أحد هذه المؤشرات يمكن استخدامه كمؤشر عنوان للنظر بالجداول في ذاكرة الفلاش للبرنامج. فراغ ذاكرة فلاش للبرنامج يقسم إلى جزأين هما:

الجزء الخاص ببرنامج التمهيد Boot program

وجزء للبرنامج التطبيقي Application program section.

كلا الجزأين مخصص ببيت قفل لحماية الكتابة و القراءة/ الكتابة.

يخزن خلال المقاطعات عنوان العودة لعداد التعليمات PC بالمكدس. ويتوضع المكدس بالذاكرة SRAM وحجمه يتحدد بحجمها. مؤشر المكدس SP متاح للقراءة/الكتابة في فضاء الـ I/O. ويمكن الولوج لذاكرة المعطيات SRAM عبر خمس أنماط عنونة مختلفة لدى معمارية AVR.

فراغ الذواكر لـ I/O يحتوي على 64 عنوان لوظائف محيطيات الـ CPU كمسجلات تحكم، SPI، ووظائف أخرى لـ I/O.

تعمل وحدة ALU بأداء عالي بالاتصال المباشر مع كل الـ 32 مسجلاً للأغراض العامة. وتنفذ كل العمليات الحسابية بين هذه المسجلات داخليا بحلقة ساعية واحدة. تقسم كل عمليات ALU إلى ثلاثة أصناف رئيسية هي: الحسابية، المنطقية، والوظائف البتية bit-functions.

أما بخصوص مسجلات المتحكم فإن كل تراكيب المتحكمات مبنية عبر مواضع 8 bit (1 Byte) بالذاكرة RAM (حيث RAM هي بنك من بايتات الذاكرة) للمتحكم تدعى بالمسجلات. كل الوظائف مخططة بمواقعها في الذاكرة.

يوجد بالمتحكم ATmega16 إجماليا 32 x 8 bit مسجل طول كل منها 8 bit وبخصوص المكونات الداخلية الأخرى للمتحكم نتعرف عليها وفق التالي

أولا : ذواكر المتحكم Memories

- ذاكرة الـ Flash memory ويحوي المتحكم على ذاكرة بسعة 16 KB من هذا النوع متضمنة داخل شريحة المتحكم . تستخدم هذه الذاكرة عادة لتخزين البرنامج عليها وتنفيذه منها، وبما أن كافة تعليمات هذا المتحكم ذات عرض كلمة 16-bit أو 32-bit فقد تم تنظيم هذه الذاكرة على أنها 8K x 16 ، وقد تم تحديد هذا الطول من خلال عداد خاص في هذا المتحكم هو ذو عرض كلمة 13-bit ، قادر على عنونة الـ 8K موقع لهذه الذاكرة والتي تمتد من $0000 وحتى $1FFF .

- ذاكرة الـ SRAM: وهي الذاكرة التي يتم فيها تخزين المعلومات قبل معالجتها وبعدها تقسم هذه الذاكرة إلى ثلاثة أقسام هي: قسم سجلات الاستخدامات العامة وقسم سجلات المدخل والخرج وقسم الـ SRAM الداخلية. يمتد عنوان كل قسم من هذه الأقسام على الترتيب وفق الشكل التالي: $0 - $1F ، $20 - $5F ، $60 – $45F. يتوضع في هذه الذاكرة جميع السجلات اللازمة للتعامل مع مداخل ومخارج هذا المتحكم إضافة إلى طرفياته ، حيث إن كل طرفية من هذه الطرفيات تمتلك سجل أو أكثر يتم من خلاله ضبط هذه الطرفية و تحديد نمطها وآلية عملها ، والحصول على المعلومات اللازمة عنها أثناء عمل المتحكم الصغري .

- ذاكرة EEPROM: يملك هذا المتحكم ذاكرة EEPROM بحجم 512 bytes ، وقد نظمت هذه الذاكرة على أنها منفصلة عن بقية الذواكر في المتحكم .

ثانياً : العدادات و المؤقتات Timers & Counters

يملك المتحكم مجموعة من العدادات التي يمكن أن يشتق منها المؤقت ، وهي توجد على نمطين: الأول 8 bit يعّد من 0 إلى 255 ،

والثاني 16 bit يعّد من 0 إلى 65535 .

ويمكن تزويد هذه العدادات بنبضات ساعة داخلية مشتقة من نبضات الساعة للمتحكم أو من نبضات ساعة خارجية . يمكن أن تفعل هذه العدادات عدة مقاطعات أثناء عملية العد مثل: مقاطعة وصول العداد إلى القيمة العظمى وانتقاله إلى الصفر .

مقاطعة حدوث المساواة بين قيمة العداد وقيمة أخرى محددة من قبل المستثمر. تعمل هذه العدادات وفق عدة أنماط هي:

1- النمط التقليدي Normal mode

يتم في هذا النمط العَد من الصفر إلى القيمة العظمى أي 255 في النمط 8 bit أو إلى 65535 في النمط 16 bit، ويتم عند ذلك تصفير العداد.

2- نمط المقارنة Clear Time on Compare match

يبدأ العد في هذا النمط من الصفر إلى قيمة ما يحددها المستثمر حيث يتم عندها تصفير العداد والبدء بالعد من جديد، كما يمكن تعريف القطب OC0 المرتبط بوحدة العداد/المؤقت ليقوم بتغيير حالته المنطقية عند وصول العداد/المؤقت لقيمته العليا وبالتالي إمكانية توليد إشارة مربعة على هذا القطب بتردد يعطى بالعلاقة:

$f_{OC0} = f_{clk} / 2xNx(1 + TOP)$ حيث: fclk : هو تردد العمل للمتحكم و

N : عامل قسمة تردد عمل المتحكم.

TOP : القيمة العليا للعداد.

Foc0 : تردد القطب OC0

3- النمط PWM Pulse Width Modulation

إن اسم هذا النمط مشتق من إمكانية استعماله لتوليد قطار من النبضات المربعة التي يمكن تعديل عرض نبضاته عند حدوث حالة المساواة مع قيم معينة يدخلها المستثمر إلى العداد ، يتم العَد في هذا النمط من الصفر إلى القيمة العظمى(255 ، 65535) ثم يعود إلى الصفر ويسمى في هذه الحالة بـ Fast PWM، أو يمكن أن يعَد من الصفر إلى القيمة العظمى بشكل تصاعدي ثم يعود إلى الصفر بشكل تنازلي. يختلف هذا النمط عن الأنماط الأخرى بالخيارات الوظائف التي يمكن أن ينفذها المتحكم.

ثالثاً : أنظمة المقاطعات Interrupts

المقاطعة هي عبارة عن أوامر داخلية أو خارجية ذات أفضلية أعلى من أفضلية التعليمات التي ينفذها المتحكم الصغري ، تقوم هذه الأوامر بمقاطعة عمل المتحكم لتنفيذ مجموعة من التعليمات الفرعية التي يحددها المستثمر، ثم تعود لمتابعة تنفيذ البرنامج الرئيسي .

يحوي المتحكم أربع مقاطعات خارجية هي :

Reset , Int0 , Int1 , Int2 وسبع عشرة مقاطعة داخلية من أهمها :

ADC : مقاطعة إنهاء عملية تحويل الإشارة من التمثيلي إلى الرقمي .

T.comp : مقاطعة حدوث المساواة بين قيمة العداد وقيمة معينة يحددها المستثمر .

T.over : مقاطعة وصول العداد إلى القيمة العظمى .

يرتبط تفعيل المقاطعات الداخلية بشروط عدة منها :

- تأهيل سجلات التحكم الخاصة بالمقاطعة المطلوبة .
- حدوث الحالة التي تتطلب حدوث المقاطعة مثل انتهاء العداد من العَد.

تعدّ مقاطعة Reset مقاطعة داخلية وخارجية في آن واحد ، حيث يمكن تفعيلها من البرنامج داخلياً أو من خلال Reset pin الخارجي .

رابعاً : وحدات الدخل والخرج Input / Output ports

إن الخاصية الأهم في هذه الوحدات هي إمكانية تغيير اتجاه سير المعطيات عبرها ما بين الدخل والخرج، أي أنها تملك قابلية التبديل ما بين الكتابة فيها (حالة الدخل)، أو القراءة منها (حالة الخرج) وذلك بوساطة ضبطها برمجياً .

يمكن عند ضبط هذه الوحدات في حالة الدخل إجراء ضبط آخر هو وصل مقاومة داخلية بين المدخل والتغذية Vcc التي تسمى بمقاومة الشد Pull up resistor كما هي الحالة في دارات المجمع المفتوح Open collector .

الهدف من هذه المقاومة هو التحكم في قيمة ومسار التيار المستجر من المتحكم الصغري أو إليه وذلك لحمايته من التلف نتيجة إستجرار تيار زائد عن القيمة الصحيحة.

خامساً: وحدة المعالجة المركزية CPU Core

ومن أهم وظائفها : – إدخال البيانات وإخراجها من الذواكر وإليها .

– التحكم بالعناصر المحيطية الموصولة مع المتحكم الصغري .

– تفعيل المقاطعات وتأهيلها وإلغاؤها .

ومن أهم أقسام هذه الوحدة :

– وحدة الحساب والمنطق ALU – Arithmetic logic unit

وهي المسؤولة عن تنفيذ التعليمات الحسابية الرياضية الأساسية والتعليمات المنطقية

– سجل الحالة Status register :

يعدّ أهم سجلات المتحكم الصغري على الإطلاق ، وهو يحوي معلومات عن معظم العمليات الحسابية حيث يمكن استخدام هذه المعلومات في التحكم بمسار تنفيذ التعليمات وتوجيهها. يتكون هذا السجل من 8 bits وفق ما هو مبين بالشكل أدناه والمعلومات التي يمكن معرفتها عن طريق هذا السجل هي :

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

حصول حالة الحمل Carry في عمليتي الجمع و الطرح ، . . . . . الخانة / C /.

الحصول على نتيجة الصفر بعد أي عملية حسابية ، . . . . . الخانة / Z /.

الحصول على عدد سالب بعد عملية الطرح . . . . . الخانة / N /.

حصول حالة حمل عند التحويل إلى المتمم الثنائي أثناء عملية الطرح، . الخانة / V /

إجراء عملية Exclusive or المنطقية بين الخانتين / N / و / V /، . الخانة / S /

حصول حالة الحمل Carry بين المرتبتين الرابعة والخامسة في العدد، الخانة / H /.

يمكن تخزين أو قراءة القيمة بالخانة / T / واستخدامها للتحكم حسب حاجة المستثمر

تأهيل أو إلغاء بت المقاطعات العام I Bit . . . . . الخانة / I /.

سادساً : وحدة التبديل من تمثيلي إلى رقمي ADC:

يقدم المتحكم الصغري ATmega16 مبدل من تشابهي إلى رقمي يعمل على مبدأ التقريب المتتالي. يرتبط هذا المحول مع ناخب بثمان قنوات مربوط على المنفذ A. تحوي وحدة ADC على دارة أخذ عينات وماسك التي تضمن مسك الجهد الداخل على قيمة ثابتة طوال فترة التحويل. تقدم هذه الوحدة الميزات التالية: – دقة تحويل تصل إلى 10-bit

– قيمة الخطأ المطلق ± 2LSB

– زمن التحويل 65 – 260 µ sec

– ثماني قنوات لاختيار الإشارة

–إمكانية اختيار جهد مرجعي داخلي2.56V

– نمط التحويل المتتالي أو التحويل الوحيد

– نمط نوم من أجل التخفيض من الضجيج.

تملك هذه الوحدة أقطاب تغذية تمثيلية منفصلة AVCC. ويمكن تزويدها بالجهد المرجعي من مصدر خارجي أو جهد مرجعي داخلي قيمته الاسمية 2.56V. ويمكن التقليل من مستوى الضجيج بتقصير خطوط الإشارة التمثيلية قدر الإمكان. يتم ربط القطب AVCC مع قطب التغذية الرقمي VCC بوساطة مرشح LC . يجب التأكد من عدم تغيير حالة أحد أقطاب المنفذ A عند استخدامه كقطب رقمي أثناء عملية التحويل.

سابعاً : وحدة الإرسال والاستقبال العامة بالنمطين التزامني واللا تزامني USART :

هي وحدة تخاطب تسلسلية تعمل بالنمطين التزامني واللاتزامني للإرسال والاستقبال، وهي عبارة عن وسيلة مرنة للتخاطب التسلسلي مع الحاسب عبر منفذ الـ COM. تقدم هذه الوحدة الميزات التالية: – نمط عمل دوبلكس (إرسال واستقبال بآن واحد)

– العمل بأحد النمطين التزامني أو اللاتزامني

– دعم إرسال حزمة تتضمن 5 ، 6 ، 7 ، 8 ، 9 من البيانات مع نبضة إنهاء إرسال واحدة أو اثنتين

– إمكانية توليد Parity زوجي أو فردي

٣-١٩ متحكمات PIC

تم تطوير متحكمات PIC من قبل شركة General Instruments وفي العام 1985 حولت شركة General Instruments قسم الـ Microelectronics إلى تكنولوجيا الشريحة المايكروية Microchip Technology وخصص PIC كمتحكم لربط المحيطيات. استخدمت الشركة مختصر PICs لمتحكماتها انطلاقا من المصطلحات: Programmable Interface Controller مثل (PIC1640 and PIC1650). انطلقت Microchip Technology بالعام 1993 من المتحكم PIC16C84 بطول 8 bit مع ذاكرة EPROM التي يمكن برمجتها باستخدام طرق البرمجة التسلسلية. واستمرت منذ عام 1998 بتطوير متحكمات جديدة بأداء عالٍ وبمعمارية معقدة وحسنت صنع الوحدات الملحقة، وأصبحت بالوقت الحاضر متحكمات PIC كثيرة الاستعمال وبخاصة للأغراض الصناعية، لأدائها العالي والاستهلاك المنخفض للطاقة، وهي معروفة بكثرة بين الهواة لانخفاض التكلفة، وإتاحتها بالسوق، وسهولة استخدامها بالأدوات التقنية والبرمجية مثل: Compilers, Simulators, Debuggers etc تعرض Microchip أربع عائلات للمتحكمات PIC كل منها مصمم للإشارة إلى الحاجات من مصممين مختلفين (على أساس الهندسة المعمارية الداخلية) وهي:

1. Base-Line: 12-bit Instruction Word length
2. Mid-Range: 14-bit Instruction Word length
3. High-End: 16-bit Instruction Word length
4. Enhanced: 16-bit Instruction Word length

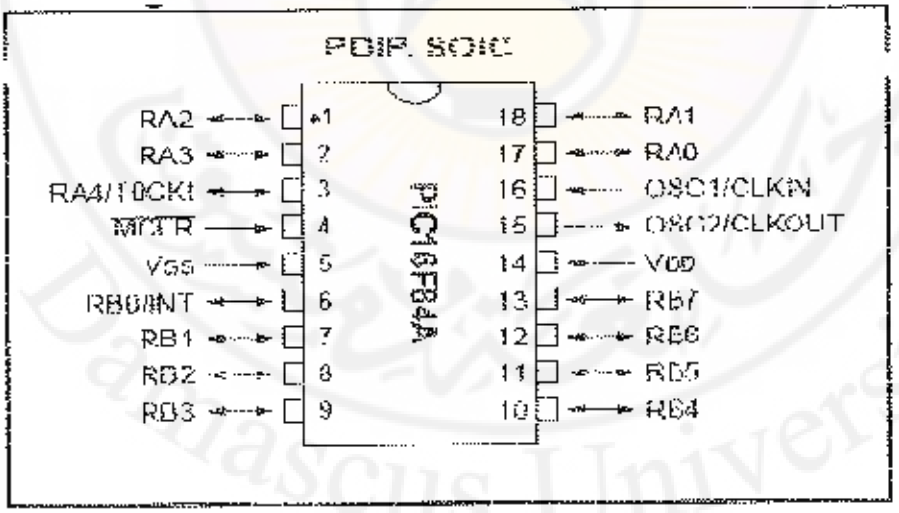
وهنا يمكن التساؤل كيف المتحكمات 8 bit تملك تعليمات بـ 12, 14, 16 bit؟ الجواب يكمن في حقيقة أن متحكمات PIC مبنية على معمارية Harvard. وبالتالي لكل من ذاكرة البرنامج وذاكرة المعطيات ممرات منفصلة، لذلك فإن متحكمات PIC تملك ممرات لذاكرة المعطيات بطول 8 bit ، وممرات لمعطيات ذاكرة البرنامج بطول 12, or 14, or 16 bit بحسب نوع العائلة.

كل متحكمات PIC تملك مزيجاً من المحيطيات المختلفة على الشريحة الواحدة مثل: محولات A/D، المقارنات، نماذج PWM، UARTs، مؤقتات، SPI و USB وغيرها. وهي تأتي بحجوم مختلفة، ابتداء من متحكم بـ 6 pins (وهو الأصغر في العالم) وحتى متحكمات بعدد عالٍ من المآخذ وتجهيزات عالية من الذواكر.

تعدّ السلسلة 16F من PIC هي الأنسب للانطلاق بتعرّف متحكمات Microchip (وتعدّ مايكرو شيب المزود الثاني الأكثر انتشارا في العالم للمتحكمات 8 bit بعد شركة Motorola)، وبالتالي سنبدأ بتعرّف السلسلة PIC 16F84 التي تبدو الأكثر شعبية (انتشارا) من بين المتحكمات كخطوة أولى وبعد ذلك نتعرف على المتحكم PIC16F877 الذي يعادل بإمكانياته المتحكم 8051 من إنتل أو Atmega16 من AVR والذي يصلح لمعظم التطبيقات التي نصادفها كخطوة ثانية. وعلى الصعيد البرمجي، يفضل المباشرة تعرّف السلسلة 18F لأنها تملك معمارية صديقة لـ C. وبهذه الحالة ينصح بالانطلاق من المتحكم PIC18F452.

١٩-٣-١ المتحكم PIC16F84A

يبين الشكل12-19 توزع الـ Pin للمتحكم PIC16F84A المصنع بالتغليف PDIP18



الشكل12-19: توزع الإشارات على مآخذ المتحكم PIC16F84A

يوجد شكل آخر من هذا المتحكم يملك 20 مأخذاً بدلا من 18 ، والمآخذ الإضافية هي واحد لـ Vss وآخر لـ Vdd. تتنوع المتحكمات من PIC بحسب الذاكرة الداخلية ضمن المتحكم إلى ثلاثة أنواع هي:

1- (PIC16CXX) : ذاكرة البرنامج من نوع (EEPROM).

2- (PIC16CRXX): ذاكرة البرنامج من نوع (ROM).

3- (PIC16FXX): ذاكرة البرنامج من نوع (FLASH).

وإن العرض أدناه حول المتحكم PIC16F84A ينطبق إلى حد كبير على متحكمات السلسلة 16F من PIC. عند التعريف بالمتحكم الصغري PIC16F84A لا نجد ضرورة لعرض بنيته الداخلية وسنكتفي بعرض مفهوم كل من إشاراته باختصار كالتالي:

- المأخذ (أو القطب Pin5) المسمى VSS: هو للتغذية (0 V) .

- المأخذ (أو القطب Pin14) المسمى VDD: هو للتغذية (+5 V) .

- المأخذ MCLR' (Pin4): مدخل تصفير (إعادة إقلاع المتحكم). ويجب أن يكون هذا المأخذ في الحالة الطبيعية موصولاً إلى (+5V). وعند وصله للصفر يتم إعادة إقلاع للمتحكم، لذلك إذا كنا في حالة تستدعي عمل إعادة إقلاع المتحكم فلابد من وصل هذا المأخذ إلى مفتاح يكون بحالته الطبيعية موصول إلى 1 منطقي، وعند الضغط على المفتاح يتم وصله إلى 0 منطقي.

