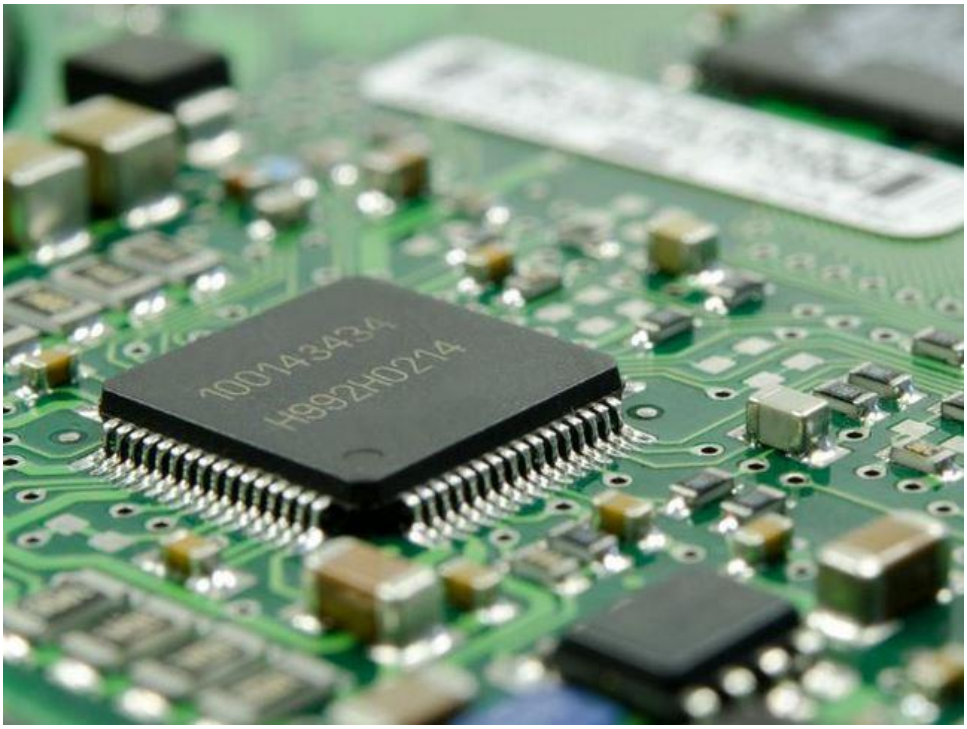
جامعة الحمدانية
كلية التربية

Lecture 8

Computer Architecture



Prepared By :

Firas Abdulrahman Yosif

Principle of Cache Memory Work

عند بدء تشغيل أي برنامج أو كتابة وقراءة أي بيانات أو تنفيذ أي عملية في النظام , يتم نقل البيانات المرتبطة بالعملية المحددة أولا من الهارد ديسك ذات التخزين بطيء الحركة إلى وحدة تخزين أسرع وهي الرامات (RAM random access memory) والمعروفة بذاكرة الوصول العشوائية وتكون من نوع DRAM Dynamic random access memory .

عندما يحتاج processor البيانات والاوامر المرتبطة بالعملية المحددة والمراد تشغيلها فعندئذ يجد الـ processor هذه البيانات موجودة داخل الرامات RAM ذات التخزين السريع مقارنة بالهارد ديسك والتي تمده بالبيانات بمعدل سريع بدلا من أن يأتي بها من الهارد ديسك ذو التخزين البطيء .

وبالرغم من سرعة الرامات الكبيرة إلا إذا أن هناك مشكلة وهي أن عمليات الـ processor تكون سريعة جدا وتكون الرامات غير قادرة على تلبية احتياجات الـ processor ولذلك يكون الحل أن تكون هناك بين الـ processor والرامات ذاكرة تخزين سريعة للغاية لمواكبة سرعة المعالج وتكون لها القدرة على سرعة تلبية الاحتياجات للمعالج من البيانات وهذه ستكون الذاكرة المؤقتة cache memory وهي أيضا نوع من ذاكرة الوصول العشوائي ولكن من النوع الساكن , Static RAM – SRAM وتكون ذات سرعة عالية.

