١ معمارية حواسيب x86

حواسيب عائلة x86 تتبع لمعمارية العالم جون فون نيومان (John von Neumann architecture) والتي تنص على أن أي تصميم لجهاز حاسب يجب أن يتكون من الثلاث وحدات التالية :

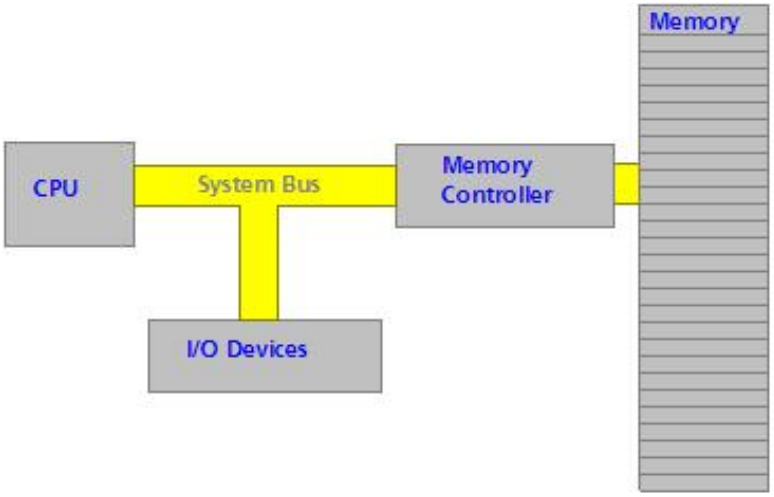
١. معالج أو وحدة معالجة مركزية (Central Processing Unit).

٢. ذاكرة (Memory).

٣. أجهزة إدخال وإخراج (I/O Devices).

الوحدة الاولى هي وحدة المعالجة والتي تقوم بتنفيذ الأوامر والعمليات الحسابية ، أما **الوحدة الثانية** فهي تحوي البيانات والتعليمات والأوامر التي يجب على وحدة المعالجة أن تنفذها ، **وأخيراً** وحدات الإدخال والإخراج وهي الاجهزة التي تستخدم في ادخال البيانات واخراجها.(انظر الشكل ١.١ حيث يوضح مثالاً لهذه المعمارية) ويربط بين كل هذه الأجزاء هو مسار النظام (System Bus) وفيما يلي سنستعرض وظيفة كل جزء على حدة.

شكل ١.١: معمارية حواسيب x86

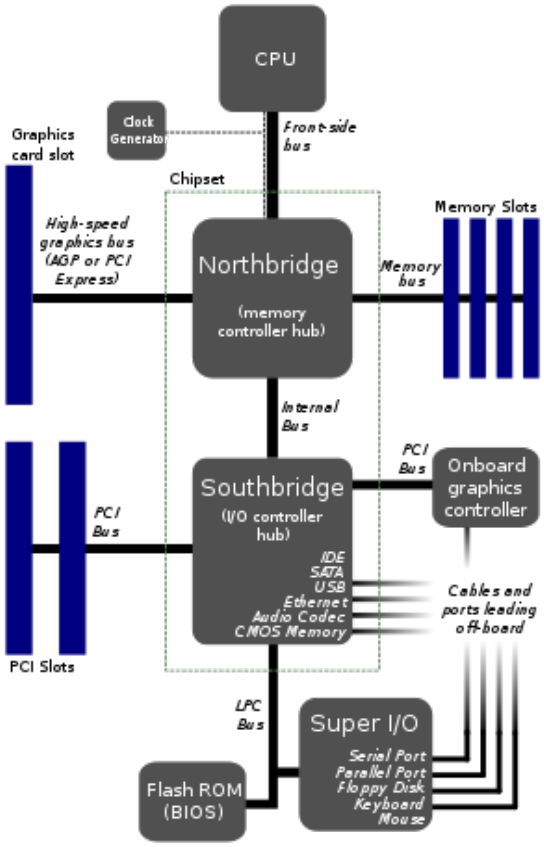


١.١ معمارية النظام

١.١.١ مسار النظام System Bus

يربط مسار النظام[١] (System Bus) وحدة المعالجة المركزية (CPU) مع متحكم الذاكرة الرئيسية . وظيفة هذه المسارات هي نقل البيانات بين أجزاء الحاسب المختلفة. والشكل ٢.١ يوضح الصورة العامة للمسارات في أجهزة الحواسيب الشخصية (Personal Computers). ويتألف مسار النظام من ثلاث مسارات وهي مسار البيانات (Data Bus) ومسار العناوين (Address Bus) ومسار التحكم (Control Bus).

شكل ٢.١: المسارات في الحواسيب الشخصية x86



[١] ويسمى أيضا Front-side Bus.

مسار البيانات Data Bus

مسار البيانات هو عبارة عن خطوط (Lines) كل خط يمثل بت واحد. وغالبا ما يكون هناك 32 خط (أي أن مسار البيانات بطول 32-bit) ويستخدم هذا المسار في نقل البيانات (Data) من المعالج (وتحديداً من وحدة التحكم Control Unit) الى متحكم الذاكرة (الى الجسر الشمالي NorthBridge تحديدا نظراً لان متحكم الذاكرة يطبق على عليه). وبسبب أن حجم مسار البيانات هو حجم ثابت فان هذا يتطلب معالجة خاصة عند ارسال بيانات بطول أقل من طول مسار البيانات ، فغالبا ما يقوم المعالج باضافة أصفار في الخطوط الغير مستخدمة (Padding). أما في حالة إرسال بيانات بطول أكبر فان عملية نقلها تتم على عدة مراحل وفي كل مرحلة ترسل 32-bit من البيانات .