- المأخذ (15,16): تستخدم لوصل كريستاله خارجية ذات تردد معين حيث إن هذه الكريستالة تقوم بإعطاء نبضات بتردد معين للمتحكم وهذه النبضات هي التي تقوم بجعل المتحكم ينتقل من تعليمة إلى أخرى حسب تسلسل التعليمات بالبرنامج. تحدد الكريستالة سرعة تنفيذ التعليمات بالبرنامج، وبالتالي تحدد سرعة المتحكم مع ملاحظة أن متحكمات PIC تتميز بأن التردد الفعلي للمعالجة فيها هو تردد الكريستالة تقسيم 4. وأن أكبر تردد للكريستالة يمكن وصله معها هو 20MHZ. وبالتالي فإن أكبر تردد فعلي

للمتحكم هو 20MHz / 4 = 5 MHz.

- المنفذ A (PORT A): وعدد مآخذه 5 تستخدم للدخل والخرج، وهي المآخذ ذوات الارقام (1 ,2, 3, 17, 18). حيث إن المأخذ رقم 3 (RA4/TOCK1) يمكن أن يكون مدخل نبضات ساعة للموقت TIMER0 إضافة لعمله كقطب دخل – خرج.
- المنفذ B (PORT B): وعدد مآخذه 8 هي الأرقام (6,7,8,9,10,11,12,13) وهذه المآخذ تستخدم للدخل والخرج، وإن المأخذ رقم (6) RB0/INT يمكن أن يكون مدخل للمقاطعة الخارجية.

١٩-٣-١-١ مزايا المتحكم PIC16F84A

- وحدة المعالجة CPU ذات تقنية RISC
- سعة ذاكرة ROM هو 1024 كلمة حيث طول كلمة التعليمات هو 14Bits
- سعة ذاكرة المعطيات من نوع RAM هو 68 Bytes x 8 Bits
- سعة الذاكرة من نوع EEPROM هو 64 Bytes .
- يوجد 4 مصادر للمقاطعة هي:
  1. مقاطعة خارجية على المأخذ RB0/INT
  2. مقاطعة تغير حالة الأقطاب RB4-RB7.
  3. مقاطعة طفحان المؤقت TIMER0.
  4. مقاطعة انتهاء كتابة المعطيات في الذاكرة EEPROM.
- يوجد 13 مأخذاً للدخل والخرج موزعة بين المنفذين A & B.
- إمكانية الدخول في وضع الراحة Sleep لتوفير الطاقة .
- عدد التعليمات 35 تعليمة بلغة الأسمبلي .
- الذواكر الموجودة في هذا المتحكم FLASH/EEPROM ذات تقنية CMOS
- يوجد مكدس بعمق 8 مستويات يستفاد منه بتخزين عنوان العودة للبرنامج الرئيسي عند تنفيذ مقاطعة أو استدعاء لبرنامج فرعي، فمثلا عندما يكون مؤشر البرنامج عند

القيمة 50، ويفرض أن التعليمة الموجودة عند هذه القيمة هي استدعاء للبرنامج الفرعي الموجود عند قيمة لمؤشر البرنامج 70 ، عندها سيعطي مؤشر البرنامج القيمة (PC = PC + 1 = 50+1 = 51)، ويتم تخزين هذه القيمة في مؤشر المكدس، وبعدها يتم إعطاء مؤشر البرنامج القيمة (PC=70) ويستمر عداد البرنامج PC بالزيادة بشكل تسلسلي PC = PC+1 = 70+1 = 71 وهكذا.

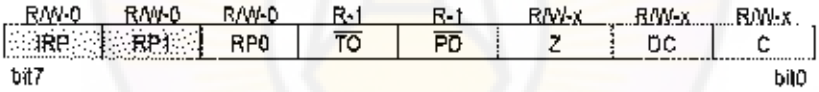
وعند الانتهاء من تنفيذ البرنامج الفرعي تكون التعليمة الأخيرة في البرنامج الفرعي هي العودة للبرنامج الرئيسي حيث يتم شحن عداد البرنامج بالقيمة (PC=51) وبعدها يتم الزيادة بالشكل التسلسلي الطبيعي للبرنامج (PC = PC+1).

١٩-٣-١-٢ مسجلات المتحكم PIC16F84A

يوجد في المتحكم الصغري مجموعة من المسجلات نستعرضها وفق التالي:

1- مسجل الحالة Status Register:

يظهر الشكل 13-19 مسجل الحالة في المتحكم PIC16F84A واسم كل من بتاته الثمانية وموضع كل بيت في المسجل، حيث:



الشكل13-19: مسجل الحالة في المتحكم PIC16F84A

- البت رقم 0 وهو علم الحمل C: ويأخذ القيمة 0 عند عدم حدوث حمل في الخانة الأخيرة الأكثر أهمية، بينما يأخذ القيمة 1 عند حدوث حمل.
- البت رقم 1 وهو علم نصف الحمل DC: ويأخذ القيمة 0 عند عدم حدوث حمل في الخانة الرابعة. وبالعكس يأخذ القيمة 1 عند حدوث حمل في الخانة الرابعة.
- مؤشر التصفير Z: ويأخذ القيمة 0 عندما تكون نتيجة عملية ما لا تساوي الصفر.
- البت 'PD': هذا البت للقراءة فقط وغير قابل للكتابة فيه، ويأخذ القيمة 0 عند استخدام الأمر Sleep، بينما يأخذ القيمة 1 عند تشغيل المايكرو، أو عند استخدام الأمر CLRWDT، حيث إن WDT يقوم بمنع المتحكم PIC من أن يدخل في حلقة مفرغة

نتيجة أخطاء برمجية أو نبضات شوكية كهربائية. يعتمد هذا البت في عمله على هزاز RC داخلي وليس خارجياً وبالتالي لا علاقة له بالكريستالة الخارجية.

– البت 'TO: هذا البت للقراءة فقط وغير قابل للكتابة فيه، ويأخذ القيمة 0 عند انتهاء مدة WDT مؤقت المراقبة. بينما يأخذ القيمة 1 عند تشغيل المايكرو أو عند استخدام الأمر CLRWDT أو الأمر Sleep.

– البت RP0: ويستخدم هذا البت ليحدد البنك المستخدم من الذاكرة كما يلي:

عند إعطائه القيمة 0 يتم اختيار البنك BANK0.

عند إعطائه القيمة 1 يتم اختيار البنك BANK1.

وهذا ينطبق فقط على المتحكم PIC16F84A لأنه يحتوي على بنكين، بينما في المتحكمات الأرقى يتم استخدام البتين RP0, RP1 لتحديد البنك المستخدم من بين أربعة بنوك ذاكرية.

– البت IRP: يستخدم في حالة العنونة غير المباشرة وهو يأخذ القيمة صفر.

2– المسجل TRISA: يحدد حالة كل بت في النافذة A هل هو دخل أم خرج . فأي بت من البتات في هذا المسجل يتم إعطاؤه القيمة 0 يكون القطب المقابل له خرجاً، وبالمقابل فإن أي بت يتم إعطاؤه القيمة 1 يكون القطب المقابل له دخلاً.

3– المسجل TRISB: هذا المسجل يحدد حالة كل بت في النافذة B هل هو دخل أم خرج . فأي بت يتم إعطاؤه القيمة 0 يكون القطب المقابل له خرجاً .

4– المسجل TMR0: وهو عبارة عن مسجل له الخصائص التالية :

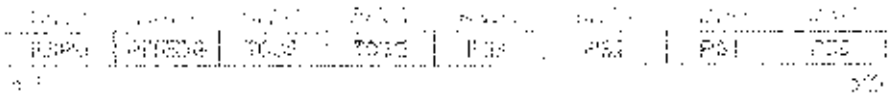
– يمكن الكتابة فيه أو القراءة منه .

– اختيار نبضات ساعة داخلية أو خارجية .

– اختيار لنوع الجبهة الصاعدة أو الهابطة عند العَد .

– يوجد مقاطعة تعتمد على طفحان هذا المؤقت .

5- مسجل الخيار Option Register



- البتات (PS0, PS1, PS2): ويتم استخدامها لتحديد معدل المقياس كالتالي:

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

فمثلا عندما تكون قيمة هذه البتات هي 010 يعني أن كل 8 نبضات على المأخذ RA4 سيزداد العداد TMR0بمقدار واحد، وأن كل 4 نبضات على نفس المأخذ سيزداد WDT بمقدار واحد .

- البت PSA: بت اختيار المقياس :

فعند إعطائه القيمة 0 يكون معدل المقياس تابعاً لـ TMR0. أما عند إعطائه القيمة 1 يكون معدل المقياس تابعاً لـ WDT.

- البت :TOSE عند اختيار المؤقت TMR0 كعداد يعّد عند النبضات على القطب RA4 فإنه يتم عن طريق هذا البت تحديد تبعية العد للنبضة المستقبلة على هذا القطب بحيث إذا كان: TOSE = 0 يتم العد عند الجبهة الصاعدة TOSE = 1 يتم العد عند الجبهة الهابطة

- البت TOCS: وهو البت الذي يحدد مصدر نبضات الساعة فيما إذا كانت داخلية أو خارجية حيث: TOCS = 0 نبضات الساعة داخلية يقوم بعدها المؤقت TIMRO أي أن المؤقت TMRO يعدّ بمثابة مؤقت TIMER.

1 = TOCS نبضات الساعة خارجية يقوم بعدها المؤقت TIMRO أي أن المؤقت TMRO يعد بمنزلة مؤقت عداد COUNTER

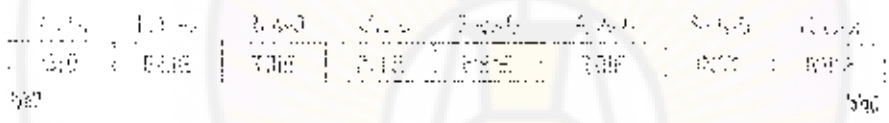
- البت INTEDG: يستخدم ليحدد تبعية المقاطعة الخارجية للجبهة الصاعدة أو الهابطة كما يلي :

INTEDG = 0 المقاطعة على المأخذ RB0 تحدث عند الجبهة الهابطة.

INTEDG = 1 المقاطعة على المأخذ RB0 تحدث عند الجبهة الصاعدة.

- البت PORT – B –PULL UP: RBPU' يستخدم لتفعيل أو عدم تفعيل مقاومات السحب على المنفذ B بحيث إذا كان RBPU' = 0 يتم تفعيل مقاومات السحب الداخلية على PORT B

6- مسجل التحكم بالمقاطعات INTCON:



- البت RBIF: وهو علم المقاطعة على الأقطاب (RB4-RB7).

حيث يأخذ القيمة 0 عند عدم حدوث مقاطعة على المأخذ (RB4-RB7). بينما يأخذ القيمة 1 عند حدوث مقاطعة على المآخذ (RB4-RB7).

- البت INTF: وهو علم المقاطعة الخارجية على القطب RB0. حيث يأخذ القيمة 0 عند عدم حدوث مقاطعة على القطب RB0.

- البت T0IF: وهو علم طفحان المؤقت TMRO. حيث يأخذ القيمة 0 عند عدم حدوث طفحان للمؤقت .

- البت RBIE: وهو بت تفعيل أو عدم تفعيل المقاطعة على المآخذ (RB4-RB7). عند إعطائه القيمة 0 لا يتم تفعيل المقاطعة على المآخذ (RB4-RB7).

بينما عند إعطائه القيمة 1 يتم تفعيل المقاطعة على المآخذ (RB4-RB7).

- البت INTE: وهو بت تفعيل أو عدم تفعيل المقاطعة الخارجية على القطب RB0. عند إعطائه القيمة 0 لا يتم تفعيل المقاطعة الخارجية على القطب RB0.

- البت TOIE: وهو بت تفعيل أو عدم تفعيل مقاطعة طفحان المؤقت TMR0 عند إعطائه القيمة 0 لا يتم تفعيل مقاطعة طفحان المؤقت TMR0 .

- البت EEIE: وهو بت تفعيل أو عدم تفعيل مقاطعة انتهاء الكتابة علىEEPROM عند إعطائه القيمة 0 لا يتم تفعيل مقاطعة انتهاء الكتابة على EEPROM.

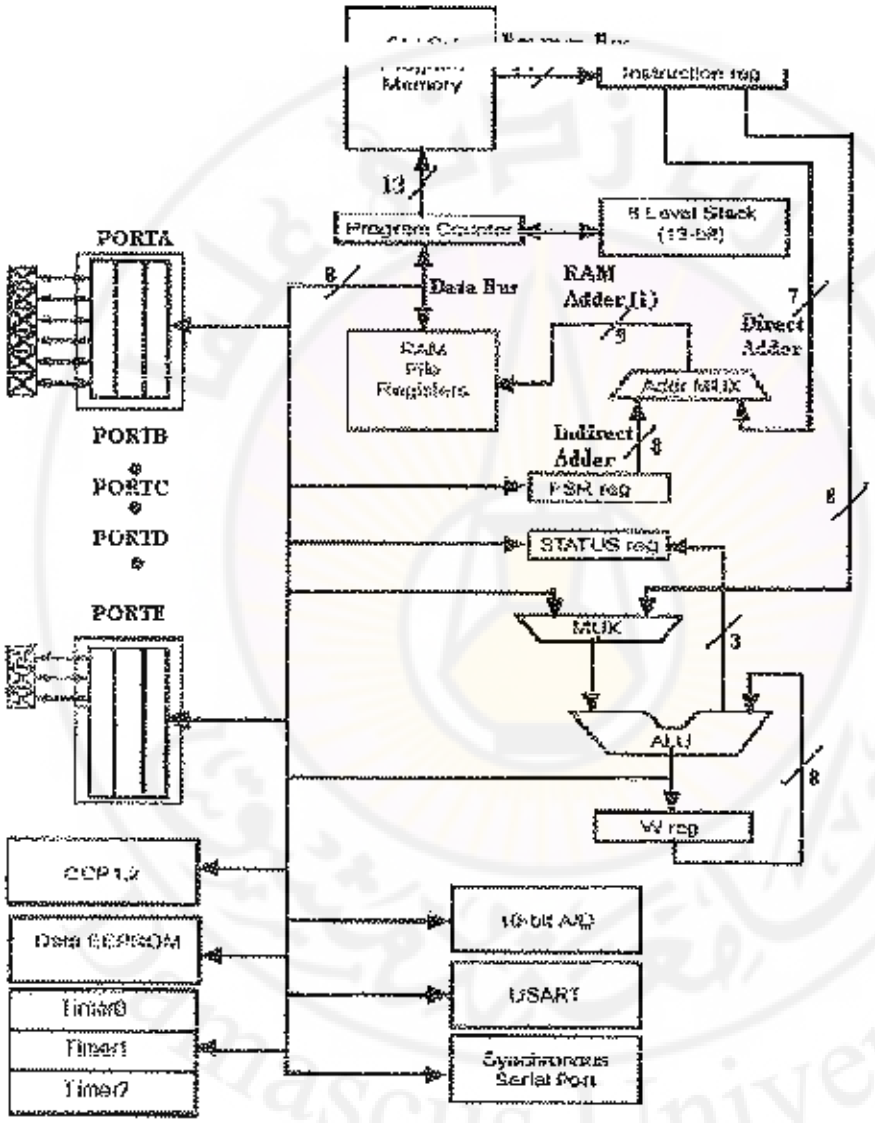
- البت (GIE): وهو بت تفعيل أو عدم تفعيل جميع المقاطعات. عند إعطائه القيمة0 يتم حجب جميع المقاطعات، ويتم تفعيلها بإعطاء هذا البيت القيمة 1.

7- المسجل PORTA: بفرض أننا قمنا بوضع النافذة A عن طريق المسجل TRISA على أنها دخل عندها فإن هذا المسجل سيحتوي على القيم المطبقة على النافذة A . وبفرض أننا قمنا بوضع النافذة A عن طريق المسجل TRISA على أنها خرج عندها فإن القيمة التي نشحن بها المسجل A ستظهر على النافذة A.

8- المسجل PORTB: بفرض أننا قمنا بوضع النافذة B عن طريق المسجل TRISB على أنها دخل عندها فإن هذا المسجل سيحتوي على القيم المطبقة على النافذة B . وبفرض أننا قمنا بوضع النافذة B عن طريق المسجل TRISB على أنها خرج عندها فإن القيمة التي نشحن بها المسجل B ستظهر على النافذة B. نكتفي عند هذا الحد من توصيف المتحكم PIC16F84A، لنشير من فئة متحكمات الـ PIC إلى متحكم آخر أكثر تطورا هو المتحكم PIC 16F877A/874A.

١٩-٣-٢ المتحكم PIC 16F877A/874A

يتميز هذا المتحكم بمعظم مزايا المتحكم السابق مع بعض التوسعات التي تلائم المتطلبات التطبيقية، ويبين الشكل 14-19 التركيبة الداخلية للمتحكم PIC16F877A.



الشكل14-19: المخطط الصندوقي للمتحكم الصغري PIC16F877A

حيث إن المنفذ B بقي بثماني بتات كما هو الحال في المتحكم PIC16F84A، أما المنفذ A فأصبح بست بتات، كما تم إضافة المنفذين C, D كل منهما بثمانية بتات تستخدم كقنواته للدخل، خرج رقمي تفرعي. وتم إضافة المنفذ E بثلاثة بتات بحيث تعمل قنواته كقنوات تماثلية. وهكذا أصبح المتحكم PIC16F877A يمتلك خمس بوابات I/O بـ 33 مأخذ من المآخذ الـ 40 للمتحكم، حيث إن:

- PA بعرض 6 bit للنقل بالاتجاهين، الاستثناء هو RA4 الذي يضاعف كدخل لواحد من المؤقتات الداخلية.
- PB بعرض 8 bit للنقل بالاتجاهين، ويمكن استخدام مآخذ هذه البوابة لاستقبال المقاطعات التقنية.
- PC بعرض 8 bit للنقل بالاتجاهين، ويمكن أن تشكل لتدعم وظائف IO التسلسلي
- PD بعرض 8 bit للنقل بالاتجاهين
- PE بعرض 3 bit النقل بالاتجاهين، ولكن يمكن تشكيله للتحكم بالبوابة D.

يصنع المتحكم الصغري PIC16F877 بالتقانة CMOS ويعمل بثلاث سرعات: 4, 8, or 20 MHz.

صمميم معمارية المتحكم هي RISC CPU تعمل بأداء عالٍ و بـ 35 تعليمة كل منها تمثل كلمة مفردة. ويمكن إجمال مواصفات المتحكم PIC16F877A بالتالي:

1- Microchip PIC16F877A Microcontroller Features

High-Performance RISC CPU

- Operating speed: 20 MHz, 200 ns instruction cycle
- Industrial temperature range (-40° to +85°C)
- 15 Interrupt Sources
- 35 single-word instructions

2- Special Microcontroller Features

- Flash Memory: 14.3 Kbytes (8192 words)
- Data SRAM: 368 bytes
- Data EEPROM: 256 bytes
- Watchdog Timer with on-chip RC oscillator
- Power-saving Sleep mode

3- Peripheral Features

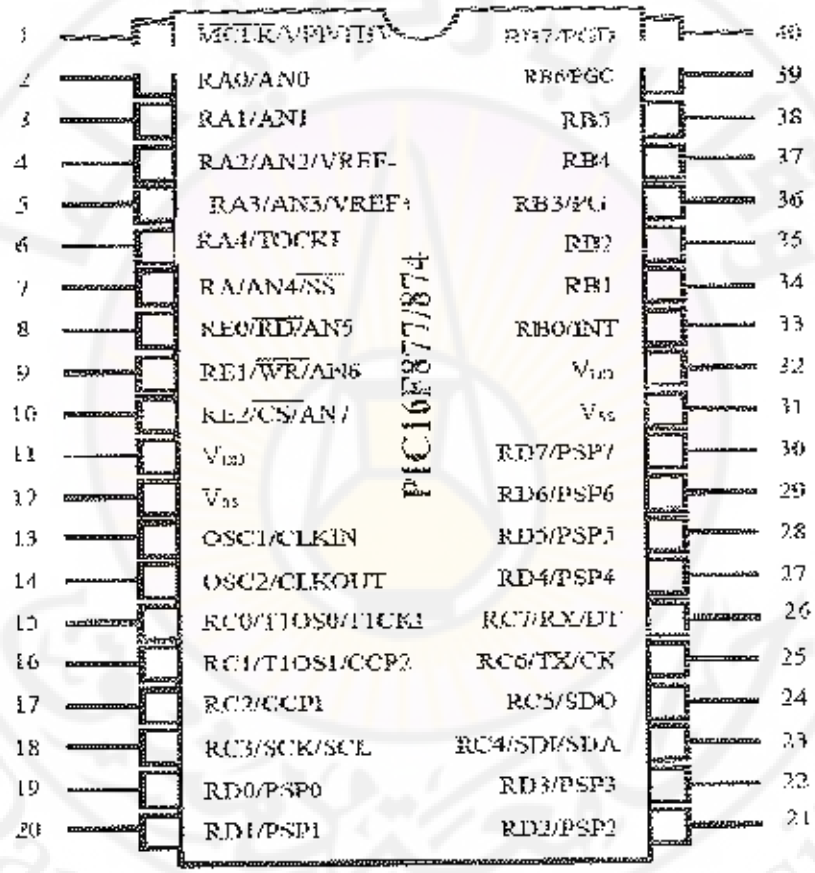
- 33 I/O pins; 5 I/O ports
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler
- Timer2: 8-bit timer/counter with 8-bit period register
- Two Capture, Compare, PWM modules
  - 16-bit Capture input; max resolution 12.5 ns
  - 16-bit Compare; max resolution 200 ns
- Synchronous Serial Port with two modes:
  - SPI Master
  - I2C Master and Slave
- USART/SCI with 9-bit address detection
- Parallel Slave Port (PSP)
  - 8 bits wide with external RD, WR and CS controls

4- Analog Features

- 10-bit, 8-channel A/D Converter
- Analog Comparator module
  - 2 analog comparators

ملاحظة: لا يقصد هنا بطول الكلمة 2 Bytes مثلاً، بل إن طول الكلمة لدى PIC يمكن أن يكون 12، 14 أو أي عدد من البتات، وهي تحديدا بالنسبة للمتحكم PIC16F877A بطول 14 bit).