Cache Memory Working:

عندما يطلب الـ CPU معلومة من الـ Cache وكانت المعلومة المطلوبة غير موجودة في الـ Cache ففي هذه الحالة تحدث عملية Miss ولكن الـ RAM تقوم بأعطاء اول معلومة من الـ block وتبعث المعلومة مع بقية الـ blocks المجاور لها الى الـ cache والسبب في نقل البلوك اكمله هو انه في المرحله المقبله سيحتاج المعالج تلك التعليمات المجاوره للتعليمه الاولى فتم وضعها في الـ cache لاخذها مباشره منه بدلا من جلبها من الذاكره الرئيسيه وبالتالي توفير الوقت والسرعة. وذلك لان الـ CPU سوف ينفذ الايعازات بشكل تسلسلي ولهذا السبب ترسل الذاكرة الرئيسية الـ block المجاور حتى يتم تنفيذه بشكل تسلسلي وتحدث في هذه الحالة عملية Hit (اي ان المعلومة موجودة في الـ cache).

ويمكن استخراج نسبة وجود المعلومة (Hit ratio) عن طريق القانون التالي:

$$\text{Hit ratio} = \frac{\text{no. of Hit}}{\text{no. of Hit} + \text{no. of Miss}}$$

EX) Assume no. of Hit= 80 find Hit ratio ?

$$\text{Hit ratio} = \frac{\text{no. of Hit}}{\text{no. of Hit} + \text{no. of Miss}}$$

$$\text{Hit ratio} = \frac{80}{80 + 20} = 0.8$$

تكون الحاسبة مثالية (perfect computer) عندما تكون Hit ratio = 1 وهذه الحالة تحدث عندما :

$$\text{no. of Hit} = 100$$

اي ان:

$$\text{Hit ratio} = \frac{100}{100 + 0} = 1$$

اذن الحاسبة تكون ذات كفاءة عالية (computer coefficient) عندما :

$$\text{Hit ratio} = \frac{\text{Hit}}{\text{no. of Hit} + \text{no. of Miss}} = \text{الناتج} \geq 0.9$$

اي ان :

$$\text{Hit ratio} \geq 0.9$$

عندما نستخدم الذاكرة المخبئية فأن الكمبيوتر يبحث عن المعلومة المطلوبة أولا بداخل الذاكرة المخبئية فأذا وجدها هذا يسمى اصابة للهدف (Cache hit) .

واذا لم يجد المعلومة يسمى هذا فقدان للهدف (Cache Miss) وفى هذه الحالة يجب على الكمبيوتر الانتظار لحين القيام برحلة إلى الذاكرة الرئيسية.

حيث ان كفاءة الـ Cache memory تقاس بكميه تدعى الـ Hit Ratio .

Cache Hit: تعني (البيانات موجودة او اصابة الهدف) وتحدث هذه العملية عندما الـ CPU يطلب معلومة من الـ Cache وهذه المعلومة تكون موجودة في الـ Cache memory .

Cache Miss: تعني (فقدان البيانات او عدم اصابة الهدف) و تحدث هذه العملية عندما الـ CPU يطلب معلومة من الـ Cache وهذه المعلومة تكون غير موجودة في الـ cache ولكن احتمالية المعلومة تكون موجودة في الـ main memory .

للحصول على (cache access time = main access time) وهذا يعني كلما الـ CPU يطلب معلومة يجدها في الـ cache اي انه كلما يكون الـ Hit ratio قريب من الواحد معناها كلما يطلب الـ CPU معلومة يجدها في الـ Cache memory وهو المطلوب.

We can increase Hit ratio by these factors:

1. Principle of Locality reference

تعتمد قدرة ذاكرة التخزين المؤقت على تحسين أداء الكمبيوتر على مفهوم المرجعية المحلية.

وتسمى **Locality reference** وتنص على ان البيانات التي تمت الإشارة عليها مسبقاً في الذاكرة الرئيسية تميل إلى إمكانية الإشارة إليها مرةً أخرى في وقت لاحق وهذا المبدأ يعتمد على عاملين اثنين هما:

- التواجد الزماني المؤقتTemporal locality : وفيه آخر تعليمة تم استدعاؤها من الذاكرة الرئيسية من المحتمل جداً أن يتم استدعاؤها مرة أخرى قريباً.