مسار العناوين Address Bus

يستخدم مسار العناوين في نقل عنوان الذاكرة المراد استخدامه سواءاً للقراءة منه أو الكتابة عليه ، ويحدد حجم مسار العناوين أكبر عنوان يمكن الوصل اليه في الذاكرة وبالتالي يحدد لنا حجم الذاكرة التي يستطيع الحاسب التعامل م□ا . وفي الأجهزة التي تستخدم معالجات انتل 8086 كان حجم هذا المسار هو 20-bit وبالتالي فان أقصى ذاكرة يتعامل معها هذا المعالج هي 1 MB[٢] أما في معالجات 80286/80386 فان حجم هذا مسار هو 24-bit وفي المعالجات التي تليها تم زيادة هذا الحجم الى 32-bit وبالتالي يمكن تنصيب ذاكرة بحجم 4 GB ، وفي المعالجات الحديثة تم زيادة هذا الحجم ، ولكننا سنقتصر في هذا البحث على المعالجات التي تدعم مسار عناوين بطول 32-bit بسبب انتشارها وسيطرتها لمدة من الزمن على أجهزة الحواسيب الشخصية.

مسار التحكم Control Bus

يستخدم مسار التحكم في ارسال الأوامر مثل أمر القراءة من العنوان الموجود على مسار العناوين أو أمر الكتابة على العنوان المطلوب . ويتألف هذا المسار من عدد من الخطوط وكل خط (بت) يؤدي وظيفة محددة. أحد هذه الخطوط هو خط الكتابة WRITE والذي يعني أن العنوان الموجود على خط العناوين يجب أن تُعيَّن له القيمة الموجودة في مسار البيانات . الخط الآخر هو خط القراءة READ والذي يدل على أن العنوان الموجود في مسار العناوين يجب أن تُقرأ قيمته الى مسار البيانات . آخر خط يهمنا هو خط الولوج ACCESS والذي يحدد ما اذا كان العنوان موجهٌ الى متحكم الذاكرة أم الى متحكم الإدخال والإخراج وفي حالة كانت قيمة هذا الخط هي القيمة 1 فان هذا يعني أن العنوان موجهُ الى متحكم أجهزة الإدخال والإخراج وبالتالي سيتم القراءة من هذا العنوان أو الكتابة اليه وذلك بحسب قيمة الخطين READ and WRITE.

[٢] ناتجة من حساب 2 مرفوع للقوة 20.

٢.١.١ متحكم الذاكرة

قبل أن نذكر وظيفة هذا المتحكم يجب إعطاء نبذة عن ماهية المتحكمات (Controllers) في جهاز الحاسب.
ويُعرَّف **المتحكم** بأنه شريحة تتحكم بعتاد ما تحوي العديد من المسجلات الداخلية وظيفتها هو استقبال الأوامر وتنفيذها على العتاد. ويمكن أن نعرفها بأنها شريحة للربط ما بين الأوامر البرمجية الى أوامر تنفذ على عتاد ما. وأي متحكم يحوي العديد من المسجلات سواءاً كانت لإرسال واستقبال البيانات أو للأوامر ، وأي مسجل يجب أن يأخذ رقم فريد يميزه عن بقية المسجلات الموجودة في هذا المتحكم أو في أي متحكم آخر وذلك حتى نتمكن من التعامل معه برمجياً ، هذا الرقم يعرف باسم المنفذ (Port) وسنطلع عليه لاحقاً.
وعمل المتحكم يبدأ عندما يُرسل أمر اليه حيث يبدأ المتحكم في تنفيذ هذا الأمر ومن ثم يضع النتيجة في أحد مسجلاته ويرسل إشارة (Interrupt) الى المعالج لكي يقوم بقرائة القيمة.
نعود الى متحكم الذاكرة الرئيسية والذي يتواجد غالبا على متحكم الجسر الشمالي (NorthBridge) إنظر الشكل ٣.١ .حيث تكمن وظيفته الأساسية في استقبال الأوامر المرسلة الى الذاكرة وتنفيذها ، ويقوم هذا المتحكم يتوجيه العناوين المرسلة الى أي من شرائح الذاكرة كذلك يقوم بإعادة تنعيش (Refresh) هذه الذاكرة طيلة عمل الحاسب حتى لا تفقد الذاكرة محتوياتها.

شكل ٣.١: الجسر الشمالي
يعتبر هذا الجسر حلقة الوصل ما بين المعالج والذاكرة الرئيسية والبايوس وذاكرة الفيديو ومتحكم الإدخال والإخراج حيث يستقبل الأوامر ويقوم بتوجيهها الى المتحكم المطلوب.

٣.١.١ متحكم الإدخال والإخراج

يستخدم متحكم الإدخال والإخراج (ويسمى أيضا الجسر الجنوبي SouthBridge) في ربط متحكمات أجهزة الإدخال والإخراج مع المعالج وهذا يتضح من الشكل ٢.١. حيث يظهر أن الجسر الشمالي يرتبط مباشرة مع المعالج بينما الجسر الجنوبي يرتبط مع الجسر الشمالي والذي بدوره يربط متحكمات عتاد الإدخال والإخراج في الحاسب. وكل جهاز يرتبط بالحاسب (مثل لوحة المفاتيح أو الفأرة أو الطابعة ...الخ) لديه متحكم بداخل الجهاز ومتحكم آخر بداخل الحاسب ، حيث يرسل المتحكم الموجود بداخل الحاسب الأوامر الى المتحكم الموجود بداخل العتاد . ولبرمجة أي جهاز فانه يجب برمجة المتحكم الموجود في الحاسب وهذا يتم عن طريق معرفة المسجلات (Registers) الموجودة به ووظيفة كل مسجل فيه حتى نتمكن من إرسال الأوامر

الصحيحة اليه. هذه المسجلات تأخذ أرقاما معينة تسمى منافذ برمجية (Software Ports) بحيث تميز هذه الأرقام المسجلات من بعضها البعض[٣].