المتحكم PIC 16F877A متاح بثلاثة أشكال للتغليف DIP, PLCC, QFP، نعرض منها التغليف الأكثر انتشارا DIP وفق ما هو مبين بالشكل 15-19. ويظهر الجدول 8-19 اسم كل إشارة من إشارات المتحكم PIC16F877A/874A



الشكل15-19 توزع إشارات المتحكم PIC16F877A/874A

الجدول8-19 إشارات المتحكم PIC16F877A/874A

pin	Pin name	description
1	~ MCLR	
2	RA0/AN0	PORTA.0/Analog Channel 0
3	RA1/AN1	PORTA.1/Analog Channel 1
4	RA2/AN2	PORTA.2/Analog Channel 2
5	RA3/AN3	PORTA.3/Analog Channel 3
6	RA4/T0CK1	PORTA.4/External Clock for Timer 0
7	RA5/AN4	PORTA.5/Analog Channel 4
8	RE0/AN5	PORTE.0/Analog Channel 5
9	RE1/AN6	PORTE.1/Analog Channel 6
10	RE2/AN7	PORTE.2/Analog Channel 7
11	Vdd	+3V ~ +5V
12	Vss	GND
13	OSC1/CLKIN	Oscillator Connection/Clock In
14	OSC2/CLKOUT	Oscillator Connection/Clock Out
15	RC0/T1CK1	PORTC.0/External Clock for Timer 1
16	RC1/CCP2	PORTC.1/CCP2
17	RC2/CCP1	PORTC.2/CCP1
18	RC3/SCK/SCL	PORTC.3/SCK(for SPI)/ SCL (for I^2C)
19	RD0	PORTD.0
20	RD1	PORTD.1
21	RD2	PORTD2
22	RD3	PORTD3
23	RC4/SDI/SDA	PORTC.4/SDI (for SPI)/ SDA (for I^2C)
24	RC5/SDO	PORTC.5/SDO (for SPI)
25	RC6/TX	PORTC.6/TX (for Serial com)
26	RC7/RX	PORTC.7/RX (for Serial com)
27	RD4	PORTD.4
28	RD5	PORTD.5

29	RD6	PORTD.6
30	RD7	PORTD.7
31	Vss	GND
32	Vdd	+3V ~ +5V
33	RB0/INT	PORTB.0/External Interrupt
34	RB1	PORTB.1
35	RB2/PGM	PORTB.2/Programming Input
36	RB3	PORTB.3
37	RB4	PORTB.4
38	RB5	PORTB.5
39	RB6/PGC	PORTB.6/Debugger/ICSP
40	RB7/PGD	PORTB.7/Debugger/ICSP

١٩-٣-٢-١ تنظيم الذاكرة في المتحكم PIC16F877

تقسم الذاكرة لدى المتحكم PIC16F877 إلى ثلاثة أنواع، نوعين من الذاكرة الداخلية (ذاكرة للبرنامج وأخرى للمعطيات)، حيث إن ذاكرة البرنامج بسعة 8 K Words بالتنظيم (8Kx14 bit) من نوع Flash، أما ذاكرة المعطيات فلها مصدران:

الأول 368 Bytes من نوع RAM والثاني هو 256 Bytes من نوع EEPROM.

ويمكن باختصار توصيف هذه الذواكر وفق التالي:

أولا: ذاكرة البرنامج Program memory

لدى المتحكم PIC16F877 عداد برنامج موسع بعرض 13 bit قادر على عنونة مجال ذاكري بسعة 8Kx14 bit من ذاكرة البرنامج. يتم استخدام هذه الذاكرة في المقام الأول لتخزين البرامج التي يتم كتابتها (حرقها) ليتم استخدامها من قبل المتحكم. ولدى هذا المتحكم أيضاً 8Kx 14 bit من ذاكرة فلاش التي يمكن أن تكون قابلة للمسح.

ثانيا: ذاكرة المعطيات Data memory

وهي مقسمة لبنوك متعددة تحتوي على مسجلات للأغراض العامة GPR ومسجلات لوظائف خاصة SPR. وبحسب نوع المتحكم يمكن أن تختلف، حيث إن المتحكم PIC16F877 يملك أربع بنوك Bank0, ..., Bank4 وكل بنك لـ 128 Bytes.

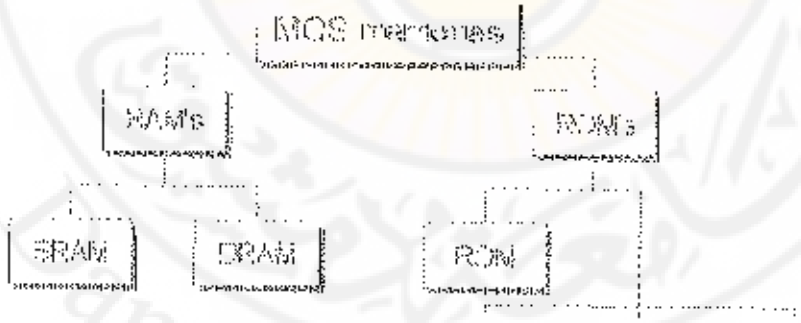
ثالثا: ذاكرة Data EEPROM

ذاكرة معطيات EEPROM والفلاش قابلة للقراءة والكتابة أثناء التشغيل العادي. هذه الذاكرة ليست مخططة mapped في مجال سجل الملف.

وعوضا عن ذلك هي معنونة بشكل غير مباشر عبر مسجلات وظيفية خاصة. يوجد هناك ستة SFRs مستخدمة للقراءة والكتابة بهذه الذاكرة هي:

EECON1, EECON2, EEDATA, EEDATH, EEADR, EEADRH

ملاحظة: بما أن معظم المتحكمات الصغرية تتعامل مع ذواكر من نوع فلاش Flash Memory، نشير هنا إلى هذا النوع من الذواكر عالي الكثافة وغير المتطاير. يظهر الشكل 16-19 موقع ذاكرة فلاش Flash من الأنواع المختلفة للذواكر الأخرى. يمكن مسح محتوى ذاكرة الفلاش وإعادة برمجتها مثل أي ذاكرة من نوع EPROM.



الشكل16-19: الأنواع المختلفة للذواكر المستخدمة بالمتحكمات الصغرية

٤-١٩ تطبيقات المتحكمات

نختم هذا الفصل عن الجوانب التقنية للمتحكمات الصغرية بموجز ملخص عن المتحكمات والمعالجات الصغرية وأهم الفوارق بينهما، لإظهار المجالات الممكنة لاستخدام كل منهما.

أصبحت الأجهزة الكهربائية والإلكترونية جزء لا يتجزأ من الحياة اليومية، ولا يكاد يخلو مكان من هذه الأجهزة، بسيطة كانت أو معقده. وقد غزت المتحكمات (المايكروكونترولر) معظم هذه الأجهزة، فأغلب الأجهزة التي حولنا تحتوي على المتحكمات ابتداء من لعب الأطفال إلى الساعات، التلفونات، الغسالات، الكاميرات، المايكروويف وصولا إلى السيارات والروبوتات وغيرها من التطبيقات في مختلف المجالات. وجميعها تحتوي على متحكمات بسيطة أو معقده.

سبق وأشرنا إلى أن الحاسوب يحوي على معالج صغري (CPU)، وحدة الحساب و المنطق (ALU) ، الذاكرتين (ROM و RAM)، وحدة تخزين البيانات (Hard Disk) ومداخل ومخارج البيانات. أما المتحكم فهو عبارة عن حاسوب صغير جدًا، بنيته الأساسية (غالبًا) ما تحتوي على كل ما يحتويه الحاسوب بما في ذلك مؤقتات، عدادات، وحدة تحويل إشارة رقمية إلى تماثلية ، بروتوكولات ... إلخ لكن بعناصر تكون أقل تعقيداً وأقل كفاءة.

تختلف المتحكمات فيما بينها بمواصفات وخصائص هذه المحتويات، إضافة إلى أنه في بعض الأحيان تكون هناك مميزات لبعض المتحكمات التي تصنع لأغراض مخصصه. وبالرغم من أن المتحكم صغير الحجم إلا أنه معقد ، لذلك يجب فهم الآلية التي يترابط فيها البرنامج مع الهاردوير الموجود.

وبما أن كل مكونات المتحكم تكون مدمجة في شريحة واحدة، يمكن التعديل عليها ، لذلك يجب اختيار المتحكم الملائم للخصائص التي يحتاجها التطبيق.

يفتقر المتحكم الصغري إلى خاصية الـ Multitasking التي تمكن الحاسوب العادي من تشغيل أكثر من برنامج في نفس الوقت.

يحتاج المتحكم مثل الحاسوب إلى برنامج يحتوي على عدة أوامر توجهه لإنجاز هدف البرنامج. ويمكن استخدامه في أكثر من مشروع بعد إعادة برمجته بالبرنامج المناسب

بحسب التطبيق. بالتالي قبل استخدام المتحكم في التطبيق، يجب ربطه بحاسوب خارجي لتتمكن من إنزال البرامج عليه ، ويمكن ربط المتحكم بالحاسوب بعدة طرق، أهمها عن طريق استخدام لوح التطوير " Development Board " خاصة، وهي عبارة عن لوحة متعددة الأغراض يمكن إدماج المتحكم فيها ، تحتوي على مبرمج مدمج، منفذ تسلسلي لربط المتحكم بالحاسوب، منافذ للطاقة ، وكذلك على عدد من الإضافات الأخرى التي تساعد في تطبيقات مختلفة للمتحكم.

ومن الخصائص العامة للمتحكم نذكر:

١- يكون المتحكم عادة بداخل جهاز آخر للتحكم به، لا يستخدم بشكل مستقل.

٢- يكون في المتحكم ما يحتاجه من الذاكرة (RAM & ROM) فهو ليس بحاجة إلى شرائح خارجية للذاكرة.

٣- يكون عمل المتحكم محدد بمهمة واحدة وتنفيذ الأوامر في برنامج واحد يكون مخزناً في ذاكرته بشكل مسبق.

٤- يكون استهلاك المتحكم من الطاقة صغيراً جدا بالنسبة للحواسيب الأخرى، فمثلا بعضها يستهلك 50 mW بينما الحاسوب العادي قد يستهلك 50 W.

الفوارق الجوهرية بين المتحكم (MCU) و المعالج (MPU) تكمن بالتالي:

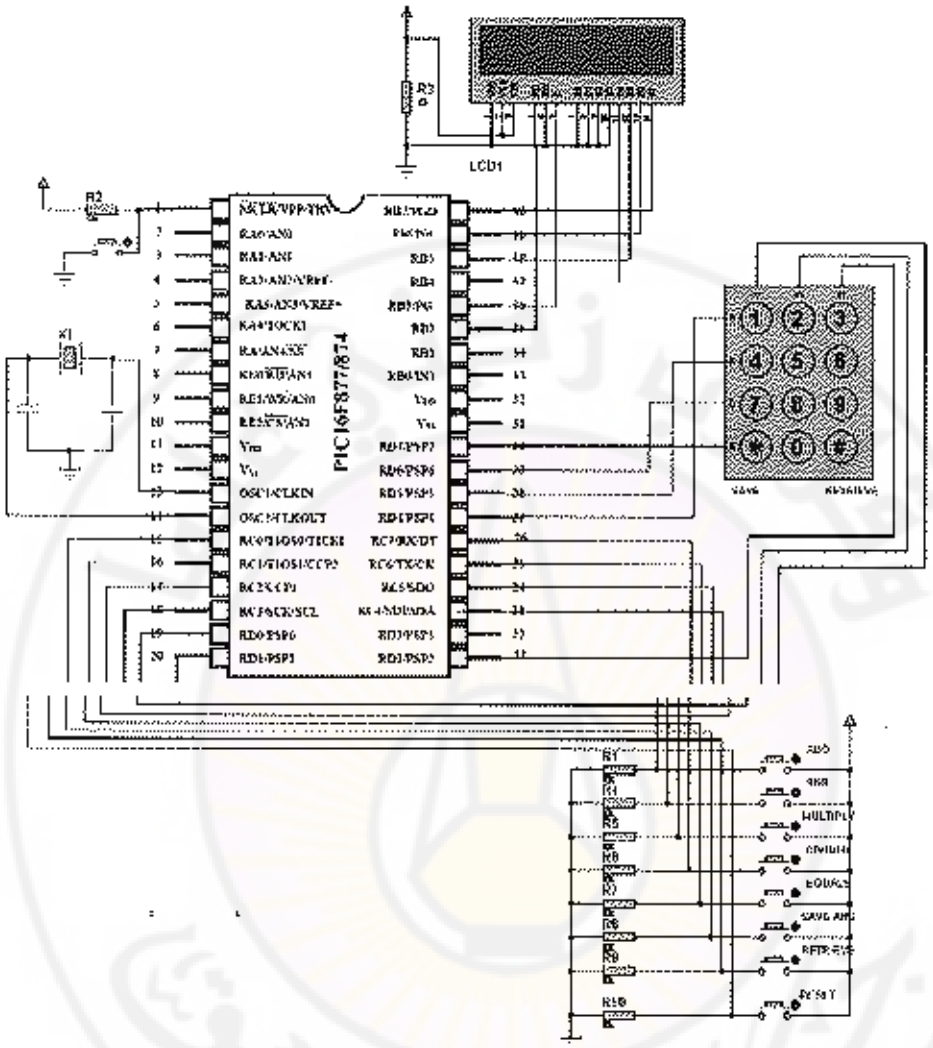
- المتحكم الصغري يحوي جميع المكونات الكافية ليعمل بشكل كامل، من وحدة معالجة إلى ذاكرة ودعم الطرفيات، بينما المعالج الصغري يحتاج إلى ربط مع ذواكر خارجية للعمل.
- وجود الذواكر داخل شريحة المتحكم الصغري يجعل لها محدودة على صعيد التوسيع والحجم، بينما إمكانية الربط الخارجي الموجودة في المعالجات الصغرية تجعل قدرات التخزين أكبر بكثير.
- نظراً لوجود خيارات واسعة بالنسبة للذاكرة في المعالجات الصغرية فإنها تعدّ الأنسب لتشغيل أنظمة التشغيل وأنظمة تشغيل الزمن الحقيقي على حد سواء، حيث إنها على الأغلب تحوي متطلبات تشغيل نظام تشغيل

من: وحدة إدارة الذاكرة MMU ، إلى إمكانية الربط مع ذواكر DRAM بينما محدودية ذاكرة المتحكم الأصغري تجعل منه محصور للتطبيقات في مجال التحكم وتنفيذ البرامج دون وجود نظام تشغيل لضبط المهام، مع إمكانية استخدامه في مجال أنظمة تشغيل الزمن الحقيقي التي لا تتطلب الكثير من الذاكرة.

- كثير من التصنيفات تتحدث عن أن المتحكمات الصغرية تحوي طرفيات أما المعالجات الصغرية فلا . تصنيفات أخرى تتحدث عن بطء المتحكم الصغري مقابل سرعات المعالج الصغري ، وأخرى تتحدث عن موضوع ربط الذواكر الخارجية. بالواقع مثل هذه التصنيفات تتلاشى تدريجياً.
- غالبا ما تكون معمارية المعالجات الصغرية هي معمارية فون نيومان و ذلك بسبب استخدام ذواكر مشتركة بين المعطيات والبرامج ، بينما تستخدم المتحكمات الصغرية بنية هارفارد لاستخدام ذواكر منفصلة بين البرامج و المعطيات.
- يعدّ استهلاك الطاقة في المعالجات الصغرية أعلى من المتحكمات الصغرية و ذلك نظراً لوجود أسباب استهلاك أعلى مثل إدارة الاتصال مع ذواكر متلاشية مثل DRAM التي تقتضي الإنعاش المستمر للمعطيات فيها.

وبخصوص تطبيقات المتحكمات، بما أن المجال واسع ولا يمكن دراستها إلا من خلال مثال محدد، نبين فيما يلي تطبيق بسيط حول استخدام إحدى المتحكمات التي تعرفنا عليها بشكل تقني على أن نظهر بالفصل اللاحق الكود المقابل لهذا التطبيق.

يظهر الشكل 17-19 توصيل ثلاثة من بوابات المتحكم بتطبيق يمثل حاسبة بسيطة تستخدم مفاتيح و 2x16 LCD keypad كنموذج عن استخدام المتحكم وآلية توصيله.



الشكل17-19: حاسبة بسيطة باستخدام المتحكم الصغري PIC16F877A

الفصل العشرون: برمجة المتحكمات

بعد أن تعرفنا بالفصل التاسع عشر المفهوم العام للمتحكم الصغري من الناحية التقنية، سنحاول بهذا الفصل تعرّف مفهوم برمجة المتحكمات بشكل مختصر. ظهرت المتحكمات الصغرية بالأساس كتطور للمعالجات عند استخدامها في بعض التطبيقات، وقد مثل ذلك محاولة ناجحة لأغراض محددة عندما كان الحجم والتكلفة واستهلاك الطاقة عوامل مهمة في الوقت الذي لا توجد فيه حاجة لقوة معالجة كبيرة. اخترنا التعرف على المتحكمات الصغرية من الناحية التقنية بالإشارة إلى الأنواع الثلاثة الأكثر انتشارا واستعمالا من المتحكمات وهي:

المتحكم 8051 الأوسع استخداما منذ عشرين عاما (ومازال يتم تطويره بشركات متعددة)، والثاني متحكم من عائلة AVR المنتج بمعرفة شركة Atmel (وهي إحدى الشركات القائدة في هذا المجال)، والثالث متحكم من إنتاج PIC (المعروفة بين الهواة لانخفاض أسعارها)، ولكل نوع من هذه المتحكمات عائلات وكل عائلة بها أفراد. يحتوي كل متحكم مثل المعالج الصغري على وحدة معالجة مركزية CPU تنجز نفس المهام الأساسية التي ينجزها المعالج بجهاز الحاسوب كالعمليات الحسابية والمنطقية وعمليات التحكم بالمداخل والمخارج I/O وغيرها.