• التواجد المكاني المؤقتSpatial locality : يشير هذا إلى الوصول إلى البيانات أو الموارد المختلفة القريبة من بعضها البعض. وفيه التعليمة القريبة من التعليمة المستدعاة من الذاكرة الرئيسية من المحتمل جداً أن يتم استدعاؤها هي الأخرى قريباً .

إن الـ Temporal يقترح أنه عند الحاجة لمعلومة من الذاكرة الرئيسية فإنه يتم الاحتفاظ بنسخة منها في الـ Cache وستبقى فيه إلى حين استدعاؤها مرة أخرى من قبل الـ CPU ، وإن لم يتم ذلك في وقت قريب فإن ذاكرة الكاش تتخلص من هذه المعلومة ، بينما الـ Special يقترح بأنه بدلاً من جلب المعلومة المطلوبة في حد ذاتها ، بل يتم جلب الـ Block كاملاً أي المعلومة المطلوبة والمعلومات المجاورة لها لإمكانية احتياجها هي الأخرى بعد استدعاء التعليمة السابقة وذلك لأنه قد تكون هذه التعليمات متتابعة ومبنية على بعضها البعض في النظام .

المعلومة المطلوب تنفيذها تكون متسلسلة وليست مبعثرة فلو كانت مبعثرة ستكون نسبة الـ Miss اكبر من الـ Hit مما يؤدي الى فشل cache .

تصف المنطقة المحلية **Locality reference** المواقف المختلفة التي تجعل النظام أكثر قابلية للتنوقع. تستفيد ذاكرة التخزين المؤقت (cache) من هذه المواقف لإنشاء نمط للوصول إلى الذاكرة الرئيسية يمكن الاعتماد عليها.

ان مبدأ المرجعية المحلية يدعم فكرة الـ cache ويحقق اداء افضل وسرعة اكبر.

2. Subroutines Programs

ممكن زيادة الـ Hit ratio باستخدام البرامج الفرعية (subroutine program) داخل البرنامج الرئيسي حيث نسبة احتمالية العثور على المعلومة داخل الـ cache memory تزداد وبالتالي تزداد الـ Hit ratio .

3. Loops Instruction

ايعاز الدوران من الايعازات البرمجية المستخدمة في البرامج وهي تمثل الحلقات مثل (for , while , do - while) حيث ان استخدامها من قبل الـ user داخل البرنامج تزيد من احتمالية العثور على المعلومة داخل الـ cache وبالتالي تزداد نسبة Hit ratio وتقل نسبة الـ Miss .

4. Arrays

تستخدم المصفوفات لزيادة الـ Hit ratio في الـ cache حيث ان المصفوفة تخزن البيانات في الذاكرة سواء كانت المصفوفة احادية او ثنائية حيث تخزن البيانات بشكل صف او عمود فعندما يطلب الـ cache اول معلومة من المصفوفة ومن اول موقع مثلا A[0] تكون عملية Miss ثم بعد ذلك تقوم الـ main memory بارسال المعلومة مع الـ block المجاور لها وترسلها الى الـ cache ثم بعد ذلك يقوم المعالج بالبحث عن المعلومة داخل المصفوفة وبذلك تحدث زيادة في احتمالية العثور على المعلومة داخل الـ cache وبالتالي تحدث زيادة في عملية الـ Hit وتقليل في عملية الـ Miss .

The factors that increase miss ratio:

1. Go to Instruction

ايعاز الـ Goto احد الايعازات التي تقلل من سرعة العثور على المعلومة اي Hit ratio وفي هذه الحالة تزداد فرصة عدم العثور على المعلومة في الـ cache وهذه العملية تسمى الـ Miss ratio.

2. JMP Instruction

ايعاز الـ JMP يعتبر من الايعازات التي تقلل من سرعة العثور على المعلومة في الـ cache اي ان الـ Hit ratio تقل وفي هذه الحالة تزداد فرصة عدم العثور على المعلومة في الـ cache وهذه العملية تسمى الـ Miss ratio.

والمبرمج الجيد هو الذي يقلل من استخدام ايعازات الـ Goto وايعاز القفز الـ JMP في البرنامج والتي بدورها تقلل من سرعة العثور على المعلومة في الـ cache وتعمل على زيادة الـ Miss .