المنافذ Ports

يستخدم مفهوم المنافذ في علوم الحاسب للدلالة على عدة أشياء فمثلا في مجال برمجة الشبكات تكون برامج الخادم لها رقم منفذ معين حتى تسمح لبرامج العميل بالاتصال معها، كذلك توجد المنافذ الموجودة في اللوحة الأم لوصل عتاد الحاسب بها ، أيضا أي مسجل في متحكم على الجهاز لديه رقم منفذ وهذا ما نقصده في حديثنا عن المنافذ في هذا البحث. و يمكن الوصول لمنافذ المتحكمات والتي تعرف ب I/O ports باستخدام تعليمة المعالج `in port_address` والتعليمة `out port_address` حيث تستخدم الأولى لقراءة قيمة من مسجل في متحكم ووضعها في أحد مسجلات المعالج أما التعليمة الثانية تستخدم لكتابة قيمة في مسجل للمعالج الى مسجل في المتحكم . وعند استخدام أحد هذين الأمرين فان ذلك يعني أن العنوان موجه الى متحكم الإدخال والإخراج وليس الى متحكم الذاكرة حيث يقوم المعالج بتعين قيمة الخط `ACCESS` الموجود في مسار التحكم (Control Bus) وبالتالي يستجيب متحكم الإدخال والإخراج ويقرأ هذا العنوان ويقوم بتوجيهه الى المتحكم المطلوب . وهناك بعض الأجهزة تستخدم عناوين الذاكرة للوصول للمتحكم الخاص بها وهو ما يعرف ب Memory Mapped I/O حيث عند كتابة أي بيانات على هذه العناوين فان ذلك يعني كتابة هذه البيانات على متحكمات للأجهزة وليس على الذاكرة الرئيسية. فمثلاً عند الكتابة على عنوان الذاكرة `0x0:0xa000` فان هذا يؤدي الى الكتابة على شاشة الحاسب نظراً لان هذا العنوان هو موجه (Memory Mapped) مع متحكم شاشة الحاسب والجدول ١.١ يوضح خريطة الذاكرة في حواسيب x86، ولا تحتاج الكتابة لمثل هذه العناوين استخدام الأوامر `in/out` بعكس الكتابة في عناوين المنافذ `port I/O` .

عناوين منافذ الإدخال والإخراج (Port I/O) هي عناوين تستخدمها المسجلات الموجودة على المتحكمات ويقوم البايوس بمهمة ترقيم هذه المسجلات ، والجدول ٢.١ يعرض قائمة بعناوين المنافذ ووظيفة كل منهم.

٢.١ المعالج

يعتبر المعالج هو المحرك الرئيسي لجهاز الحاسب حيث يستقبل الأوامر ويقوم بتفيذها .

[٣]هناك بعض المسجلات لبعض المتحكمات تأخذ نفس الرقم ، لكن طبيعة الأمر المُرسل (قراءة أو كتابة) هو الذي يحدد المسجل الذي يجب التعامل معه.

جدول ١.١: مخطط الذاكرة لحواسيب x86

عنوان البداية	عنوان النهاية	الوصف
0x00000	0x003ff	جدول المقاطعات IVT
0x00400	0x004ff	منطقة بيانات البايوس
0x00500	0x07bff	غير مستخدمة
0x07c00	0x07dff	برنامج محمل النظام
0x07e00	0x9ffff	غير مستخدمة
0xa0000	0xaffff	ذاكرة الفيديو Video RAM
0xb0000	0xb7777	ذاكرة الفيديو أحادية اللون Monochrome VRAM
0xb8000	0xbffff	ذاكرة الفيديو الملونة Color VRAM
0xc0000	0xc7fff	ذاكرة Video ROM BIOS
0xc8000	0xeffff	منطقة BIOS Shadow Area
0xf0000	0xfffff	نظام البايوس

١.٢.١ دورة تنفيذ التعليمات

لكي يُنفذ المعالج البرامج الموجودة على الذاكرة فان هذا يتطلب بعضا من الخطوات التي يجب أن يقوم بها ، وفي كل دقة للساعة (Clock tick) يقوم المعالج بالبدء بخطوة من هذه الخطوات ، وفيما يلي سردا لها.

أولاً مرحلة جلب البيانات (Fetch) وفيها يتم جلب البيانات من الذاكرة الرئيسية الى المسجلات بداخل المعالج.

ثانياً مرحلة تفسير البيانات (Decode).

ثالثاً مرحلة تنفيذ البيانات (Execute).

رابعاً مرحلة حفظ النتائج (Write back).