تملك المتحكمات الصغرية خطوط عنونة وخطوط معطيات وخطوط تحكم وتملك القدرة على قراءة وتنفيذ التعليمات من ذاكرة قراءة خارجية وتحتفظ بالمعطيات المؤقتة والبرامج في ذاكرة وصول عشوائي. وبالتالي عند برمجة المتحكم لابد أيضاً من التعرف على تقنية الذاكرة المستخدمة فيه، فمثلا الشرائح ذات الذاكرة فلاش Flash تسمح بتغيير محتواها بوساطة الشحن الكهربائي مما يعطي ميزة البرمجة ضمن الدارة، بحيث لا تحتاج مثل هذه الشرائح إلى وجود نوافذ محو للذاكرة.

نشير فيما يلي باختصار إلى الجانب البرمجي للمتحكمات الصغرية بقصد إتمام موضوع المتحكمات كما هو الحال بالنسبة للمعالجات الصغرية، بالرغم من أنه يصعب توصيف برمجة المتحكمات بفصل واحد لاعتبارات عدة يأتي في مقدمتها أن لكل متحكم جملة

تعليمات خاصة به تختلف عن تعليمات متحكم آخر من نفس العائلة، وأن تعليمات المتحكمات تختلف بين عائلة وأخرى من حيث العدد والصيغة.

عند التعريف بلغة برمجية عالية المستوى يمكننا توصيف اللغة بشكل عام، أما عند التعريف ببرمجة المتحكمات الصغرية، لابد من تحديد نوع المتحكم.

وبالنظر إلى أن متحكمات الـ PIC هي الاختيار المعتاد للبدء في مجال المتحكمات على الصعيدين التقني والبرمجي (انطلاقا من إن شركة ميكروشيب اعتمدت نظام الحاسوب ذو التعليمات المخفضة RISC حيث يمكن بعدد قليل من التعليمات كتابة برنامج للاستخدام الفعلي. كما إن كل حركة (انتقال) للبيانات تعتمد على سجل واحد فقط يسمى مسجل العمل 'W') ننطلق بالتعرف على المبادئ العامة المتعلقة ببرمجة المتحكمات، من المتحكم 16F84A الذي سبق وتمت الإشارة له بالفصل السابق لاستعماله كأساس للانطلاق إلى برمجة المتحكمات الأحدث.

فمثلا طول التعليمة للمتحكمات المنتجة من Microchip متغير ويمكن تصنيفها إلى ثلاث معماريات مختلفة وفق التالي:

I- 8-bit (8-bit data bus) PICmicro, with a single accumulator:

PIC10 and PIC12: 12-bit instruction words

PIC16 series: 14-bit instruction words, one address pointer

PIC18 series: 16-bit instruction words, three address pointers

II- 16-bit (16-bit data bus) microcontrollers

PIC24: 24-bit instruction words

dsPIC: based on PIC24, plus DSP functions, such as a

single-cycle MAC (multiply-accumulate) into two 40-bit Acc.

III- 32-bit (32-bit data bus) microcontrollers:

PIC32MX series: 32-bit instructions, uses the MIPS archit.

برمجة المتحكم ليست عامة كما هو الحال بالنسبة للمعالج الصغري، بل هي دوما موجهة لتطبيق محدد.

يعتمد عمل المتحكم الصغري على مبدأ تنفيذ مجموعة التعليمات الموجودة في ذاكرة البرنامج، إذ يقوم بتنفيذ التعليمات بشكل تتابعي إلى أن يصل إلى نهاية البرنامج، وعندها يعود إلى التعليمة الأولى في بداية البرنامج ليبدأ بتنفيذ دورة أخرى من البرنامج . ولإظهار المعطيات الأساسية المتعلقة ببرمجة المتحكمات الصغرية، نعرض مقارنة عامة بين أشهر عائلات المتحكمات وفق ما هو مبين بالجدول 1-20.

الجدول 1-20: مقارنة أولية بين متحكمات من Intel, AVR, PIC

العائلة / الميزة	8051 (Intel)	AVR (Atmel)	PIC (Microchip)
البنية الأساسية	Von-Neumann	Harvard	Harvard
تقنية النواة	CISC	RISC	RISC
تردد التشغيل	24MHz	20MHz	20MHz
نبضة لكل تعليمة	12	1	4
تعليمة في الثانية	2MIPS	16MIPS	5MIPS
عدد التعليمات	215	132	32
حجم ذاكرة البرنامج	32KB	256KB	64KB
عرض الناقل	8bit	16bit	12-bit

يكتب المبرمجون تعليماتهم بلغات برمجة مختلفة، بعضها يفهمها المتحكم مباشرة وبعضها الآخر يحتاج إلى خطوات ترجمة وسيطة، حيث يمكن تقسيم لغات البرمجة المستخدمة إلى ثلاث مستويات هي:

١- لغة الآلة Machine Code: يستطيع المتحكم أن يفهم مباشرة لغة الآلة الخاصة به وهي لغته الطبيعية. تتكون هذه اللغات من سلاسل من الأصفار والواحدات وترتبط بالآلة

ارتباطا وثيقا machine-dependent . تعدّ البرمجة بلغة الآلة عملية مرهقة ولذلك يلجأ المبرمجون عادة إلى مستوى آخر من البرمجة مثل لغة المجمّع.

٢- لغة المجمّع أو لغات البرمجة منخفضة المستوى، مثل:

... ,MASM, TASM, NASM. بسبب بطء البرمجة بلغة الآلة وانتشار الحواسيب لجأ المبرمجون إلى استخدام مختصرات لمجموعات الأصفار والواحدات بلغة الآلة للتعبير عن العمليات الأولية، وقد شكلت هذه المختصرات الأساس الذي انطلقت منها لغة المجمّع، حيث تم تطوير مترجمات للبرامج سميت بالمجمعات Assemblers لتحويل البرامج من لغة المجمّع إلى لغة الآلة. بالطبع هذه اللغات أوضح وأسهل من لغات الآلة إلا إنها أبطء.

٣- لغات عالية المستوى: بالنظر إلى أن أبسط المهام قد يتطلب استخدام العديد من التعليمات بلغات المجمّع وبغية تسريع عملية البرمجة، تم تطوير لغات عالية المستوى بتعليمات تساعد على القيام بالعديد من المهام الجوهرية. تسمح لغات البرمجة عالية المستوى للمبرمجين بكتابة تعليمات قريبة من العبارات باللغة الانكليزية وتحتوي على الرموز الرياضية شائعة الاستخدام. اصطلح على تسمية البرامج التي تحول نصوص البرامج المكتوبة بهذه اللغات إلى لغة الآلة بالمترجمات Compilers. ويمكن تقسيم لغات البرمجة عالية المستوى إلى ثلاثة أنواع هي:

- لغات وظيفية، مثل: Basic, Pascal, C, ...
- لغات إجرائية، مثل C++/Embedded C...
- لغات رسومية، مثل Lab VIEW

مما لا شك فيه أن لغة الأسمبلي هي أكثر فعالية من غيرها من اللغات ذات المستوى الأعلى عندما يتعلق الأمر بالتعامل المباشر مع وظائف وسلوك المتحكم الصغري، ولكن البرمجة بلغة الأسمبلي تحتاج إلى مستوى عالٍ من الخبرة في التركيبة التقنية للمتحكم لتوظيف لغة الأسمبلي بشكل فعّال، كما أن كتابة برنامج بلغة الأسمبلي تحتاج إلى أضعاف مضاعفة من الوقت لبناء تطبيق محدد.

تتمثل أساسيات برمجة المتحكم الصغري بلغة الأسمبلي أي كان نوع المتحكم بالتالي:

١- المتحكم لديه مجموعة تعليمات محددة يمكنه تعرّف كل تعليمة وتنفيذها.

٢- البرنامج الذي يقوم المتحكم بتنفيذه هو قائمة من التعليمات مأخوذة من مجموعة تعليماته يقوم بقراءتها بالنظام الثنائي من ذاكرة البرنامج الخاصة به.

٣- يعمل المتحكم من خلال تعليمات البرنامج ويفعل ما تخبره به كل تعليمة على حده.

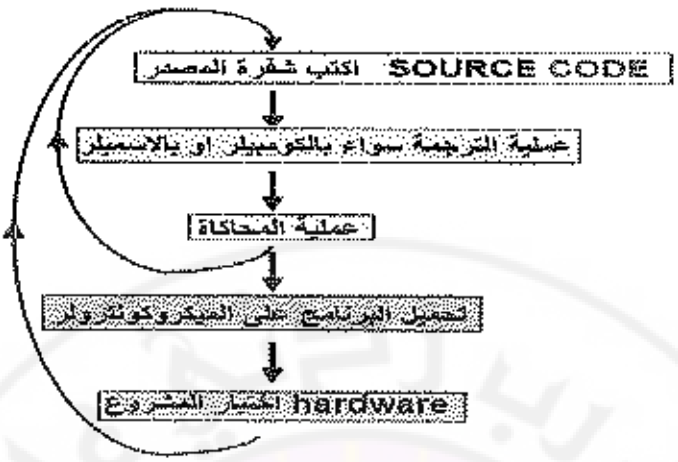
٤- يجب على المبرمج إيجاد طرق أو وسائل لتجزئة وترجمة أفكاره على شكل خطوات يمكن للمتحكم القيام بها، حيث كل خطوة في جوهرها يجب أن تكون تعليمة من مجموعة تعليمات المتحكم.

يسمى برنامج الأسمبلي الذي يقوم المبرمج بكتابته ببرنامج المصدر Source code. يحتاج المستخدم إلى برنامج آخر لترجمة برنامج المصدر إلى لغة الآلة حتى يتمكن الحاسوب من تنفيذ هذا البرنامج يسمى بالاسمبلر Assembler. يوجد نوعان من الاسمبلر هما: أسمبلر طبيعي (محلى) Native أسمبلر متقاطع.

النوع الأول: Native Assembler يعمل في جهاز الحاسوب ويولد كود الآلة الذي ينفذ في نفس الحاسوب أو في حاسوب آخر له نفس مجموعة التعليمات النوع الثاني: Cross Assembler يعمل في جهاز حاسوب ولكنه يقوم بتوليد شفرة الآلة والتي سيتم تنفيذها من خلال أجهزة الحاسوب التي لديها مجموعة تعليمات مختلفة.

أي أن البرمجة بلغة الاسمبلى يجب أن توضع في السياق الأوسع وفق ما هو مبين بالشكل 2-1 مع الإشارة إلى أنه بعد الانتهاء من توصيل دارة المتحكم، تتم محاكاة الدارة للتأكد من أنها تعمل بشكل جيد.

بعد كتابة الكود المقابل لأي تطبيق، يجب شحنه بالمتحكم ليقوم بتنفيذه، وهذا البرنامج يوضع في ذاكرة غير متطايرة Non-Volatile تكون عادة أحد أنواع الذاكرة ROM . الميزة الجيدة للمتحكم من الناحية البرمجية هي أننا نستطيع بعد برمجته انتزاعه من المبرمج ووضعه أينما شئنا في أي داره، فهو يحتفظ بالمعلومات لمدة طويلة وبدون كهرباء ناهيك عن قدرته الفائقه في تنفيذ ملايين الأوامر في الثانية حسبما نريد.



الشكل 1-20: طريقة تطوير برنامج بسيط بلغة الأسمبلي

للبرمجة بلغة الأسمبلي بعض السلبيات نذكر منها :

١- يجب أن يكون المبرمج على علم ببنية الجهاز الذي سينفذ البرنامج عليه.

٢- البرنامج المكتوب بلغة الأسمبلي (وخاصة الطويل) يصعب على الكثيرين فهمه فيما عدا مؤلفه.

٣- إنتاجية برمجية غير مرضية، وخاصة لبرمجة المشاريع الكبيرة لان المبرمج بحاجة للعمل بمنطق برمجي في مستوى منخفض جدا.

تستلزم عملية البرمجة بلغة الأسمبلي لأي متحكم، العودة إلى تعليمات المتحكم نفسه وأن البرنامج المكتوب لمتحكم ما بلغة الأسمبلي قد لا ينفذ من قبل متحكم آخر ما لم يكن من نفس الشركة المنتجة للمتحكم.

يكتب البرنامج عادة على حاسوب شخصي بلغة التجميع (الأسمبلي) الخاصة بالمتحكم بوساطة محرر نصوص ثم يجري تجميعه ليتم تحويله إلى رموز تمثل تعليمات البرنامج بلغة الآلة لذاك المتحكم.

بعد كتابة الملف الذي يمثل البرنامج الذي نود من المتحكم أن يقوم بتنفيذه ممثلا بلغة الآلة الخاصة به يتم استخدام جهاز يدعى بالمبرمجة Programmer لنقل البرنامج من الحاسوب وتخزينه داخل المتحكم، والمبرمجة توصل مع الحاسب عبر أحد منافذه (كالمنفذ

التسلسلي RS232 والمسمى عادة COM أو منفذ الطابعة المسمى عادة LPT أو منفذ USB أو غيرها) وترفق مع برنامج Software يقوم بالتخاطب معها وقيادتها ويتيح إجراء العديد من المهام على ذاكرة المتحكم.

وهنا نشير إلى أنه يصعب الدخول بتفاصيل أي من لغات البرمجة للمتحكمات الصغرية بغض النظر عن نوعها إن كانت منخفضة المستوى مثل لغة الأسمبلي أو عالية المستوى مثل C. ونكتفي هنا بالإشارة إلى بعض الملاحظات حول برمجة المتحكمات بإحدى لغات البرمجة عالية المستوى لخروج الموضوع عن نطاق هذا الكتاب.

٢٠-١ برمجة المتحكمات الصغرية بلغة عالية المستوى

لتجنب مشاكل البرمجة بلغة الأسمبلي وبقصد الاستفادة من مهارات مطوري البرمجيات الحاسوبية، والحد من طابع تعقيدات برمجة الكيان الصلب يتم اللجوء إلى برمجة المتحكمات باستخدام لغات عالية المستوى، حيث إن العبارة الواحدة Statement في اللغات عالية المستوى غالبا ما تناظر عشرات التعليمات من لغة الأسمبلي. كما ويمكن للمبرمج العمل بمنطق برمجي عالي المستوى، الأمر الذي يجعل مهمة البرمجة أسهل بكثير. من مجمل اللغات عالية المستوى التي تبرمج بها المتحكمات نشير فقط إلى لغة البرمجة C و C++ لأنها استخدمت على نطاق واسع في برمجة المايكروكونترولر في الصناعة وإن معظم مصنعي المايكروكونترولر لهم سي كومبايلر C compilers. يسمى البرنامج المكتوب بلغة عالية المستوى أيضاً بشفرة المصدر Source code. ويحتاج إلى مترجم (Compiler) لترجمته إلى تعليمات لغة الآلة. ويسمى خرج الكومبايلر أيضاً بشفرة الهدف Object code.

وكما يوجد أسمبلر متقاطع يوجد كومبايلر متقاطع، أي يعمل على جهاز ولكن يترجم التعليمات لتنفذ على جهاز آخر مختلف معه في التعليمات.

وبخصوص لغة البرمجة C فإنها تتصف باستقلالها عن البنية الصلبة، إضافة إلى كونها ذات تصميم يمكن استخدامها لكتابة برامج قابلة للنقل بين معظم الحواسيب. تم مع مطلع الثمانينيات من القرن الماضي تطوير لغة البرمجة C++ التي تعدّ امتداد للغة C ونسخة

نص البرنامج التي تساعد على فهمه.

يوجد طريقة أخرى لإضافة الملاحظات إلى برنامج C++ تتمثل باستخدام تعليقات وحيدة السطر، التي تبدأ بالرمزين // وتنتهي بانتهاء السطر.

يستخدم الشكل الأول للتعليقات متعددة الأسطر، أما النوع الثاني فيستخدم للتعليقات القصيرة ذات السطر الواحد.

أما السطر #include <stdio.h> فيمثل توجيه للمعالجة لمرحلة ما قبل الترجمة الفعلية، حيث كل سطر يبدأ بالإشارة # تتم معالجته قبل الترجمة. ويخبر هذا السطر مرحلة ما قبل الترجمة بضم محتوى الملف الرأسي (ذي الإمتداد h.) المحتوي على العمليات الخاصة بالدخل/خرج إلى نص البرنامج. ويجب ضم هذا الملف إلى أي برنامج يحتوي عمليات خرج على الشاشة أو دخلا لها من لوحة المفاتيح. والسطر main() يعد جزء من أي برنامج C والقوسان بعد كلمة main تعني أن main هي كتلة برمجية تسمى تابعا (دالة) Function.

تحتوي برامج C على تابع أو أكثر، إحداها يجب أن يكون التابع الرئيس main. يعبر القوس الكبير الأيسر { عن بداية جسم أي تابع ويعبر القوس الكبير الأيمن } عن نهاية التابع. بالطبع يجب أن تنتهي كل تعليمة بفاصلة منقوطة (;) التي تسمى بعلامة نهاية التعليمة Statement terminator.

ومن أهم ميزات C++ استخدام الأصناف وهي الآلية التي تستخدم لإنشاء الكائنات. يصرح عن صنف باستخدام الكلمة المحجوزة class وله الشكل العام التالي:

```
Class class-name {
        // private function and variable      متغيرات ودوال خاصة
public:
        // public function and variable       متغيرات ودوال عامة
} object-list;
```

العنصر object-list اختياري وأن class-name أيضاً اختياري تقنيا إلا أنه في الواقع ضروري دائما، وذلك لان class-name تصبح اسم نوع جديد يستخدم للتصريح عن كائنات من الصنف.

تدعى المتحولات والدوال المصرح عنها داخل تصريح الصنف بأعضاء الصنف class members افتراضيا وهي خاصة بالصنف. هذا يعني أن الوصول إليها ممكن فقط من قبل الأعضاء الأخرى في هذا الصنف.

للتصريح عن أعضاء عامة في الصنف تستخدم الكلمة المحجوزة public متبوعة بنقطتين. يمكن الوصول إلى جميع الأعضاء المصرح عنها بعد public من قبل أعضاء الصنف الأخرى وأيضاً من قبل أي جزء آخر من البرنامج الذي يحتوي الصنف. من أكثر العمليات التي نصادفها خلال استخدام المتحكمات هي عمليات الإدخال من طرفية للمتحكم والإخراج من المتحكم إلى طرفية أخرى.

وهنا نشير بداية إلى أن لغة C++ هي مجموعة تشمل C، أي أن جميع عناصر اللغة C موجودة أيضاً في C++ وبالتالي جميع برامج C هي برامج C++ بشكل افتراضي. ربما تكون الميزة الأكثر شيوعا في C++ والمستخدمة من قبل المبرمجين هي طريقتها في الإدخال والإخراج إلى الطرفية Console I/O.

ينجز الإدخال والإخراج في C++ باستخدام عوامل إدخال وإخراج I/O operators عوضا عن دوال إدخال وإخراج I/O function .

عامل الإدخال هو << وعامل الإخراج هو >> . تستخدم هذه العوامل في C للإزاحة نحو اليمين ونحو اليسار على الترتيب.

تحتفظ هذه العوامل في C++ بنفس معناها القديم، لكنها أيضاً تأخذ دورا إضافيا في عملية الإدخال والإخراج. فمثلا عبارة C++ التالية:

```
cout<< "this string is output to the screen. \n";
```

تؤدي إلى إظهار السلسلة الرمزية إلى الشاشة، حيث إن cout هي مجرى stream معرف مسبقا يرتبط تلقائيا بطرفية الحاسوب عند بدء تنفيذ برنامج C++. وهو مشابه لـ stdout في لغة C. وكما في C يمكن في C++ إعادة توجيه دخل وخرج الطرفية. نكتفي بهذا

القدر من الإشارة إلى لغة C كواحدة من لغات البرمجة عالية المستوى، لان توصيف أي لغة مثل لغة C بشكل كامل قد يلزمه عدد من الصفحات يزيد عن عدد صفحات هذا الكتاب، مع العلم أن اللغات عالية المستوى ليست مثالية أيضاً، فمن المشاكل الرئيسية التي تواجهها هذه اللغات هي أن شفرة الآلة الناتجة من برنامج مكتوب بلغة عالية المستوى تكون أطول ولا تستطيع أن تعمل بسرعة لغة الأسمبلي. ولهذا السبب مازال الكثير من البرامج التي يكون فيها الزمن حرج تكتب بلغة الأسمبلي.