٢.٢.١ أنماط عمل المعالج CPU Modes

عندما طرحت شركة أنتل أول اصدارة من معالجات 16-bit لم يكن هناك ما يعرف بأنماط المعالج حيث كان المعالج يعمل بنمط واحد وهو ما يعرف الان بالنمط الحقيقي (Real Mode) ، في هذا النمط يقوم المعالج بتنفيذ أي أمر موجه اليه ولا يوجد ما يُعرف بصلاحيات التنفيذ حيث يمكن لبرنامج للمستخدم أي يقوم بتنفيذ أمر يتسبب في ايقاف النظام عن العمل (مثل الأمر hlt) ، كذلك توجد عددٌ من المشاكل في هذا النمط فمثلا لا توجد حماية للذاكرة من برمجيات المستخدم ولا يوجد أي دعم لمفهوم تعدد المهام (Multitasking). لذلك سارعت أنتل بادخال عدة أنماط على بنية المعالج لتحل هذه المشاكل ، بحيث يُمكن للمعالج أي يعمل في أي نمط وأن يقوم بالتحويل وقتما شاء. ويُعرَّف **نمط المعالج** بأنه طريقة معينة يتبعها

جدول ٢.١: منافذ الإدخال والإخراج لحواسيب x86

رقم المنفذ	الإستخدام
0000-000f	Slave DMA controller
0010-001F	System
0020-0021	First Interrupt controller (8259 chip)
0030-0031	Second interrupt controller
0040-0043	Programable Interval Timer 1 (8254 chip)
0048-004B	Programable Interval Timer 2
0050-006F	System devices
0070-0071	NMI Enable / Real Time Clock
0080-008B	DMA Page registers
0090-009F	System devices
00A0-00A1	Slave interrupt controller
00C0-00DE	Master DMA controller
00F0-00FF	System devices
0100-0167	System devices
0168-016F	IDE Interface - Quaternary channel
0170-0177	IDE interface - Secondary channel
01E8-01EF	IDE Interface - Tertiary channel
01F0-01F7	IDE interface - Primary channel
0200-0207	Games Port (joystick port)
0220-022F	Usually used by sound cards, also used by NOVEL NETWARE KEY CARD
0270-0273	Plug and Play hardware
0278-027A	Parallel Port *
0280-028F	Sometimes used for LCD Display I/O
02B0-02DF	Alternate VGA Video Display Adaptor assignment (secondary address)
02E0-02E7	GPIB 0, data aquisition card 0 (02E1 to 02E3 only)
02E8-02EF	Serial Port - COM 4
02F8-02FF	Serial Port - COM 2
0300-031F	Often used as a default for Network Interface cards (was prototype card)
0320-023F	ST506 and ESDI Hard Disk Drive Interface (mostly used in PX/XT and early PC/AT)
0330-0331	MPU-401 (midi) interface, on Sound Cards
0360-036F	Sometimes used for Network Interface cards
0376-0377	Another address used by the Secondary IDE Controller (see 0170-0177)
0378-037A	Parallel Port *
0388-038B	FM (sound) synthesis port on sound cards
03B0-03BB	MDA, EGA and VGA Video Display Adaptor (only 03B0 to 03BB used)
03BC-03BF	Parallel Port (originally only fitted to IBM mono display adaptors) *
03C0-03DF	EGA / VGA Video Display Adaptor, (Primary address)
03E0-03E7	PCIC PCMCIA Port Controller
03E8-03EF	Serial Port - COM 3
03F0-03F6	Floppy Disk Drive Interface
03F7-03f7	Another address used by the Primary IDE Controller (see 01F0-01F7)
03F8-03FF	Serial Port - COM 1
0533-0537	Windows sound system (used by many sound cards)

المعالج أثناء عمله لتنفيذ الأوامر فمثلا يحدد النمط المستخدم ما إذا كان هناك حماية لعنواين الذاكرة بحيث لا يمكن لبرنامج لا يمتلك صلاحيات معينة الوصول لأي منطقة في الذاكرة.

٣.٢.١ النمط الحقيقي Real Mode

هذا النمط هو الذي يبدأ الجهاز الحاسب بالعمل عندما يقلع وهذا بسبب أن حواسيب x86 تم تصميمها بحيث تدعم الأجهزة القديمة وحتى تحافظ انتل على ذلك فان هذا ما جعلها تدع المعالج يبدأ بالنمط الحقيقي عند الإقلاع توافقاً مع الحواسيب القديمة ، وبعد ذلك عندما يستلم نظام التشغيل زمام التحكم بالحاسب فانه مخيرٌ ما بين الإستمرار بالعمل في هذا النمط وبالتالي يسمى هذا النظام **نظام تشغيل 16-bit** وبين تحويل نمط المعالج الى النمط الاخر وهو النمط المحمي (Protected Mode) وبالتالي يسمى النظام **نظام تشغيل 32-bit**. في هذا النمط يستخدم المعالج مسجلات من طول 16-bit (مثلاً المسجلات ax,bx,cx,dx,...etc) ويستخدم عنونة **المقطع:الإزاحة** (Segment:Offset) للوصول الى الذاكرة الرئيسية – سيتم شرحها في الفقرة التالية– وأيضا يدعم ذاكرة بحجم 1 ميجابايت ولا يقدم أي دعم لحماية الذاكرة والذاكرة التخيلية (Virtual Memory) ولا يوفر حماية للذاكرة من برمجيات المستخدم.

عنونة المقطع:الإزاحة (Segment:Offset Addressing)