وبما أنه لا يمكن كتابة برنامج بدون تطبيق محدد، نختم هذا الفصل بمثال حول كتابة code لإحدى المتحكمات الصغرية، تحديدا الكود المقابل للتطبيق الوارد في نهاية الفصل السابق حول استخدام المتحكم الصغري PIC16F877 بآلة حاسبة بسيطة، كعينة عن برمجة المتحكمات بلغة عالية المستوى وفق ما هو مبين في الجدول 2-20.

الجدول2-20: برنامج للمتحكم PIC16F877 بلغة عالية.

```
program
dim text as char[20]
dim kp, cnt, operator_sign as byte
dim txt as string[5]
dim    number,    number_temp,    number_temp2,    save_number,
save_number2  as word

main:
  TRISB = 0
  Keypad_Init(PORTD)              ' PORTB is output
  Lcd_Init(PORTB)                 ' Initialize LCD on PORTB
  Lcd_Cmd(Lcd_CURSOR_OFF)         ' Turn off cursor
  Lcd_Cmd(LCD_CLEAR)
  text    "--------COE 5A--------"
  Lcd_Out(1, 1, text)             ' Print text at LCD
  delay_ms(5000)
  Lcd_Cmd(LCD_CLEAR)

while TRUE
    kp = 0
    '--- Wait for key to be pressed
    while kp = 0
   '--- un-comment one of the keypad reading functions
      kp = Keypad_Released
```

```
      if (Button(PORTC,7,1,1)) then ' If PORTB.7
          is pressed Plus
      kp = 16
      end if
      if (Button(PORTC,6,1,1)) then ' If PORTB.7
       is pressed Minus
      kp = 17
      end if
      if (Button(PORTC,5,1,1)) then ' If PORTB.7
        is pressed multiply
      kp = 18
      end if
      if (Button(PORTC,4,1,1)) then ' If PORTB.7
         is pressed divide
      kp = 19
      end if
      if (Button(PORTC,3,1,1)) then ' If PORTB.7
        is pressed equals
      kp = 20
      end if
      if (Button(PORTC,2,1,1)) then ' If PORTB.7
        is pressed equals
      kp = 21
      end if
      if (Button(PORTC,1,1,1)) then ' If PORTB.7
        is pressed equals
      kp = 22
      end if
      if (Button(PORTC,0,1,1)) then ' If PORTB.7
        is pressed equals
      kp = 23
      end if
    wend

  select case kp      ' Prepare value for output
    case 1 number = 1
    case 2 number = 2
    case 3 number = 3
    case 5 number = 4
    case 6 number = 5
    case 7 number = 6
    case 9 number = 7
    case 10 number = 8
    case 11 number = 9
    case 13 save_number = number_temp
            Lcd_Cmd(LCD_CLEAR)
            cnt = 4
```

```
    case 14 number = 0

    case 15 WordToStr(save_number, txt)
            number_temp = save_number
            Lcd_Out(1, 12, txt)
            cnt = 4

    case 16 Lcd_Cmd(LCD_CLEAR)
            operator_sign = 1 '---------------sign operator
            Lcd_Out(2, 1, "+")
            number_temp2 = number_temp
            WordToStr(number_temp2, txt)
            Lcd_Out(2, 12, txt)
            cnt = 4

    case 17 Lcd_Cmd(LCD_CLEAR)
            operator_sign = 2 '---------------sign operator
            Lcd_Out(2, 1, "-")
            number_temp2 = number_temp
            WordToStr(number_temp2, txt)
            Lcd_Out(2, 12, txt)
            cnt = 4

    case 18 Lcd_Cmd(LCD_CLEAR)
            operator_sign = 3 '---------------sign operator
            Lcd_Out(2, 1, "*")
            number_temp2 = number_temp
            WordToStr(number_temp2, txt)
            Lcd_Out(2, 12, txt)
            cnt = 4

    case 19 Lcd_Cmd(LCD_CLEAR)
            operator_sign = 4 '---------------sign operator
            Lcd_Out(2, 1, "/")
            number_temp2 = number_temp
            WordToStr(number_temp2, txt)
            Lcd_Out(2, 12, txt)
            cnt = 4

    case 20
            if operator_sign = 1 then
               number_temp = number_temp2 + number_temp
               number_temp2 = 0
            end if
            if operator_sign = 2 then
               number_temp = number_temp2 - number_temp
               number_temp2 = 0
```

```
            end if
            if operator_sign = 3 then
               number_temp = number_temp2 * number_temp
               number_temp2 = 0
            end if
            if operator_sign = 4 then
               number_temp = number_temp2 / number_temp
               number_temp2 = 0
            end if

            Lcd_Cmd(LCD_CLEAR)
            WordToStr(number_temp, txt)
            Lcd_Out(1, 12, txt)
            operator_sign = 0
            cnt = 4

    case 21 save_number2 = number_temp
            Lcd_Cmd(LCD_CLEAR)
            cnt = 4

    case 22 WordToStr(save_number2, txt)
            number_temp = save_number2
            Lcd_Out(1, 12, txt)
            cnt = 4

    case 23 number_temp   0
            number_temp2 = 0
            save_number = 0
            save_number2 = 0
            Lcd_Cmd(LCD_CLEAR)
            cnt = 4
    end select
   inc(cnt)

select case cnt
  case 1 number_temp = number
         WordToStr(number_temp, txt)
         Lcd_Out(1, 12, txt)

  case 2 number_temp = (number_temp * 10) + number
         WordToStr(number_temp, txt)
         Lcd_Out(1, 12, txt)

  case 3 number_temp =  (number_temp * 10) + number
         WordToStr(number_temp, txt)
         Lcd_Out(1, 12, txt)
```

```
  case 4 Lcd_Cmd(Lcd_Clear)
         number_temp = number
         WordToStr(number_temp, txt)
         Lcd_Out(1, 12, txt)
         cnt = 1
  case 5 cnt   0
end select
  wend
end.
```

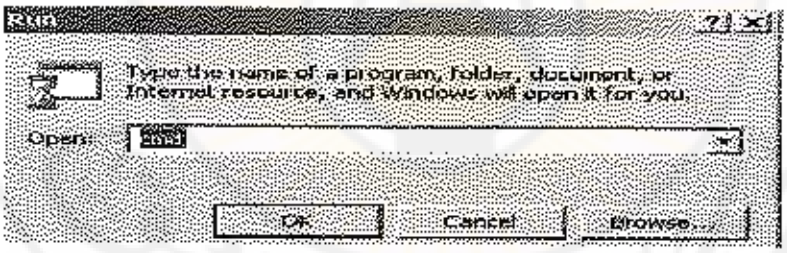
يمكن كتابة الكود المبين أعلاه بلغة الأسمبلي لكنه طويل، علما أن الاسمبلر لميكروشيب يسمى MPASM وهو من النوع Cross Assembler مصمم ليعمل على الحاسوب الشخصي لترجمة برامج لغة الأسمبلي للميكروكونترولر من طراز PIC.

الجزء الخامس: القسم العملي

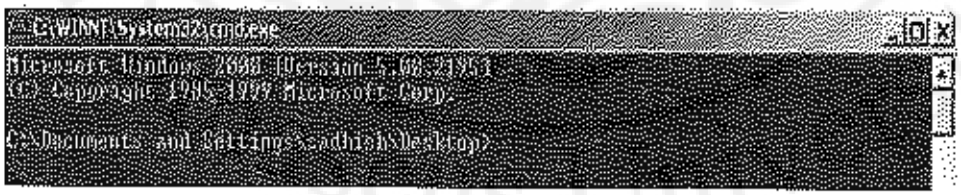
يتضمن الجانب العملي لمقرر المعالجات والمتحكمات الصغرية تطبيقات تقنية وبرمجية يرتبط نوع التطبيقات التقنية بنوع المعالج أو المتحكم الصغري ودارات الدعم المستخدمة وبالتالي يترك هذا الجانب للتطبيق المباشر في المخبر بحسب نوع العناصر المتوفرة. أما بخصوص الجانب البرمجي، وبالنظر لتركيزنا في هذا الكتاب على توصيف معالجات Intel 8086 Family، يمكننا تطبيق بعض البرامج الواردة بالجزء الثالث من الكتاب أو كتابة بعض البرامج الجديدة بحسب نوع التطبيق، بمعدل برنامج واحد على الأقل بكل جلسة (ونترك الجانب البرمجي العائد للمتحكمات الصغرية نظرا إلى عدم تحديد متحكم). نورد فيما يلي بعض البرامج لأمثلة متنوعة حول إجراء بعض العمليات الحسابية والتعامل مع الملفات والسلاسل أو تعريف البوابات وغيرها من البرامج التي يمكن اختبار أي منها بالجانب العملي للمقرر.

ولربط المعالج الصغري 8086 مع الحاسوب الشخصي PC نطبق الخطوات التالية:

الأولى: من شريط المهام لـ PC نذهب لقائمة START لاختيار RUN.



الثانية: نكتب "cmd or command" ونضغط OK لنحصل على مؤشر الـ DOS



الثالثة: نخرج من المجلد الحالي ونكتب CD\

الرابعة: نكتب الآن CD 8086 (للدخول إلى داخل المجلد الجزئي للمعالج 8086

الخامسة: نكتب (to enter into the program window) EDIT

السادسة: نشكل ملفاً جديداً لإدخال برامج المعالج 8086

السابعة: نخزن بعد ذلك البرنامج بالإمتداد "ASM" ونذهب إلى نافذة الأوامر

الثامنة: قبل الدخول إلى البرنامج ندخل بشكل مسلم به العبارات التالية:

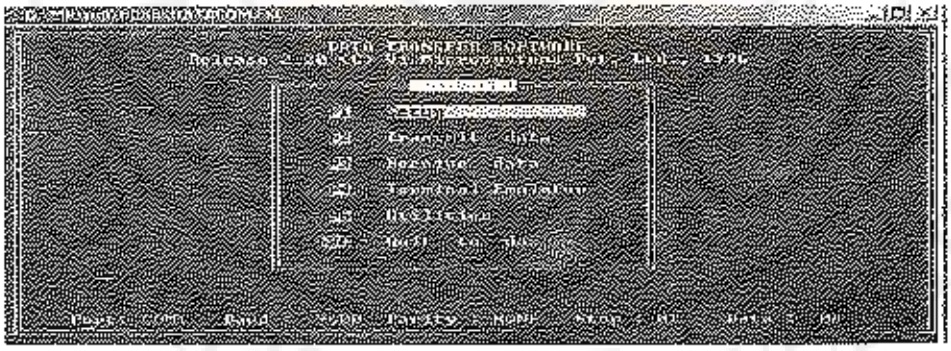
```
CODE SEGMENT
        ASSUME CS: CODE, DS: CODE
        ORG (starting address)
        ..... Program....
        HLT
        CODE ENDS        END
```

التاسعة: C:\8086\> MASM filename, ,;(to create the opcode and Machine language)

C:\8086\> LINK filename, ,;(to create the executable file)

C:\8086\> DEBUG filename.exe (to create binary file for downloading from host to 8086)

العاشرة: نكتب C:\8086\>DATACOM وتظهر قائمة DATACOM كما بالشكل



من ضمن هذه القائمة نختار "Transmit data" ونضغط enter *

الحادية عشر : نكتب اسم الملف بالإمتداد ".BIN"

الثانية عشر : نضغط enter ثلاث مرات.

وفيما يلي نعرض مجموعة من البرامج الإضافية للتطبيق والاختبار في المخبر.

مثال1: اكتب برنامج بلغة الأسمبلي للمعالج 8086 لإظهار رسالة "GOOD MORNING" على الشاشة

ALGORITHM:

1. Initialize the data segment and the message to be displayed.
2. Set function value for display.
3. Point to the message and run the interrupt to display the message

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
MSG DB 0DH, 0AH, "GOOD MORNING" , 0DH, 0AH, "$"
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS, AX
        MOV AH, 09H
        MOV DX, OFFSET MSG
        INT 21H
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

مثال2: اكتب برنامج بلغة الأسمبلي لنسخ سلسلة كلمات من موضع لاخر

ALGORITHM:

- Load the source and destination index register with starting and the ending address respectively.
- Initialize the counter with the total number of words to be copied.
- Clear the direction flag for auto incrementing mode of transfer.
- Use the string manipulation instruction MOVSW with the prefix REP to copy a string from source to destination.

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
SOURCE EQU 2000H
DEST EQU 3000H
COUNT EQU 05H
DATA ENDS
CODE SEGMENT
START:    MOV AX, DATA
          MOV DS, AX
          MOV ES, AX
          MOV SI, SOURCE
          MOV DI, DEST
          MOV CX, COUNT
          CLD
REP       MOVSW
          MOV AH, 4CH
```

INT 21H

CODE ENDS

END START

INPUT:		OUTPUT:	
2000	48	3000	48
2001	84	3001	84
2002	67	3002	67
2003	90	3003	90
2004	21	3004	21

مثال3: اكتب برنامج بلغة الأسمبلي لإيجاد واستبدال كلمة من سلسلة

ALGORITHM:

- Load the source and destination index register with starting and the ending address respectively.
- Initialize the counter with the total number of word to be copied
- Clear the direction flag for auto incrementing mode of transfer.
- Use the string manipulation instruction SCASW with the prefix REP to search a word from string.
- If a match is found (z=1), replace the old word with the current word in destination address. Otherwise, stop.

ويكون البرنامج على النحو التالي:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW 53H, 15H, 19H, 02H
REPLACE EQU 30H
```

```
        JMP   L2
L1      INC  BX
L2      LOOP  L3
        MOV  [1300], BX
        MOV  [1302], DX
        HLT
CODE ENDS
END
```

Before execution

Address	Data		Address	Data
1100	1111		1200	1111
1102	2222		1202	2222
1104	3333		1204	4444
1106	4444		1206	6666
1108	5555		1208	8888
Source index			Destination index	

After execution

Address	Data
1300	0002
1302	0003

مثال5: اكتب برنامج بلغة الأسمبلي لتهيئة دارة الدعم 8255 بحيث تكون البوابة A دخل والبوابة B خرج بالنمط 0 (mode0)

ALGORITHM:

- Load the control word to the control registers
- Load the data in the A register
- Put that data on the port C
- Stop

PROGRAM:

```
MOV AL, 90  ; load the data 90 in location AL
OUT C6, AL  ; out the data to C6 port address
IN AL, C0   ; get the data from C0 and put it in location AL
OUT C2, AL  ; output the data from AL register to port addr. C2
HLT         ; Stop
```

When the input is 01 in the swich

Port B before execution:

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0	0	0	0	0	0	0	0

Port B after execution:

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0	0	0	0	0	0	0	1

مثال6: اكتب برنامج بلغة الأسمبلي لقراءة حرف ASCII من لوحة المفاتيح عبر البوابة A للدارة 8255 عندما البت PC4 من البوابة PC يكون بالصفر. وافترض أن العنوان الأساسي للدارة هو 20H

```
PORTA      EQU   20H
PORTC      EQU   22H
CONTROL  EQU     23H
READ PROC NEAR
   MOV AL, 98H    ; 1001 1000
   OUT CONTROL, AL    ; Initialize PORTS
READ1:
   IN AL, PORTC    ; Is Strobe PC4 Low?
   TEST AL, 10H    ; 0001 0000
   JNZ READ1
   IN AL, PORTA    ; Read ASCII Character
   RET
READ ENDP
;++++++++++++++++++++++++++++++++++++++++++++++
```

مثال7: اكتب برنامج بلغة الأسمبلي (ALP) لتوليد موجة مربعة من خلال ربط دارة الدعم 8253 بالمعالج الصغري 8086.

ALGORITHM:

- Select the channel and load the count
- Use the proper control word and control register
- Stop

PROGRAM:

```
MOV AL, 36  ; move the control word to the control register
OUT CE, AL
MOV AL, 10  ; load the count to the proper address
OUT C8, AL
MOV AL, 00
OUT C8, AL
HLT         ; Stop
```

مثال8: اكتب برنامج بلغة الأسمبلي لربط دارة الدعم المبرمجة 8251 مع المعالج 8086 لارسال واستقبال المعطيات بشكل تسلسلي

ALGORITHM:

- Initialize the 8253 and 8251
- load the control word in the control register
- Give the input in the main program
- Start a separate program in another memory location to get the transmitted data.
- Give the proper control word for the reception
- Stop

PROGRAM

Transmission:

```
MOV AL, 36      ; move the control word to the control register
OUT CE, AL
MOV AL, 10      ; get the data and output at the port C
OUT C8, AL
```

```
MOV AL, 00
OUT C8, AL
MOV AL, 4E          ; initialize the control word from 8251
OUT C2, AL
MOV AL, 37
OUT C2, AL
MOV AL, 55          ; transmit the data
OUT C0, AL
INT2                ; Reset the processor
```

Reception:

```
IN AL, C0           ; Receive the data
MOV BX, 1250  ; store in the location
MOV [BX], AL
INT2                ; Reset the processor
```

Result:

The ALP is written and executed

The transmitted data is on location (1255) "55"

The received data is on location (1250) "55"

ونكمل فيما يلي كتابة بعض البرامج المتفرقة الأخرى التي تأتي كتطبيق لتقنية معينة ونحاول تحليل خوارزمية الحل خطوة بخطوة وشرح الكود المقابل.

مثال9: اكتب برنامج لإيجاد الأعداد المكونة من ثلاثة أرقام (ضمن المجال 100 وحتى 999) التي يكون فيها العدد مساوياً لمجموع مكعبات خاناته، مثال العدد 153 حيث: $(153 = 3^3 + 5^3 + 1^3 = 27 + 125 + 1)$

البرنامج:

```
SEGMENT        DATA
               First Digit        DW1
               Second Digit       DW0
               Third Digit        DW0
DATA ENDS
Working-Storage          SEGMENT
                         DW   100H   DUP ( ? )
Working-Storage          ENDS
CODE  SEGMENT
        ASSUME DS: DATA, CS:  CODE, SS: Working-Storage
Start:          MOV   AX , DATA
                MOV   DS , AX
New-number: MOV   AX , First Digit
                MOV   CX , AX
                MUL   CX
                MUL   CX
                MOV   BX , AX
                MOV   AX , Second Digit
                MOV   CX , AX
                MUL   CX
                MUL   CX
                ADD   BX , AX
```

```
        MOV   AX , Third Digit
        MOV   CX , AX
        MUL   CX
        MUL   CX
        ADD   BX , AX
; Binary version of the number, the digit represent in AX
        MOV   AX , First Digit
        MOV   CX , 10D
        MUL   CX
        ADD   AX , Second Digit
        MUL   CX
        ADD   AX , Third Digit
; Number = ((( Third Digit * 10) + Second) * 10 + First Digit)
        CMP     AX , BX
; If they are the same, then display the digits
        JNZ       New-Digit
; Convert each digit to ASCII and display
        MOV     DX , First digit
        ADD     DL , 30H
        MOV   AH , 2H
        INT     21H
        MOV   DX , Second digit
        ADD   DL , 30H
        INT     21H
        MOV   DX , Third digit
```

```
                ADD     DL , 30H
                INT      21H
; Display a new line
                MOV     DL , 0DH
                INT      21H
                MOV     DL , 0AH
                INT      21H
; Next three digit
New-Digit:  INC     Third Digit          JNZ   New-number
            CMP     Third Digit , 10D    MOV   Second digit , 0H
            JNZ      New-number          INC     First Digit
            MOV     Third digit , 0H     CMP     First Digit ,10D
            INC     Second Digit         JZ       Finish
            CMP     Second Digit,10D     JMP     New-number
; Return to DOS
Finish:     MOV     AX , 4C00H
            INT      21H
CODE        ENDS
            END      Start
```

Damascus University

مثال10: حول كتابة برنامج يستخدم تعليمات الدخل والخرج ودارات الربط المبرمجة من جهة، ويأتي كتطبيق مباشر على استخدام البنية IF-THEN-ELSE المتعددة من جهة أخرى. نستخدم الدارة 8255 بالنمط صفر، ونص المثال يتلخص بالتالي:

بفرض أننا سنراقب درجة حرارة مريض عن طريق إحدى تجهيزات المراقبة، بحيث إذا كانت درجة حرارة المريض أقل من $37\ C^o$ يضيء مصباح أصفر وإذا كانت درجة الحرارة أعلى أو تساوي $37\ C^o$ لكن أدنى من $40\ C^o$ يضيء مصباح أخضر، وأن يضيء مصباح أحمر إذا كانت درجة الحرارة تساوي $40\ C^o$ وما فوق.

قبل المباشرة بحل هذا المثال، نذكر بأن هناك ثلاث بنى للشرط نلخصها بالتالي:

1- IF-THEN	2- IF-THEN-ELSE	3- IF-THEN-ELSE
وشكلها العام كالتالي	وشكلها العام كالتالي	وشكلها العام كالتالي

IF Condition THEN
Action
	IF Condition THEN
	Action
	ELSE
	Action
		IF Condition THEN
		Action
		ELSE IF Condition
		THEN
		Action ELSE
		Action

ولحل مثل هذه المسألة نهيء بداية لهذا البرنامج منفذ خرج من الدارة 8255 ولتكن البوابة B بالعنوان FFFBH توصل بالمصابيح الثلاثة لتضيء المصباح المناسب. ولنفرض أن درجة حرارة المريض التي نقرأها من حساس حراري موصول بالجهاز عن طريق محول A/D Converter إلى بوابة الدخل A بالعنوان FFFAH. وبعد ذلك نبدأ بمقارنة درجة الحرارة المقروءة مع القيمة الاسمية الأولى وهي $37\ C^o$ (أو مايعادل 25H). فإذا كانت درجة الحرارة أدنى من $37\ C^o$ سيتم القفز بتعليمة

القفز إذا كان أدنى (JB) إلى اللافته Label التي تشير إلى عملية الاضاءة باللون الاصفر Yellow. وإذا لم يتم القفز، نعلم أن درجة الحرارة هي أعلى من أو تساوي 37 C° وبالتالي يجب أن نقارن القيمة المقروءة مع القيمة 28H (التي تقابل درجة الحرارة 40 C°) لنرى إذا كانت أدنى منها، وبهذه الحالة نقفز إلى اللافته Green. وإذا لم يتم القفز يجب أن تكون درجة الحرارة مساوية لـ 40 C° أو أعلى وبهذه الحالة نذهب مباشرة لتشغيل الضوء الاحمر.

ونشير أيضاً إلى أن مسجل التحكم للدارة 8255 سيكون بهذه الحالة على العنوان FFFDH وكلمة التحكم (CW) ستكون بالقيمة 99H . فإذا فرضنا في هذا البرنامج أن خطوط التحكم بالمصابيح الثلاثة موصولة إلى البوابة B وأن المصباح الاصفر موصول إلى البت 0 والاخضر إلى البت 1 والاحمر إلى البت 2، وأن تشغيل أي مصباح يكون بإخراج 1 إلى البت المناسب للبوابة FFFBH، يصبح البرنامج بشكله المتكامل كالتالي:

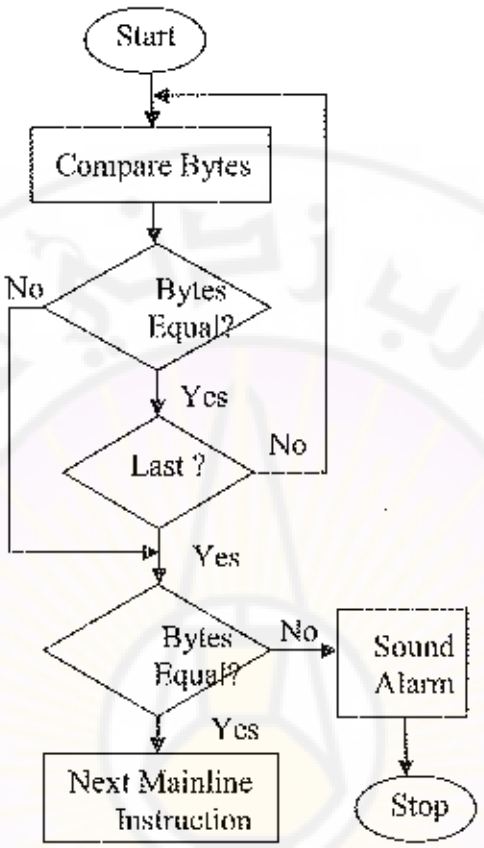
```
; Abstract: this program section reads the temperature of  patient
and
  lights one of three lamps according to the temperature read.
; ports used:   FFFAH as a temperature input
                FFFBH as lamp control output (yellow = bit0,
                green = bit1, red = bit2).
CODE-HERE    SEGMENT
                ASSUME  CS: CODE-HERE
; initialize port FFAAH as an output port
                MOV  DX, 0FFFDH ; point DX to port control
                MOV  AL, 99H ; load CW to set up output port
                OUT   DX, AL  ; send CW to control register
; initialization complete
```

```
                MOV  DX, 0FFFAH ; point DX at input port
                IN  AL, DX  ; read temp from sensor on input port
                CMP  AL, 25H  ; comper temp with 37 C°
                JB  yellow  ; if temp < 37 C° go light yellow lamp
                CMP  AL, 28H  ; compare with 40 degrees
                JB  green  ; if temp < 40 C° go light green lamp
RED:            MOV  AL, 04H  ; temp >= 40 C° load code red
                OUT  DX, AL  ; send code to light red lamp
                JMP  EXIT  ; go to next mainline instruction
Yellow:         MOV  AL, 01H
                OUT  DX, AL
                JMP  EXIT
Green:          MOV  AL, 02H
                OUT  DX, AL
EXIT:           MOV  DX, 0FFFCH  ; next mainline instruction
                IN  AL, DX  ; read sensor
CODE--HERE  ENDS
        END
```

مثال11: اكتب برنامج بلغة التجميع للمعالج 8086 يقارن بين كلمة السر المدخلة من قبل شخص ما يريد استخدام الحاسوب مع كلمة السر الصحيحة المخزنة في الذاكرة، بحيث يصدر إنذار صوتي في حال عدم تطابق الكلمتين.

الجواب: بفرض أن الكلمة المخزنة هي Khaddour بطول 8 Bytes. فان هذا المثال يأتي كتطبيق مباشر على استخدام مقارنة بايتات السلسلة. يبين المخطط الانسيابي أدناه خوارزمية فعل ذلك. تستخدم التركيبة REPEAT–UNTIL للإنهاء عندما لاتتطابق

البايتات المقارنة. وتستخدم البنية IF-THEN إما لاصدار صوت التنبيه إذا كانت السلاسل المقارنة غير متساوية عند أي نقطة أو توجه التنفيذ إلى البرنامج الرئيسي.



وهنا نحتاج إلى منفذ خرج لتشغيل الإنذار On بإخراج 1 إلى دارة التحكم بالإنذار، كما يلزم مؤشر إلى كل سلسلة وإلى عداد يحفظ عدد عناصر السلسلة التي تمت مقارنتها. تحجز العبارة 'Password DB 'Khaddour في بنية المعطيات ثماني بايتات من الذاكرة وتعطي الموقع الذاكري الأول، الاسم Password كما تهيأ المواقع الذاكرية الثمانية بشيفرات ASCII للأحرف: K, h, a, d, d, o, u, r وستحجز العبارة Input-word DB 8 DUP (?) ثمانية مواقع ذاكرية وتعين إلى الموقع الأول الاسم Input--word. تخبر DUP (?) المجمع بعدم تهيئة هذه المواقع الثمانية، حيث يمكننا أن نفرض بأن برنامج آخر يمكن أن يحمّل هذه المواقع بشيفرات ASCII المقروءة من لوحة

المفاتيح. نلاحظ أن أي اسنادات إلى DS, ES ستعني القطعة DATA–Here ونكتب ذلك وفق التالي:

```
DATA–Here        Segment
        Password    DB  'Khaddour'
        Input–word  DB  8 DUP (?); Space for user input
DATA–Here        ENDS
STACK–Here       Segment Stack
        DW  100 DUP (0); set aside 100 words for Stack top
STACK–Here       ENDS
CODE–Here        Segment
ASSUME CS: CODE–Here, SS; STACK–Here,
DS: DATA–Here, ES: DATA–Here
```

وبما أننا نهيىء DS و ES بنفس القيم، فإن كل من SI و DI سيشيران إلى مواقع في نفس القطعة DATA–Here. وتقوم التعليمات التالية بتهيئة منفذ بعنوان محدد كبوابة خرج.

```
; Initialize Data & Extra Segment
MOV  AX, DATA–Here
MOV  DS, AX
MOV  ES, AX
MOV  DX, FFFAH ;Set up port–as an output port for Alarm
MOV  AL, 99H
OUT  DX, AL
```

نفرض بهذا البرنامج أن الإنذار موصول إلى البيت الأقل أهمية لبوابة بطول 8 bit وبالعنوان FFFAH. يمكن بتعليمة من الشكل MOV AL, 01 تحميل الواحد في البت LSB للمسجل AL. أما تهيئة البوابة ذات العنوان FFFAH التي ستستخدم كخرج يتم

بالتعليمة MOV DX, 0FFFAH ملحقة بتعليمة من الشكل OUT DX, AL. وبعد الانتهاء من التهيئات الأساسية، يكمل البرنامج وفق التالي:

```
        LEA  SI, Password    ; load source pointer
        LEA  DI,  Input-word ; load destination pointer
        MOV  CX, 08H         ; Counter = passord length
        CLD                  ; Increment DI & SI
REPE    CMPSB                ; Compare the two strings
        JNE   Alarm          ; Not equal, sound alarm
        JMP   Ok             ; Equal – continue
Alarm:  MOV  AL, 01          ; to sound alarm, send
        MOV  DX, 0FFFAH      ; a 1 to the output
        OUT  DX, AL          ; port whose address is
        HLT                  ; in DX and HALT
Ok:     NOP                  ; rest of program for user whose
                             ; password = Khaddour
CODE-Here       ENDS
                END
```

تحمّل العبارة LEA SI, Password العنوان الفعال أو الإزاحة لبداية السلسلة Khaddour في المسجل SI. وبما أن Password هي بند المعطيات الأول في القطعة DATA-Here فإن SI سيحمل بالقيمة 0000H. وبأسلوب مماثل، تحمّل العبارة LEA DI, Input-word إزاحة بداية السلسلة في DI، أي بالقيمة 0008H وتهيىء العبارة MOV CX, 08H عدد البايتات في السلسلة بالمسجل CX. وتقوم CLD بإخبار المعالج بزيادة SI و DI آليا بعد مقارنة بايتين في السلسلة. ستقارن التعليمة CMPSB البايت المشار اليه بالمسجل SI مع البايت المشار إليه بالمسجل DI، وترفع المؤشرات وفقا للنتيجة وستزيد المؤشرين SI و DI ليشيرا إلى العناصر التالية في السلسلة وتنقص العداد

CX ليشير إلى انه تمت مقارنة عنصرين. تخبر البادئة REPE في مقدمة التعليمة CMPSB المعالج 8086 بإعادة هذه التعليمة إذا كانت البايتات المقارنة متساوية ولم يكن CX قد وصل إلى الصفر. إذا لم يرفع مؤشر الصفر عندما يغادر التنفيذ حلقة الإعادة، نعرف عندها أن السلسلتين غير متساويتين، وهذا يعني أن كلمة السر المدخلة كانت غير صحيحة، ولذلك يجب إطلاق صوت الإنذار، حيث إن التعليمة JNE Alarm هي التي تفحص مؤشر الصفر. أما إذا كانت الكلمة صحيحة ينتقل التنفيذ لـ JMP Ok .

مثال12: لحساب العاملي Factorial وليكن على سبيل المثال !3،

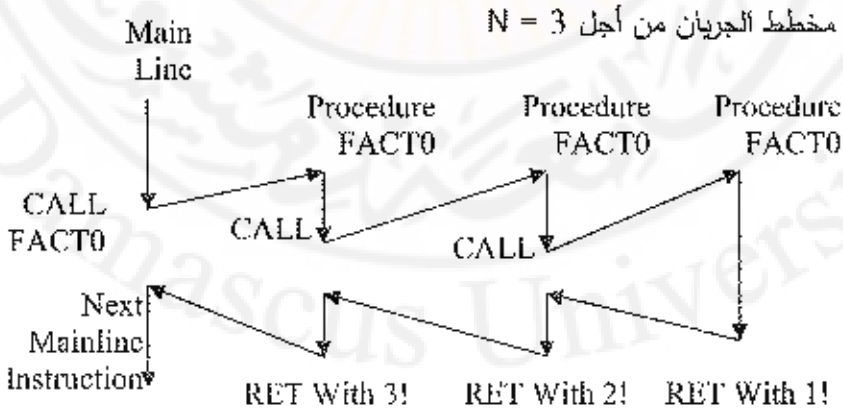
من المعلوم أن N! = N.(N−1).(N−2).....4.3.2.1 وبالتالي يأتي هذا المثال كتطبيق مباشر للإجراءات المتكررة Recursiv Procedures أي الإجراء الذي يستدعي نفسه. يمكننا أن نعبّر بشكل بسيط عن الإجراء المتكرر الذي يحسب العاملي بالخوارزمية التالية:

If N = 1 Then Factorial = 1, ELSE

Factotrial = N x (Factorial of N – 1)x x2x1

عندما يستدعى الإجراء لاول مرة فإنه يملك القيمة 3 من أجل N لكنه لايملك قيمة العاملي للعدد N−1 التي يحتاجها لانجاز الضرب المشار اليه بالخوارزمية، حيث يحل الإجراء هذه المشكلة باستدعاء نفسه وذلك لحساب العاملي الذي يحتاجه للعدد N−1 ويستدعي نفسه مرارا وتكرارا وصولا إلى عاملي الواحد. يبين الشكل أدناه عدة طرق لإمكانية تمثيل هذه العملية بحيث يبين الشكل الأول: جريان البرنامج الذي سيحدث إذا استدعينا الإجراء

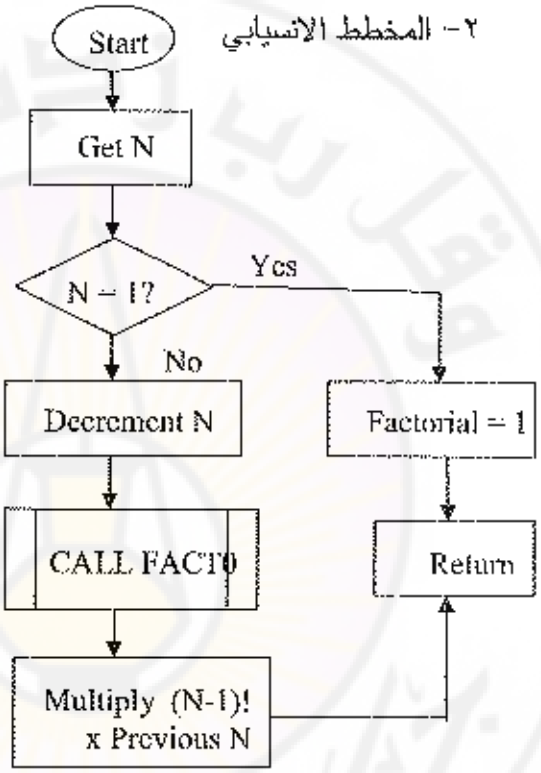
١- مخطط الجريان من أجل N = 3



بالقيمة N = 3 (الإجراء يستدعي نفسه لحساب !(N−1) أو 2! ويستدعي نفسه بعدئذ مرة ثانية لحساب القيمة التالية لعاملي !(N−1) أو 1!).

ويبين الشكل الثاني: المخطط الانسيابي لنفس الخوارزمية. ويبين الشكل الثالث: تمثيل الخوارزمية بشيفرة زائفة موسعة.

٢- المخطط الانسيابي



٣- الشيفرة الزائفة

```
Procedure FACT0
If N =1
    Factorial = 1
    Ret
ELSE
     Repeat
         Decrement N
         CALL FACT0
    Until N = 1
    Multiply (N−1) x Previous N
   RET
```

ويمكننا المباشرة بكتابة الكود المقابل بالتصريح عن قطعة المكدس وتجهيز المكدس ليتسع على سبيل المثال إلى 100 كلمة مع لافتة عند قمة المكدس.

```
; Procedure Used: FACT0
STACK-Here        SEGMENT STACK
            DW  200 DUP (0)   ; set aside 200 words for stack
STACK-Top         LABEL  Word   ; Assign name to word above
```

```
                                    ; stack top
STACK-Here          ENDS
Number              EQU   03
CODE-Here           SEGMENT
        ASSUME CS:CODE-Here, SS:STACK-Here
          MOV  AX, STACK-Here  ; initialize stack segment reg.
          MOV  SS, AX
          MOV  SP, OFFSET STACK-Top ; initialize stack Pointer
          SUB  SP, 0004H  ; make space in stack for factorial to
                                    ; be returned
           MOV  AX, Number ; put number to be passed on stack
          PUSH  AX
          CALL  FACT0  ; compute factorial of number
          POP  AX          ; get result
          NOP         ; simlate next mainline instructions
          NOP
          NOP
```

تهيىء التعليمات الثلاثة الأولى في قطعة الشفرة لبرنامج الخط الرئيسي، مسجل قطعة المكدس SS ومسجل مؤشر المكدس SP. وبعد هذا ستنقص التعليمة SUB SP, 04 مسجل مؤشر المكدس بمقدار 4، بمعنى أننا جعلنا مؤشر المكدس يقفز (إلى الأسفل) فوق كلمتين في المكدس. وسيستخدم موقعا هاتين الكلمتين لتمرير العاملي المحسوب من الإجراء العائد إلى برنامج الخط الرئيسي. نحمّل بعد ذلك العدد الذي نريد حساب العاملي له، في AX وندفع هذه القيمة على المكدس حيث يراد ولوج الإجراء إليها لاحقا. والآن جاهزون لاستدعاء الإجراء. وقد أعطي الإجراء الاسم FACT0 بالتوجيهين:

FACT0 PROC NEAR و FACT0 ENDP، وهذا الإجراء قريب لأنه موجود في نفس قطعة الشيفرة مع التعليمة التي تستدعيه.

```
; Procedure: FACT0
FACT0   PROC   Near
        PUSHF   ; save flags and registers on stack
        PUSH   AX
        PUSH   DX
        PUSH   BP
        MOV BP, SP   ; point BP at top of stack
        MOV AX, [BP + 10]   ; copy number from stack to AX
        CMP AX, 0001H   ; if number not equal 1 then go on
        JNE   go-on   ; and compute factorial
        MOV   WORD  PTR [BP + 12], 0001H  ; else load 1!
        MOV   WORD  PTR [BP + 14], 0000H  ; stack and
                                          ; return to mainline
        JMP   EXIT
go-on   SUB   SP, 0004H   ; make space in stack for factorial
        DEC   AX   ; decrement number now in AX
        PUSH   AX   ; save number – 1 on stack
        CALL   FACT0   ; compute factorial of number – 1
        MOV   BP, SP   ; point BP at top of stack
        MOV   AX, [BP + 2]   ; last (N-1)! From stack to AX
        MUL   WORD  PTR [BP + 16]   ; multiply by previous N
        MOV  [BP + 18], AX   ;copy new factorial to stack
        MOV  [BP + 20], DX   ;copy new factorial to stack
```

```
        ADD   SP, 0006H   ; point stack pointer at pushed reg.
EXIT    POP   BP   ; restore registers
        POP   DX
        POP   AX
        POPF
        RET
FACT0   ENDP
Code-Here   ENDS        END
```

يلاحظ من البرنامج أعلاه أن قيمة N مطمورة بعد عشرة عناوين أعلى المكدس اعتبارا من المكان الذي ترك فيه مؤشر المكدس بعد أن دفع BP على المكدس. وللولوج إلى هذه القيمة المطمورة، ننسخ أولا SP إلى BP مشيرا إلى قمة المكدس. ونستطيع بعدئذ أن نستخدم التعبير [BP + 10] ليشير إلى العنوان حيث دفعت N.