بعد طرح أنتل لمعالج 8086 وهو أول معالج ١٦ بت ، ظهرت مشكلة حجم الذاكرة حيث أن طول المسجلات المستخدمة في هذا المعالج (مسجلات البيانات والعناوين) هو ١٦ بت وهذا ما سمح للمسجل بأن يتعامل مع ٦٤ كيلوبايت فقط من الذاكرة على الرغم من أن مسار العناوين (Address Bus) في هذه الأجهزة كان بحجم ٢٠ بت وهو ما يسمح باستخدام ذاكرة بحجم ١ ميجا. الى هنا كان الخيار أمام شركة أنتل هو بزيادة حجم المسجلات الموجودة بداخل المعالج ولكن هذا الحل كان مكلفاً جدا آنذاك نظراً لإن هذه المسجلات هي ذواكر من النوع SRAM وهو نوع مكلفاً على الرغم من إمكانياته العالية. ما فعلته انتل هو إيجاد طريقة مختلفة لعنونة الذاكرة فبدلاً من استخدام مسجل واحد للوصل الى عناوين الذاكرة تم استخدام مسجلين كل منهما بطول ١٦ بت ، الفكرة كانت في تقسيم الذاكرة الى مقاطع (Segments) ويُستخدم أحد المسجلات للدلالة على رقم أو عنوان المقطع (Segment Number or Address) وبالتالي هناك ٦٥٥٣٦ مقطع مختلف[٤] ويُستخدم المسجل الآخر للوصل الى العناوين بداخل المقطع وهي ما تعرف بالقيم (Offsets) بداخل المقطع وبالتالي كل مقطع يحوي ٦٥٥٣٦ بايت (أي أن حجم المقطع هو ٦٤ كيلوبايت). إذاً يُعرَّف **المقطع Segments** بأنها منطقة من الذاكرة بحجم ٦٤ كيلوبايت ويمكن الوصول الى أي مقطع وذلك بتحميل رقم المقطع أو عنوان المقطع الى أي من مسجلات المقاطع الموجودة بداخل المعالج (مثل المسجلات CS,SS,DS,ES) – سيتم شرحها لاحقا – ، ويمكن الوصول الى محتويات المقطع **الإزاحة Offset** وذلك

[٤] هذا ناتج من حساب 2^{16}.

بتحميل العنوان المطلوب الوصل اليه الى أي من مسجلات القيم (تبدأ العناوين في أي مقطع من العنوان 0x0 الى 0xffff). هذه الطريقة التي اقترحتها انتل للوصول الى عناوين الذاكرة خلقت لنا مفهوم العنوان المنطقي (Logical Address) حيث لكي نصل الى أي مكان في الذاكرة فانه يجب تحديد عنوان المقطع والعنوان بداخل هذا المقطع وذلك على الشكل Segment:Offset حيث الجزء الأول يحدد عنوان المقطع والجزء الثاني يحدد العنوان بداخل المقطع. مهمة المعالج حاليا هي تحويل العنوان المنطقي الى عنوان فيزيائي أو حقيقي لكي يقوم بارساله عبر مسار العناوين الى متحكم الذاكرة ، و طريقة التحويل تعتمد على أن الإزاحة (Offset) يتم جمعها الى عنوان المقطع (Segment) [٥] ولكن بعد أن يتم ضربها في العدد ١٦ وذلك بسبب أن أي مقطع يبدأ بعد ١٦ بايت من المقطع السابق له . والتحويل يتم كالأتي :

$$physical\_address = segment * 0x10 + offset$$

فمثلا العنوان المنطقي 0x07c0:0x0000 يتم تحويله وذلك بضرب العنوان 0x07c0 بالعدد ١٦ (أو العدد 0x10 بالنظام السادس عشر) ليصبح هكذا 0x07c00، وبعد ذلك يتم جمعه الى ال Offset ليخرج العنوان الفيزيائي 0x07c00.

مشكلة تداخل المقاطع

ذكرنا في الفقرة السابقة أن أي مقطع يبدأ مباشرة بعد ١٦ بايت من المقطع السابق له ، وهذا يعني أن المقاطع متداخلة حيث يمكن الوصول لعنوان فيزيائي معين بأكثر من طريقة مختلفة. مثلاً في مثالنا السابق استخدمنا العنوان المنطقي 0x07c0:0x0000 للوصول الى المنطقة الذاكرية 0x07c00 ، ويمكن أن نستبدل العنوان المنطقي السابق بالعنوان 0x0000:0x7c00 وبعد اجراء التحويل سنحصل على نفس العنوان الفيزيائي 0x07c00. وفي الحقيقة هناك ٤٠٩٦ طريقة مختلفة للوصل لعنوان في الذاكرة[٦] والشكل ٤.١ يوضح لنا تداخل هذه المقاطع. هذا **التداخل Overlapping** سمح لأي برنامج ما إمكانية الوصول الى بيانات برنامج آخر والكتابة عليها وهذا ما جعل النمط الحقيقي ضعيف من ناحية حماية أجزاء الذاكرة.

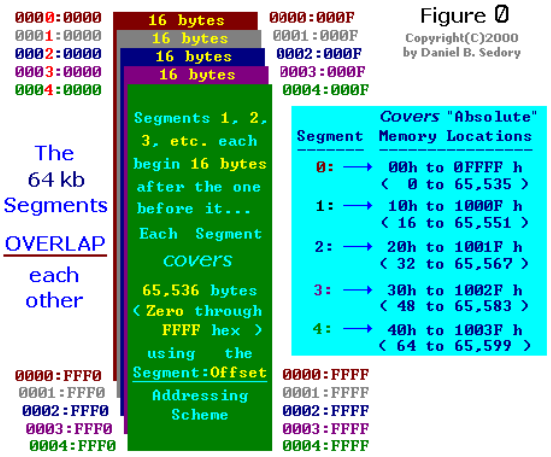
٤.٢.١ النمط المحمي Protected Mode

بعد أن تم التعرف على هذه المشاكل سارعت أنتل باصدار المعالج 80286 والذي كان أول معالج يعمل في نمطين (الحقيقي والمحمي) . هذا المعالج (والمعالجات التي تليها) حل أهم مشكلة وهي حماية مقاطع الذاكرة من الوصول العشوائي من قبل برامج المستخدم وذلك عن طريق وصف مقاطع الذاكرة وصلاحيات الوصول اليها في جداول تسمى جداول الوصفات (Descriptor Table). المعالج 80386 هو أول معالج ٣٢ بت ويستخدم مسجلات بحجم ٣٢ بت وحجم مسار البيانات أيضا بنفس الحجم مما سمح بإمكانية التعامل مع ذاكرة بحجم ٤ جيجابايت .كذلك تم اضافة دعم للذاكرة التخيلية ومفهوم الصفحات (Paging) ودعم تعدد

[٥] بحيث نعتبر عنوان المقطع هو عنوان بداية (Base Address) لعناوين القيم (Offset).