ستنسخ التعليمة MOV AX, [BP + 10]، القيمة N من المكدس إلى AX فإذا كانت قيمة N المقروءة 1 سيكون العاملي عندئذ 1.

وهكذا نريد أن نضع القيمة 00000001H في المواقع المكدسية التي تم حجزها من أجل النتيجة. تستخدم التعليمة MOV WORD PTR [BP + 12], 0001H لتحميل قيمة ما إلى موقع مطمور في المكدس، حيث يخبر التوجيه WORD PTR، المجمع أننا نريد نقل كل كلمة إلى الموقع الذاكري المحدد، وبدون هذه التوجيهات لن يعرف المجمع ما إذا كانت شيفرة التعليمة مخصصة لنقل بايت أم لنقل كلمة.

ونستخدم التعليمة MOV WORD PTR [BP + 14], 0000H بشكل مشابه لنقل قيمة كلمة إلى موقع كلمة أخر محجوز في المكدس من أجل العاملي. عند تمرير العدد 3 إلى FACT0 تقرر التعليمة CMP AX, 0001H والتعليمات JNE go-on أن N ليست 1، وتنتقل التنفيذ إلى التعليمة SUB SP, 04H. ووفقا للخوارزمية فإننا نذهب لإيجاد القيمة N!، بضرب N مع القيمة N-1 وسوف نستدعي FACT0 مرة ثانية لإيجاد القيمة (N-1)!.

المراجع

[1] أحمد خضور، الدارات المنطقية، منشورات جامعة دمشق، ٢٠٠٨

[2] معجم مصطلحات المعلوماتية، من منشورات الجمعية العلمية السورية للمعلوماتية، الطبعة الأولى، ٢٠٠٠.

[3] زهير دحروج، صلاح الدوه جي، بسام كردي، عمار جوخدار، محسن حسين، لغات البرمجة (١)، منشورات جامعة دمشق، ١٩٩٩

[4] المعالجات الصغرية، البرمجة والربط البيني مع المكونات الصلبة، شعاع للنشر والعلوم، ترجمة المهندس شوقي مريشة، ١٩٩٨

[5] نديم شاهين، الالكترونيات الطبية والإشارات الحيوية، جامعة دمشق، ١٩٩٢

[6] عدة مواقع من الانترنيت يطول عرضها، ولكن يحدد معظمها بالبحث تحت اسم المفردة أو الفقرة نفسها الواردة في الكتاب (مثال ذلك المرجع [7] و[8])

[7] الموسوعة العربية www.arab-ency.com/index.php القياسات الحيوية

[8] *Pentium* - Wikipedia, the free encyclopedia

[9] Sundararjan V. Madihally, Principles of Biomedical Engineering, Artech House, 2010

[10] Vincent P Heuring; Harry F Jordan, Computer systems design And architecture, NJ Pearson Prentice Hall c 2004

[11] V Rajaraman; T Radhakrishnan, An introduction to digital Computer design, New Delhi Prentice Hall 1997

[12] James K. Roberge; Kent H. Lundberg, Operational Amplifiers: Theory and Practice, Massachusetts Institute of Technology, 2007

[13] Jonathan VALVANO, Embedded microcomputer systems: real time interfacing, Thomson, 2007

[14] A.P. Godse; D.A. Godse, Advanced Microprocessors, Second Revised Edition, Technical Publications Pune, 2007

[15] M. Rafiquzzaman, Fundamentals of Digital Logic and Microcomputer Design, 5th Edition, 2005

[16] A.P. Godse; D.A. Godse, Microprocessors and

Microcontrollers, Technical Publications Pune, 2009
[17] Jerry C. Whitaker, The Electronics Handbook, Published In 2005 by CRC Press
[18] John E Uffenbeck, Microcomputers and microprocessor: the 8080, 8085, and Z-80 programming, interfacing, and troubleshooting, NJ Prentice Hall 2000
[19] Sunil Mathur, Microprocessor 8085 and its interfacing, Second Edition, New Delhi PHI Learning 2011
[20] A.P. Godse; D.A. Godse, Microprocessors and its Application Technical Publications Pune, 2006
[21] K. Udaya Kumar; B.S. Umashankar, The 8085 Microprocessor: Architecture, Programming and Interfacing, Published by Darling Kinddersley (India), 2008
[22] A.P. Godse; D.A. Godse, Microprocessors Techniques: 8085 and 8086 Architecture, Instruction Set, Programming& Interfacing, 4st Edition, Technical Publications Pune, 2008
[23] Mathur Sunil, Microprocessor 8086: Architecture, Programming and Interfacing, by PHI Learning Private Limiter, New Delhi, 2011
[24] Nilesh B. Bahadure, Microprocessors: The 8086/8088, 80186/80286, 80386/80486 and the Pentium Family, by PHI Learning Private Limiter, New Delhi, 2010
[25] Sivarama P. Dandamudi, F.B. Schneider, Introduction to Assembly Language Programming: From 8086 to Pentium Processor, Springer-Verlag New York, LLC, 2000
[26] Sivarama P. Dandamudi, Fundamentals of computer organization and design, New York: Springer,cop.,2003
[27] Steven F Barrett; Daniel J Pack, Atmel AVR microcontroller Primer: Programming and interfacing, Morgan&Claypool Publishers, 2008
[28] Milan Verle, PIC Microcontrollers, mikroElektronika; 1st edition, 2008
[29] Thomas L. Floyd, Digital Fundamentals, 10th Edition , Pearson Education, 2009.

ملحق ١: مصطلحات علمية

مصطلحات من الهندسة الطبية Biomedical Engineering	
Accuracy	الدقة أو الضبط
Acquisition	التحصيل (الاقتباس)
Amplitude	مطال أو مدى الإشارة
Bio–Signals	الإشارات الحيوية
Bioacoustical Signals	الإشارات الصوتية الحيوية
Biochemical Signals	الإشارات الكيميائية الحيوية
Bioelectric Signals	الإشارات الكهربائية الحيوية
Biomagnetic Signals	الإشارات المغناطيسية الحيوية
Biomedical Signals	الإشارات الطبية الحيوية
Biooptical Signals	الإشارات البصرية الحيوية
Bio – Measurements	القياسات الحيوية
Contact lens electrodes	إلكترودات كعدسات لاصقة
Conversion time	زمن التحويل
Converters	المحولات
ECG (ElectroCardioGram)	تخطيط القلب الكهربائي
EMG (ElectroMyoGram)	تخطيط العضلات الكهربائي
EOG (Electro –Oculogram) ERG (Electro– Retinogram)	تخطيط العين
Electromagnetic	الكهرومغناطيسية
Flow Sensors	حساسات التدفق
Impedance	الممانعة

Light Sensors	الحساسات الضوئية
Band– pass filter	مرشح تمرير الحزمة
High– pass filter	مرشح التمرير المرتفع
Low–pass filter	مرشح التمرير المنخفض
MCG (MagnetoCardiogGaph)	تخطيط القلب المغناطيسي
MEG (MagnetoEncephaloGraphy)	تخطيط الدماغ المغناطيسي
Needle electrodes	الإلكترودات الإبرية
Non–Invasive	الطرق غير الجراحية
Operational Amplifier	مضخم العمليات
Oximetry	تركيز الأوكسجين في الدم
pH electrode	الكترود الحموضة
Pneumotachometer	مقياس التنفس
Resolution	دقة التحويل
Scalp electrodes	الإلكترودات على فروة الرأس
Sensor	الحساس أو المتحسس
Skin electrodes	الإلكترودات على سطح الجلد
Stimulators	وحدات إثارة
Temperature Sensors	الحساسات الحرارية
Thermocouplers	المزدوجات الحرارية
Transducers	المبدلات
Ultrasonic	الامواج فوق الصوتية
Unipolar	احادي القطبية

مصطلحات حول الحواسيب الصغرية (أو المايكرو كومبيوتر) Microcomputers والمعالجات والمتحكمات الصغرية Microprocessor & Microcontroller والبرمجة بلغة الأسمبلي (التجميع) Assembly Language Programming	
Accumulator	المراكم
AC – Auxiliary Carry	الحمل المساعد
Active Filters	المرشحات الفعالة
Address Bus - AB	ممر العناوين
Addressing Mode	نمط العنونة (الولوج أو العمل)
Algorithm	الخوارزمية أو منهجية الحل
Amplification	التضخيم
Analog to Digital Converter	محول التمثيلي إلى رقمي
Analysis	التحليل
Arbitration Bus	ممر التحكيم
ALU– Arithmetic Logic Unit	وحدة الحساب والمنطق
Assembler	المجمّع
Asynchronous Serial Data Commun.	النقل التسلسلي غير المتزامن
Band Width	عرض حزمة
Batch processing	المعالجة الدفعية
Baud Rate	معدل البود
Bidirectional bus	ممرات ثنائية الاتجاه
Binary–Weighted--Resistors	المقاومات الموزونة ثنائياً
Bipolar technologies	تقنيات ثنائية القطبية
Bit Slice Processor	المعالجات الشرائحية

Break point	نقطة القطع
Buffer	الحاجز
Bypass	تجاوز
Cache Memory	الذاكرة الفورية (ذاكرة الكاش)
CPU – Central Processing Unit	وحدة المعالجة المركزية
Chip Select	انتقاء القطعة (الدارة)
CISC – Complex Instruction Set Computer	الحواسيب ذات جملة التعليمات المعقدة
Clock Generator	مولد الساعة
Closed–loop–gain (Avcl)	ربح الحلقة المغلقة
Comparator with Hysteresis	مقارن بتخلف
Compiler	المترجم
CB – Control Bus	ممر التحكم
Core Duo	معالجات ثنائية الانوية
Critical frequency	التردد الحدي
Cross Assembler	المجمع المتعارض
Current Output	خرج تياري
CWR – Control Word Register	مسجل كلمة التحكم
Data Bus – DB	ممر المعطيات
DDR – Data Direction Register	مسجل توجيه المعطيات
Debuggers	المنقحات
Destination	الهدف أو الوجهة
Digital Computer	الحاسوب الرقمي

Digital Ground	تأريض رقمي
Digital to Analog Converter	محول الرقمي إلى تمثيلي
Digitization	التبديل الرقمي
DMA- Direct Memory Access	نظام الولوج المباشر للذاكرة
Direction Flag	مؤشر أو علم الاتجاه
Display System	وحدة أو منظومة العرض
Distributed Processing	المعالجة الموزعة أو المتعددة
Duplicate Directive	توجيه المضاعفة
Effective Addresses	العنوان الفعّال
EISA (Extended ISA)	ISA موسع (ممرات قياسية للنقل)
EEPROM Electrically--Erasable Programmatically ROM	ROM قابلة للبرمجة والمحي كهربائيا
Embedded System	النظام المدمج
Enabled	ممكن (متاح)
Even Parity	الندية أو الازدواجية الزوجية
Execute	عملية تنفيذ التعليمة
Fetch	عملية جلب التعليمة
Filtering	الترشيح
Firmware	البرامج البنوية الداخلية
Flag Register (FR)	مسجل المؤشرات
Floating Point	الفاصلة العائمة
Full Duplex	نقل ثنائي الاتجاه
Full-Scale Output Voltage	جهد خرج كامل المقياس أو التدريج

Functional Block Diagram	التركيبة الداخلية لهذا المعالج
Gain–Bandwidth Product	ضرب الربح بعرض الحزمة
General Purpose Registers	مسجلات للأغراض العامة
Half Duplex	نصف ثنائي الاتجاه
Hardware	العتاد المادي
High–Level Programming Language	لغات البرمجة عالية المستوى
ISA (Industry Standard Arch.)	المعمارية الصناعية القياسية
Input/output Ports	بوابات الدخل/الخرج
Input Peripherals	طرفيات أو محيطيات الدخل
Instruction Cycle	دورة التعليمة
Instruction Decoder	مفكك ترميز التعليمة
Instruction Set	جملة التعليمات (للمعالج)
Interfacing Microprocessor	معالج ربط
Interrupt	المقاطعة
Inverting Input	المدخل العاكس أو القالب
Latch	الماسك
Local Bus Systems	الناقل أو الممر المحلي
Location	الموقع (موضع في الذاكرة)
Machine Code	لغة الآلة
Machine Cycle	دورة أو حلقة الآلة
Machine–dependent	الارتباط أو الاعتماد على الآلة
Mainframe computers	حواسيب كبيرة
Mask Programming	برمجة القناع

Math co–processor	المعالج الحسابي
Memory Matrix	مصفوفة الذاكرة
MR – Memory Read	القراءة من الذاكرة
Memory Segmentation	تقطيع الذاكرة
Memory Word	كلمة الذاكرة
Micro architecture	المعمارية المصغرة
Minicomputers	حواسيب متوسطة
Mnemonics	مختصرات
Monolithic Microprocessor	معالج وحيد البلورة
Most – Significant bit	البت الأكثر أهمية (وزنا)
Multiplexed Address/data Bus	ممر العناوين/المعطيات المركب
Multi processor System	متعددة المعالجات
Multiprogramming	البرمجة المتعددة
Negative--Feedback	التغذية العكسية السالبة
Nesting Procedures	تعشيش الإجراءات أو تداخلها
Nibble	نصف البايت (4 bit)
Noninverting Input	المدخل غير العاكس
Nonmaskable Interrupt	مقاطعة غير قابلة للتقنيع
Nonvolatile Memories	الذواكر الدائمة
NC – Not Connected	غير موصول
OOP Object–Oriented Programming	البرمجة غرضية التوجه
Odd Parity	مؤشر الندية أو التكافؤ الفردية
Offset	الإزاحة

Open collector	المجمع المفتوح
Open–Loop	نموذج الحلقة المفتوحة
Operational Amplifier (OP AMP)	مضخم العمليات العادي
Operating Systems	أنظمة التشغيل
Optical Encoder	مشفر ضوئي
Oscillator	المذبذب
Overflow Flag	مؤشر الطفح
Pair Registers	زوج مسجلات
Personal Computer	الحاسوب الشخصي
Phase	الطور أو الفاز
Photodiode	الديود الضوئي
Physical or Absolute Address	العنوان الفيزيائي أو العنوان المطلق
Pin Configuration	تركيبة المآخذ (توزع الماخذ)
Pipelining	عملية جلب التعليمة وتنفيذها معا
PIC (Priority Interrupt Controller)	متحكم المقاطعة بالأولوية
Processing	المعالجة
Program Development Tools	أدوات تطوير البرنامج
Programmable Peripheral Interface	حد أو ربط المحيطيات المبرمجة
Programmable Timer/ Counter	المؤقت / العداد المبرمج
Protect Mode	النمط المحمي
Prototype	النموذج الأولي
Pseudo operation	العملية الزائفة
Punched Cards	البطاقات المثقبة

RAM – Random Memory Access	الذاكرة ذات الولوج العشوائي
Real Mode	النمط الحقيقي
Reference Voltage	الجهد المرجعي
Refresh	إنعاش
Register Instruction	مسجل التعليمة
Removing ambiguity	إزالة الغموض
RISC – Reduced Instruction Computer	الحواسيب ذات التعليمات المخفضة
ROM – Read Only Memory	ذاكرة قابلة للقراءة فقط
RWM – Read Write Memories	ذاكرة للقراءة والكتابة
Saturation	حد الإشباع
Screen	شاشة
Segment Register	مسجلات القطع
Semiconductor Technology	تقنيات أنصاف النواقل
Simple Handshaking I/O	دخل وخرج بإشارة مصافحة
Simple Strobe I/O	دخل وخرج بسيط بإشارة تمكين
Simplex	النقل البسيط
Software Compatibility	التوافق البرمجي
Source	المصدر أو المنبع
Special Function Registers	المسجلات ذات الوظائف الخاصة
SP – Stack Pointer	مؤشر المكدس أو الخزان
Statement terminator	نهاية العبارة أو التعليمة
Status Register	مسجل الحالة

Straight Binary	المشفر ثنائيا أو الثنائي المباشر
Support Circuits	دارات الدعم
Synchronous Serial Data Commun.	النقل التسلسلي المتزامن للبيانات
System Resources	موارد النظام
Templates	نماذج التشفير (لتعليمات المعالج)
Three-state	ثلاثي الحالة
Time-Share and Multitasking System	الأنظمة المتشاركة بالزمن متعددة مهام
Timing and Control	دارات التحكم والتزامن
Unipolar	أحادية القطبية
Vertual Memory	الذاكرة الافتراضية
Virtual Ground	الأرضي الظاهري
VLSI (Very Large Scale Integration)	درجة تكامل عالية جدا
Volatile Memories	الذواكر غير الدائمة
Von Neumann Architecture	معمارية فون نويمان (للحاسوب)
Waveforms	أشكال الموجات الزمنية (للمعالج)

Damascus University

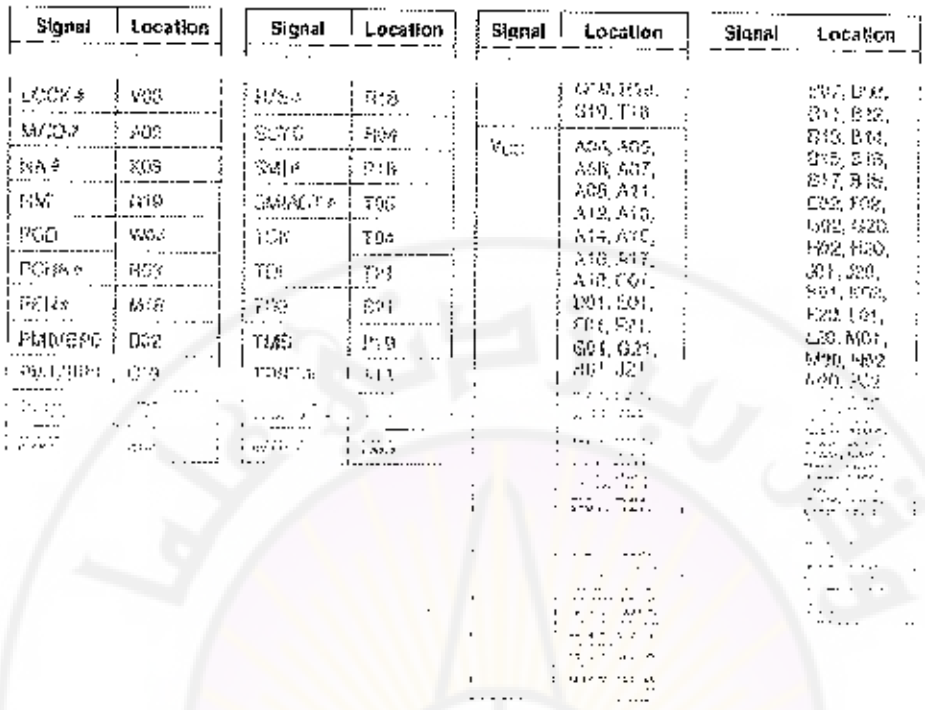
ملحق ٢: إشارات المعالج Pentium 510/60, 567/66

Signal	Location
A3	T17
A4	W19
A5	U18
A6	U17
A7	T16
A8	U16
A9	T15
A10	U15
A11	T14
A12	U14
A13	T13
A14	U13
A15	T12
A16	U12
A17	T11
A18	U11
A19	T10
A20	U10
A21	U21
A22	U09
A23	U20
A24	U08
A25	U19
A26	T09
A27	V21
A28	V06
A29	V20
A30	W05
A31	V19
A20M#	U05
ADS#	P04
AHOLD	L02
AP	P03
APCHK#	W03
BE0#	U04
BE1#	Q04

Signal	Location
BE2#	U06
BE3#	V01
BE4#	T06
BE5#	S04
BE6#	U07
BE7#	W01
BOFF#	K04
BP2	B02
BP3	B03
BRDY#	L04
BREQ	V02
BT0	T08
BT1	W21
BT2	T07
BT3	W20
BUSCHK#	T03
CACHE#	J04
CLK	K18
D0	D03
D1	E03
D2	E04
D3	F03
D4	C04
D5	G00
D6	B04
D7	G04
D8	F04
D9	C12
D10	C13
D11	E05
D12	C14
D13	D04
D14	C13
D15	D05
D16	D06
D17	B09