[٦] انظر الى مقالة الكاتب Daniel B. Sedory على الرابط http://mirror.href.com/thestarman/asm/debug/Segments.html.

شكل ٤.١: تداخل المقاطع في النمط الحقيقي



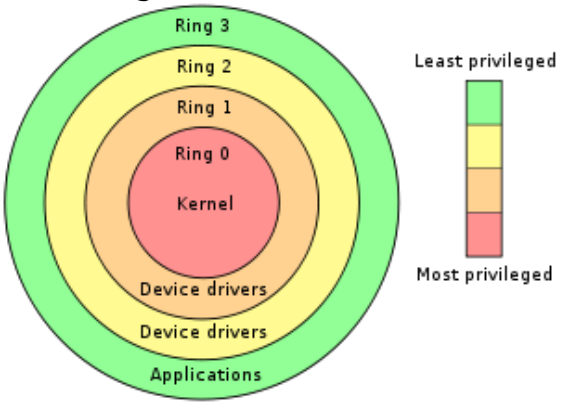
المهام. وفي هذا البحث سيتم الحديث عن معالجات ٣٢ بت باعتبارها أحد الأكثر انتشاراً حتى وقتنا هذا ،و على الرغم من ظهور معالجات ٦٤ بت إلا ان الدراسة حول معالجات ٣٢ بت تعتبر هي الأساس نظراً لان المعالجات الحديثة ما هي الا تطوير واضافات للمفاهيم الموجودة على المعالجات السابقة.

حلقات المعالج CPU Rings

عندما يعمل المعالج في النمط المحمي فان هذا يضمن حماية للذاكرة من برمجيات المستخدم ، وهذا بسبب توصيف الذاكرة وصلاحيات الوصول لها في جدول يستخدمه المعالج لعنونة الذاكرة وهو جدول الواصفات. نظام الصلاحيات الذي تم ادخاله الى المعالج عند عمله في النمط المحمي يسمى **بحلقات المعالج** (**CPU Rings**)، هذه الحلقات تحدد مستوى الحماية المطلوب لكي يستخدمها المعالج في تقرير ما اذا كان تنفيذ أمر ما يحتاج الى صلاحية أعلى أم لا، وكذلك لكي يقرر ما اذا كان الوصول الى عنوان معين في الذاكرة مسموحٌ باستخدام صلاحية معينة أم لا.وتوجد أربع حلقات للمعالج تبدأ من الحلقة صفر (Ring0) وتنتهي بالحلقة ٣ (Ring3). الحلقة صفر تسمى نمط النواة (Kernel Mode) بسبب أن أي برنامج يعمل في الحلقة صفر لديه الصلاحيات الكاملة على النظام بالوصول الى أي عنوان في الذاكرة وتنفيذ أي تعليمية حتى لو تسببت في ايقاف النظام عن العمل (المسؤولية تقع على البرنامج) لذلك غالبا البرامج التي تعمل في الحلقة صفر هي البرامج التي تتبع لنظام التشغيل. أما الحلقة ٣ تسمى بنمط المستخدم (User Mode) حيث أن البرامج التي تعمل عليها لا تملك صلاحيات لتنفيذ العديد من الأوامر (مثل الامر `cli` والأمر `hlt`) ولا تملك الوصول الى أي عنوان في الذاكرة بخلاف مساحة العنونة التخيلية (Virtual Address Space) الخاصة بالبرنامج نفسه وهذا ما رفع درجة حماية الذاكرة الى أقصى حد ممكن ، والشكل ٥.١ يوضح هذه الحلقات وصلاحياتها. وعندما يبدأ النظام بالإقلاع فان المعالج يكون في النمط الحقيقي وهو نمط لا يحوي على حلقات حيث أنه يمكن تنفيذ كل الأوامر والوصول الى أي عنوان في الذاكرة ، وعند التحويل الى النمط المحمي (PMode) فان المعالج يكون

في الحلقة صفر (Kernel Mode) ، ويتم تحويل الحلقة الى حلقة معينة تلقائيا عند نقل التنفيذ الى عنوان في الذاكرة موصوف في جدول الواصفات بأنه يعمل بتلك الحلقة.

شكل ٥.١: حلقات المعالج



٥.٢.١ معمارية معالجات x86

أي معالج يتعرف على مجموعة من الأوامر تسمى Instruction Set بعضاً منها تتطلب صلاحية معينة (الحلقة صفر) لكي يقوم المعالج بتنفيذها (انظر الجدول ٣.١ لمعرفة هذه الأوامر) وإلا فان هذا سيتسبب في حدوث خطأ من المعالج يسمأ العام (General Protection Fault) والذي ان لم تتوفر دالة تتعامل معه (Exception Handler) فان هذا يؤدي الى توقف النظام عن العمل.
وتحوي معالجات x86 العديد من المسجلات منها ما يستخدم للأغراض العامة (General Registers) ومنها ما يستخدم لحفظ العناوين وأرقام المقاطع (Segments Registers) وتوجد أيضا مسجلات لا يمكن استخدامها إلا في برامج الحلقة صفر (أي النواة) حيث أن التغيير فيها يؤثر على عمل النظام وأخيرا هناك مجموعة من المسجلات الداخلية للمعالج والتي لا يمكن الوصول لها برمجياً. والقائمة التالية توضح هذه المسجلات :

- مسجلات عامة : RAX (EAX(AX/AH/AL)), RBX (EBX(BX/BH/BL)), RCX (ECX(CX/CH/CL)), RDX (EDX(DX/DH/DL)).
- مسجلات عناوين:
  - مسجلات مقاطع:CS,SS,ES,DS,FS,GS .
  - مسجلات إزاحة: RSI (ESI (SI)), RDI (EDI (DI)), RBP (EBP (BP)). RSP (ESP (SP)), RIP (EIP (IP)).
- مسجل الأعلام: RFLAGS (EFLAGS (FLAGS)).
- مسجلات التنقيح: DR0, DR1, DR2, DR3, DR4, DR5, DR6, DR7.