Signal	Location
D18	C06
D19	C15
D20	D07
D21	C16
D22	C07
D23	A10
D24	D10
D25	C08
D26	C11
D27	D09
D28	D11
D29	C09
D30	D12
D31	C10
D32	D10
D33	C17
D34	C19
D35	D17
D36	C18
D37	D16
D38	D19
D39	D15
D40	D14
D41	B19
D42	D20
D43	A20
D44	D21
D45	A21
D46	E18
D47	B20
D48	B21
D49	F19
D50	C20
D51	F18
D52	C21
D53	G18

Signal	Location
D54	E20
D55	G19
D56	H21
D57	F20
D58	J18
D59	H19
D60	L19
D61	K19
D62	J19
D63	H18
D/C#	V04
DP0	H04
DP1	C05
DP2	A9
DP3	D08
DP4	D18
DP5	A19
DP6	E19
DP7	F21
EADS#	M03
EWBE#	A03
FERR#	H03
FLUSH#	U02
FRCMC#	M19
HIT#	W02
HITM#	M04
HLDA	Q03
HOLD	V05
IBT	T19
IERR#	C02
IGNNE#	S20
INIT	T20
INTR	N18
INV	A01
IU	J02
IV	B01



Signal	Location
LOCK#	V03
M/IO#	A05
NA#	X03
INV	D19
PGD	W04
PCI#s	R03
PCI#s	M18
PM0/GPIO	D02
PM1/GPIO	C19

Signal	Location
RESET	R19
SCYC	P04
SMI#	P18
SMIACT#	T05
TCK	T04
TDI	T01
TDO	U01
TMS	P19
TRST#	L13

Signal	Location
	C17, D04, G19, T18
V_{CC}	A04, A05, A06, A07, A08, A11, A12, A13, A14, A15, A16, A17, A18, C01, J01, L01, T04, T21, G01, G21, J01, J21

Signal	Location
	B07, B11, B12, B13, B14, B15, B16, B17, B18, C02, F02, G02, G20, H02, H20, J01, J20, K01, K02, K20, L01, L21, M01, M20, N02, N20, P02

ملحق٣: قائمة بالأشكال الواردة في الكتاب

رقم الشكل	اسم الشكل	رقم الصفحة
1-1	إشارة تخطيط القلب حسب طريقة غولدبرغ ECG	23
1-2	إشارة تخطيط القلب ECG حسب طريقة أينتهوفن	24
1-3	طريقة ويلسون في تحصيل إشارة الـ ECG	24
1-4	أ- المخ منبع لإشارة الدماغ. ب- نظام الالتقاط العالمي 10-20 ج- إشارة EEG وتقسيمها إلى أربعة أنواع من الأمواج	26
1-5	نظام تحصيل إشارة الـ EOG وإشارة ERG	27
2-1	المخطط الصندوقي للمايكرو كومبيوتر	33
2-2	تركيبة الحاسوب الرقمي وفق معمارية فون نويمان	34
2-3	البنى الأساسية المتعارف عليها للحاسوب الصغري	34
2-4	المخطط الصندوقي للحاسوب الصغري (المايكروي)	35
2-5	الرمز الصندوقي للذاكرة مع أهم الإشارات القياسية	41
2-6	الأنواع الأساسية للذواكر	43
2-7	المخطط الصندوقي لذاكرة 16 x 16 RAM	44
2-8	المخطط الصندوقي لذاكرة من نوع ROM	45
2-9	زيادة طول كلمة الذاكرة	47
2-10	زيادة حجم الذاكرة	48
2-11	توزع مغارز الدارة المتكاملة 2147	49
2-12	كرت ذاكرة بسعة 8 KB باستخدام الدارة 2147	50
2-13	توزع إشارات بعض ذواكر Static RAM و EPROM (2716- 6116 – 2764 – 6264 – 4364 – 43256A)	53

2-14	كرت ذاكرة EPROM & RAM بسعة إجمالية 16 KB	56
2-15	مخطط ذاكرة يظهر آلية إلغاء الفراغ في المجال العنوني	57
3-1	توزع مآخذ الدارة المتكاملة 741 (مضخم عمليات)	72
3-2	توصيلات مختلفة لمضخم العمليات a) Common Amplifier b) Non Inverting Amplifier, c) Inverting Amplifier	72
3-3	توصيلات مضخم العمليات كمقارن ومفاضل ومكامل	75
3-4	دارة مقارن بتخلف	77
3-5	توصيلات مضخم العمليات كجامع ومرشح	78
3-6	محول بسيط من رقمي إلى تشابهي بعرض 4 bit	84
3-7	المحول MC 1408 8 bit D/A مع محول من تيار إلى جهد	86
3-8	محول A/D مكون من مقارنات متوازية	89
1-1	التركيبة الأساسية لمعالجات صغرية بكلمة 8 bits	95
1-2	الإشارات الأساسية لمعالجات صغرية بكلمة 8 bits	97
4-1	المخطط الصندوقي للبنية الداخلية للمعالج الصغري I8080	100
4-2	الشكل العام لمسجل المؤشرات لدى المعالج 8080	101
4-3	توزع الإشارات على مآخذ المعالج 8080	103
4-4	قسم من المخطط الزمني لفعالية المعالج الصغري 8080	107
5-1	مخطط توزع مآخذ بعض الدارات المتكاملة (7400 – 7404 – 7408 – 7432 – 74139 – 74LS244 – 74LS245 – 74F299 – 74LS373)	114
5-2	الشكل العام لتوصيل الدارة 8224 مع المعالج الصغري 8080	115
5-3	المخطط الصندوقي وتوزع الإشارات للدارة 8224	116

5-4	المخطط الصندوقي وتوزع الإشارات للدارة 8228/8238	118
5-5	توزع إشارات الدارة 8205	119
5-6	توزع إشارات الدارة 8212	119
5-7	توزع إشارات الدارة 8216/8226	120
5-8	توزع إشارات الدارة 8282/8283	121
5-9	توزع إشارات الدارة 8286	122
5-10	توزع إشارات الدارة 8214	123
5-11	توصيل المعالج 8080A لتشكيل الممرات AB, DB, CB	125
5-12	توصيل 4KB ذاكرة EPROM و 4 KB ذاكرة RAM وبوابتي دخل وبوابة خرج مع المعالج 8080A لتشكيل حاسوب صغري	126
5-13	الجزء الثاني من التصميم المطلوب بالمثال 2-5	131
6-1	دخل خرج بنبضة تمكين بسيطة	136
6-2	دخل/خرج بمصافحة وحيدة	136
6-3	التركيبة الداخلية للدارة القابلة للبرمجة 8255	138
6-4	توزع إشارات الدارة 8255	139
6-5	مفهوم التراكيب المختلفة لإشارات التحكم بالدارة 8255	140
6-6	كلمة الحالة الأولى للدارة 8255	141
6-7	توصيل بوابات الدارة 8255 بال Mode0 كخرج	142
6-8	كلمة الحالة وتوصيل بوابات الدارة 8255 بال Mode1 كدخل	143
6-9	كلمة الحالة وتوصيل بوابات الدارة 8255 بال Mode1 كخرج	144
6-10	الشكل العام لكلمة التحكم بالدارة 8255 بال Mode2	145
6-11	توصيل الدارة 8255 بال Mode2 والبوابة B بال Mode0	146
6-12	كلمة الحالة الثانية للدارة 8255	146

6-13	النقل التسلسلي المتزامن وغير المتزامن للمعطيات	148
6-14	ربط طرفية بحاسوب باستخدام الـ Modem	149
6-15	المخطط الصندوقي للدارة 8251A وتوزع الإشارات على الماخذ	151
6-16	صيغ كلمات التحكم للدارة 8251A	155
6-17	المخطط الصندوقي وتوزع الإشارات للدارة 8254	159
6-18	صيغة كلمة التحكم للدارة 8254	160
6-19	توصيل دارة المعالج 8080 بوجود متحكم للمقاطعة 8214	163
6-20	تصميم مايكرو كومبيوتر باستخدام المعالج الصغري 8080A وبوجود دارات دعم مبرمجة	164
7-1	التركيبة الداخلية للمعالج 8085	165
7-2	توزيع الإشارات للمعالج 8085A	168
7-3	المخطط الصندوقي وتوزع الإشارات للدارة 8185	171
7-4	المخطط الصندوقي للبنية العامة للدارة 8155	172
7-5	توزع إشارات 8155	173
7-6	المخطط الصندوقي وتوزع إشارات الدارة 8555	178
7-7	المخطط الصندوقي وتوزع إشارات الدارة 8755	181
7-8	التركيبة الصغرى لحاسب مايكروي	180
7-9	نظام مايكرو كومبيوتر متكامل بثلاث دارات متكاملة	183
7-10	توصيل المعالج 8085A لتشكيل الممرات AB, DB, CB	185
7-11	نظام مايكرو كومبيوتر يوافق المثال 4-7	188
7-12	آلية إلغاء الفراغ في المجال العنوني للمعالج	189
8-1	المخطط الصندوقي لمنظومة متعددة المعالجات	198

8-2	المخطط الصندوقي لمتحكم الممر 8288 وتوزع إشاراته	200
8-3	التركيبة الداخلية للمعالج الصغري 8086	202
8-4	مسجل الأعلام لدى المعالج 8086	203
8-5	الشكل العام لإشارة الساعة لدى المعالج 8086	207
8-6	توزع إشارات المعالج 8086	208
8-7	مخطط بنوك الذاكرة لدى المعالج 8086	211
8-8	a- المخطط الزمني لحلقة القراءة للمعالج 8086 بالنموذج الاصغري b- المخطط الزمني لحلقة الكتابة للمعالج 8086 بالنموذج الاصغري	220
8-9	a- مخطط القراءة من الذاكرة للمعالج 8086 بالنموذج الأعظمي b- مخطط الكتابة إلى الذاكرة للمعالج 8086 بالنموذج الأعظمي	221
9-1	المخطط الصندوقي لطريقة Polled I/O	226
9-2	آلية تحقيق ذاكرة كاش في حاسوب صغري	233
9-3	مخطط صندوقي يبين كيفية عمل المتحكم DMA في منظومة حاسوب صغري	236
10-1	توزع إشارات الدارة 8284	239
10-2	الشكل القياسي لتوصيل الكريستال للمعالج 8086 عبر 8224	240
10-3	توصيل إشارة RESET للمعالج 8086 عبر الدارة 8224	241
10-4	المخطط الصندوقي للتركيبة الداخلية للدارة 8259A	242
10-5	توزع إشارات الدارة 8259A	243
10-6	المخطط الصندوقي للمتحكم 8237	244
10-7	توزع إشارات المتحكم 8237 على مأخذه	245
10-8	نظام مايكرو كومبيوتر باستخدام المعالج 8088 ويوجود متحكم DMA	246

10-9	التركيبة الداخلية للمعالج المساعد 8087	247
10-10	توزع إشارات المعالج المساعد 8087	249
10-11	المخطط الصندوقي للمحكم 8289	250
10-12	توزع إشارات محكم الممر 8259	251
10-13	المخطط الصندوقي للمعالج الصغري 8086 بالنموذج الأعظمي بوجود متحكم ومحكم الممر 8288 و 8289	252
11-1	النماذج الفيزيائية لممر المعطيات	253
11-2	a- نقل بايت (زوجي) على 16 bit لممر المعطيات b- نقل بايت (فردي) على 16 bit لممر المعطيات c- نقل كلمة (زوجية) على 16 bit لممر المعطيات d- نقل كلمة (فردية) على 16 bit لممر المعطيات	255
11-3	الترميز المطلق للذواكر	256
11-4	تقنية الترميز الخطي للذواكر	258
11-5	تقنية الترميز الكلي للذواكر	260
11-6	إشارات المعالج 8086 المستخدمة بالنموذج الاصغري	261
11-7	الشكل العام لتوليد ممرات AB و الـ DB للمعالج 8086	262
11-8	توصيل المعالج 8086 بالنمط الاصغري	263
11-9	توصيل 64KB من ذاكرة EPROM و 64 KB من ذاكرة RAM مع المعالج 8086 بالنمط الاصغري	265
11-10	ربط المعالج المساعد 8087 مع المعالج 8086	267
11-11	توصيل المعالج 8086 بالنمط الأعظمي	269
11-12	مخطط الترميز لوصل ذواكر وبوابات مع المعالج 8086	271
11-13	توصيل الذواكر والبوابات بالمعالج 8086	272

12-1	التركيبة العامة للمعالج 80186/80188	276
12-2	توزع مآخذ المعالج 80186 بكل من التغليفان LCC و PGA	277
12-3	الوجه العلوي والسفلي لمآخذ المعالج 80186 بالتغليف PGA	278
12-4	توزع إشارات المعالج 80186 على مآخذه بالتغليف LCC	278
12-5	المخطط الصندوقي الداخلي للمعالج 80286	282
12-6	توزع إشارات المعالج 80286	283
13-1	المصفوفة الشبكية للمعالج 386 DX التي تبين مآخذه	291
13-2	توزع إشارات المعالج 386 SX على مآخذه	292
13-3	توزع مآخذ المعالج 80486	300
14-1	المخطط الصندوقي لمعالجات Pentium بالتوترات 75/90/100/120/133/150/166/200 MHz	312
14-2	المسجلات العامة في معالج البنتيوم	313
14-3	مسجل المؤشرات في معالجات بنتيوم	313
14-4	مسجلات المقاطع في معالجات بنتيوم	314
14-5	مجموعات الإشارات الأساسية للمعالج Pentium	315
14-6	الوجه العلوي لمآخذ المعالج Pentium 510/60, 567/66	316
14-7	الوجه السفلي لمآخذ المعالج Pentium 510/60, 567/66	317
14-8	المظهر العام لمعالج Core i7 من إنتل	322
15-1	القالب المستخدم من إنتل لتشفير تعليمات المعالج 8086	336
IV-1	التركيبة العامة للمنظومة المدمجة	426
19-1	التركيبة العامة للمتحكم الصغري	431
19-2	التركيبة الداخلية للمتحكم Intel 8051	434
19-3	توزع مآخذ المتحكم Intel 8051 (8031)	440

19-4	الأساليب الممكنة لتأمين الساعة اللازمة للمتحكم 8051	440
19-5	دارة تصفير المتحكم 8051	441
19-6	تنظيم ذاكرة البرنامج لدى المتحكم 8051	445
19-7	تنظيم ذاكرة المعطيات Data Memory Organization	445
19-8	الأجزاء الأساسية لذاكرة المعطيات الداخلية (128 Bytes) للمتحكم الصغري 8051	446
19-9	الشكل العام لمعمارية متحكمات AVR	449
19-10	توزع الإشارات للمتحكم الصغري At mega 16	455
19-11	صميم معمارية AVR	456
19-12	توزع الإشارات على مآخذ المتحكم PIC16F84A	464
19-13	مسجل الحالة في المتحكم PIC16F84A	467
19-14	المخطط الصندوقي للمتحكم الصغري PIC16F877A	472
19-15	توزع إشارات المتحكم PIC16F877A/874A	475
19-16	الأنواع المختلفة للذواكر المستخدمة بالمتحكمات الصغرية	478
19-17	حاسبة بسيطة باستخدام المتحكم الصغري PIC16F877A	482
20-1	طريقة تطوير برنامج بسيط بلغة الأسمبلي	488

ملحق ٤: قائمة بالجداول الواردة في الكتاب

رقم الجدول	اسم الجدول	رقم الصفحة
1-1	أهم الإشارات الحيوية المقاسة مع مجالات قياساتها وتردداته	29
1-2	أهم الطرق المستخدمة في أجهزة قياس الإشارات الحيوية	32
2-1	أرقام دارات متكاملة تمثل ذواكر مختلفة (RAM & EPROM)	51
2-2	مخطط عنونة ذاكرة بسعة 16 KB ربعها قابل للقراءة	55
2-3	مخطط ذاكرة لـ 64 KB بمعدل 2 KB EPROM و 62 KB RAM	58
4-1	أزواج المسجلات الممكنة لدى المعالج 8080A	102
5-1	بعض الدارات المتكاملة من رتبة SSI و MSI بالتقانة TTL	112
5-2	مخطط توزيع ذواكر EPROM و RAM مع المعالج 8080A	128
5-3	مخطط توزيع البوابات مع المعالج 8080A	130
5-4	عنونة 64 KB ذاكرة (بمعدل 4KB EPROM و 60KB RAM)	133
6-1	إشارة الدارة 8251A المبرمجة للنقل التسلسلي	153
7-1	المقاطعات لدى المعالج 8085A (العنوان والأفضلية)	167
7-2	ترميز إشارات الحالة لدى المعالج 8085A	169
7-3	ترميز كل من إشارتي WR' , RD' وإشارتي الحالة S0, S1 للمعالج 8085A مع IO/M'	170
7-4	عناوين المكونات الأساسية لدارة الدعم 8155 للمعالج 8085A	174

7-5	برمجة بوابات الدارة 8155	176
7-6	برمجة بوابات الدارة 8355	180
8-1	مفهوم إشارات متحكم الممر 8288	201
8-2	ترميز الاشارتين A0 و BHE' لتحديد نوع المعطيات	210
8-3	ترميز الاشارتين QS0 و QS1 لتحديد حالة الرتل	213
8-4	ترميز S0', S1', S2' لتحديد كلمة الحالة للمعالج 8086	214
8-5	ترميز الاشارتين S3, S4 لتحديد نوع المقطع	214
8-6	ترتيب أولويات المقاطعة لدى المعالج 8086	216
8-7	مسجلات الدليل والقطع في حساب العنوان الفيزيائي	218
10-1	مفهوم إشارات محكم الممر 8289	251
11-1	ترميز A_0 و BHE' لتحديد طول البيانات التي يتعامل معها المعالج 8086 مع الذاكرة	253
11-2	خارطة عنونة ذاكرة EPROM بسعة 16 KB مع المعالج 8086	257
11-3	ترميز 16KB ذاكرة EPROM و 16 KB ذاكرة RAM مع المعالج 8086	259
11-4	ترميز 64KB ذاكرة EPROM و 64 KB ذاكرة RAM مع المعالج 8086	266
11-5	ترميز 64KB ذاكرة EPROM و 128 KB ذاكرة RAM	273
14-1	أسماء المؤشرات الإضافية في معالجات بنتيوم عن مؤشرات المعالج 8086	314
15-1	التوافقيات الممكنة لمعاملات تعليمة MOV من تعليمات المعالج	331

	8086	
15-2	الترميز (الكود) المقابل لكل من مسجلات المعالج 8086	337
17-1	تعليمات قفز المعالج 8086 المتكافئة	381
17-2	مفهوم بعض الوظائف المتاحة تحت المقاطعة 21H	386
19-1	الفوارق الجوهرية بين متحكمات Intel,AVR,PIC من رتبة 8bit	433
19-2	المواصفات الأساسية لبعض المتحكمات الصغرية من إنتل	439
19-3	المواصفات الفنية الأساسية لمجمل سلاسل متحكمات AVR	450
19-4	المواصفات الأساسية لمتحكمات العائلة MegaAVR Family	451
19-5	الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة B	455
19-6	الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة C	455
19-7	الميزات الخاصة للمتحكم ATmega16 على مآخذ البوابة D	456
19-8	إشارات المتحكم PIC16F877A/874A	476
20-1	مقارنة أولية بين متحكمات من Intel, AVR, PIC	485
20-2	برنامج للمتحكم PIC16F877 بلغة عليا	494

اللجنة العلمية:

الأستاذ الدكتور حسن أبو النور

الأستاذ المساعد الدكتور مازن محايري

الأستاذ المساعد الدكتور صلاح الدوجي

المدقق اللغوي:

الأستاذ المساعد الدكتور علي كردي

حقوق الطبع والترجمة والنشر محفوظة لمديرية الكتب والمطبوعات

وقل رب زدني علما
جامعة دمشق
Damascus University


وقل رب زدني علما
جامعة دمشق
Damascus University