جدول ٣.١: الأوامر التي تتطلب صلاحية الحلقة صفر

تنفيذ هذه الأوامر من قبل برمجيات المستخدم يؤدي الى حدوث خطأ وتوقف النظام عن العمل في حالة لم تتوفر دالة تتعامل مع هذا الخطأ.

الأمر	الوصف
LGDT	تحميل جدول الواصفات العام الى المسجل GDTR
LLDT	تحميل جدول الواصفات الخاص الى المسجل LDTR
LTR	تحميل مسجل المهام
MOV cr_x	نقل بيانات الى مسجل تحكم
LMSW	تحميل new Machine Status WORD
MOV dr_x	نقل بيانات الى مسجل تنقيح
CLTS	تصفير Task Switch Flag في مسجل التحكم الأول
INVD	Invalidate Cache without writeback
INVLPG	Invalidate TLB Entry
WBINVD	Invalidate Cache with writeback
HLT	إيقاف عمل المعالج
RDMSR	قراءة مسجل MSR
WRMSR	الكتابة الى مسجل MSR
RDPMC	قراءة Performance Monitoring Counter
RDTSC	قراءة time Stamp Counter

- مسجلات التحكم: CR0, CR1, CR2, CR3, CR4, CR8.
- مسجلات الإختبار: TR1, TR2, TR3, TR4, TR5, TR6, TR7.
- مسجلات أخرى: mm0, mm1, mm2, mm3, mm4, mm5, mm6, mm7, xmm0, xmm1, xmm2, xmm3, xmm4, xmm5, xmm6, xmm7, GDTR, LDTR, IDTR, MSR, and TR.

المسجلات العامة General Purpose Registers

في المعالجات ٣٢ بت يوجد ٤ أربع مسجلات عامة طول كل منها هو ٣٢ بت (٤ بايت) وتقسم أي من هذه المسجلات الى جزئين: الجزء الأعلى (High Order Word) وهو بطول ١٦ بت والجزء الأدنى (Low Order Word) وهو أيضا بطول ١٦ بت ، كذلك يُقسم الجزء الأدنى الى جزئين: الجزء الأعلى (High Order Byte) وهو بطول ٨ بت والجزء الأدنى (Low Order Byte) وهو أيضا بطول ٨ بت. على سبيل المثال مسجل EAX حيث يقسم الى جزء أعلى (لا يمكن الوصول اليه بشكل مباشر) وجزء أسفل وهو AX الذي يُقسم أيضا الى قسمين AH و AL. كل مسجل من هذه المسجلات العامة يستخدم لأي شيء لكن هناك بعض الإستخدامات الغالبة لكلٌ منهم توضحها القائمة التالية.

- المسجل EAX: يستخدم لنقل البيانات والعمليات الحسابية.

- المسجل EBX: يستخدم في الوصول للذاكرة بشكل غير مباشر وذلك باستخدام مسجل آخر يعمل كعنوان رئيسي Base Address.
- المسجل ECX: يستخدم في عمليات التكرار والعد.
- المسجل EDX: يستخدم في تخزين البيانات.

مسجلات المقاطع Segment Registers

مسجلات المقاطع تستخدم لتخزين أرقام وعناوين المقاطع (Segments) وتوجد ٦ مسجلات مقاطع تستخدم في النمط الحقيقي كما يلي:

- المسجل CS: يحوي عنوان بداية مقطع الشفرة للبرنامج المراد تنفيذه.
- المسجل DS: يحوي عنوان بداية مقطع البيانات للبرنامج المراد تنفيذه.
- المسجل SS: يحوي عنوان بداية مقطع المكدس للبرنامج المراد تنفيذه.
- المسجل ES: يحوي عنوان بداية مقطع البيانات للبرنامج المراد تنفيذه.
- المسجل FS: يحوي عنوان مقطع بعيد.
- المسجل GS: يستخدم للأغراض العامة.

أما في النمط المحمي (PMode) فإن هذه المسجلات لا تشير الى مقاطع البرامج والبيانات وإنما تشير الى واصفات معينة في جدول الواصفات العام ، هذه الواصفات تحدد عنوان بداية المقطع ونوع المقطع (يحوي شفرات أم بيانات) وتحدد صلاحية التنفيذ وصلاحية والقراءة والكتابة فيها – كما سنرى ذلك في الفصل الرابع بإذن الله–.

مسجلات الإزاحة Offset Registers

بجانب مسجلات المقاطع فإن الوصول الى الذاكرة في النمط الحقيقي يتطلب عنوان الإزاحة بداخل المقطع ، وتوجد ٤ مسجلات إزاحة في معالجات x86 حجم كل منها هو ٣٢ بت في الأنظمة ٣٢ بت و ١٦ بت في أنظمة ١٦ بت. والقائمة التالية توضح هذه المسجلات:

- المسجل SI: يحوي عنوان الإزاحة في مقطع البيانات.
- المسجل DI: نفس الوظيفة السابقة.
- المسجل BP: يحوي عنوان الإزاحة بداخل مقطع المكدس ويمكن استخدام للأشارة على أي عنوان في أي مقطع آخر.

- المسجل SP: يحوي عنوان الإزاحة بداخل مقطع المكدس.

مؤشر التعليمة Instruction Pointer

هذا المسجل (IP) يمثل إزاحة بداخل مقطع الشفرة (CS) وهو يحوي عنوان التعليمة التالية التي سيقوم المعالج بتنفيذها ، والعنوان CS:IP يمثل العنوان الفيزيائي للتعليمة التالية. هذا المسجل هو بطول ٣٢ بت (EIP) في أنظمة ٣٢ بت و ١٦ بت (IP) في أنظمة ١٦ بت، وهو مسجل لا يمكن تغيير محتواه باستخدام تعليمة المعالج MOV وإنما يتم تغيير محتواه عن القفز الى مكان آخر للتنفيذ.

مسجل الأعلام FLAGS Register

مسجل الأعلام هو مسجل بحجم ٣٢ بت (EFLAGS) في أنظمة ٣٢ بت و بحجم ١٦ بت (FLAGS) في أنظمة ١٦ بت ، وهذا المسجل هو عبارة عن بتات (بالحجم السابق ذكره) كل بت لديه وظيفه محدده ، وينقسم بشكل عام الى **بتات حالة** (**Status**) بحيث تعكس حالة الأوامر التي يقوم المعالج بتنفيذها و **بتات تحكم** (**Control**) بحيث تتحكم في بعض الخصائص و **بتات للنظام** (**System**). والجدول ٤.١ يوضح وظيفة كل بت في هذا المسجل.
ويحدد البتين IOPL مستوى الحماية المطلوب لتنفيذ مجموعة من الأوامر (مثل الأوامر CLI,STI,IN,OUT) حيث لن يتم تنفيذ مثل هذه التعليمات إلا في حالة كان مستوى الحماية الحالي Current Priviledge Level أعلى من أو مساوياً للقيمة الموجودة في البتين IOPL[٧] ، وغالباً ما تكون القيمة هي صفر دلالة على أن التعليمات السابقة لا يتم تنفيذها الا لبرامج النواة (Ring0).

مسجلات التحكم Control Registers

توجد في معالجات ٣٢ بت ستة مسجلات للتحكم في سلوك وعمل المعالج وهي CR0, CR1, CR2,CR3, CR4, CR8 ، ونظراً لخطورة التعامل معها فان هذه المسجلات لا يمكن الوصول لها إلا عند العمل في نمط النواة (Kernel Moder/Ring0) ولا يُمكن لبرمجيات المستخدم الوصول الى هذه المسجلات والتعامل معها. وفي الوقت الحالي يهمنا فقط أول مسجل تحكم وهو CR0 حيث من خلاله يمكن أو نقوم بعملية تحويل نمط المعالج من النمط الحقيقي الى النمط المحمي (PMode) وكذلك يمكن أن نقوم بتفعيل خاصية الصفحات (Paging) ، والتركيبة التالية توضح محتويات كل بت في مسجل التحكم CR0 وهو مسجل بحجم ٣٢ بت.

- Bit 0 (PE) : **Puts the system into protected mode.**
- Bit 1 (MP) : Monitor Coprocessor Flag This controls the operation of the WAIT instruction.
- Bit 2 (EM) : Emulate Flag. When set, coprocessor instructions will generate an exception

[٧] أعلى مستوى حماية هو الحلقة صفر (Ring0) ويليها الحلقة ١ ثم ٢ و٣.

جدول ٤.١: مسجل الأعلام EFLAGS

رقم البت	اسم البت	الإستخدام
0	CF	Carry Flag - Status bit
1	-	محجوزة
2	PF	Parity Flag
3	-	محجوزة
4	AF	Adjust Flag - Status bit
5	-	محجوزة
6	ZF	Zero Flag - Status bit
7	SF	Sign Flag - Status bit
9	TF	Trap Flag - System Flag
9	IF	Interrupt Enabled Flag - System Flag
10	DF	Direction Flag - Control Flag
11	OF	Overflow Flag - Status bit
12-13	IOPL	I/O Priviledge Level - Control Flag
14	NT	Nested Task Flag - Control Flag
15	-	محجوزة
16	RF	Resume Flag (386+ Only) - Control Flag
17	VM	v8086 Mode Flag (386+ Only) - Control Flag
18	AC	Alignment Check (486SX+ Only) - Control Flag
19	VIF	Virtual Interrupt Flag (Pentium+ Only) - Control Flag
20	VIP	Virtual Interrupt Pending (Pentium+ Only) - Control Flag
21	ID	Identification (Pentium+ Only) - Control Flag
22-31	-	محجوزة

- Bit 3 (TS) : Task Switched Flag This will be set when the processor switches to another task.
- Bit 4 (ET) : ExtensionType Flag. This tells us what type of coprocesor is installed.
  - 0 - 80287 is installed
  - 1 - 80387 is installed.
- Bit 5 (NE): Numeric Error
  - 0 - Enable standard error reporting
  - 1 - Enable internal x87 FPU error reporting
- Bits 6-15 : Unused
- Bit 16 (WP): Write Protect
- Bit 17: Unused
- Bit 18 (AM): Alignment Mask
  - 0 - Alignment Check Disable
  - 1 - Alignment Check Enabled (Also requires AC flag set in EFLAGS and ring 3)
- Bits 19-28: Unused
- Bit 29 (NW): Not Write-Through
- Bit 30 (CD): Cache Disable
- Bit 31 (PG) : **Enables Memory Paging.**
  - 0 - Disable
  - 1 - Enabled and use CR3 register