

جامعة المنارة

كلية:الهندسة.....

قسم:المعلوماتية.....

اسم المقرر:بنيان حاسوب 1.....

رقم الجلسة (6+5)

عنوان الجلسة

.....الذواكر...



العام الدراسي

الفصل الدراسي

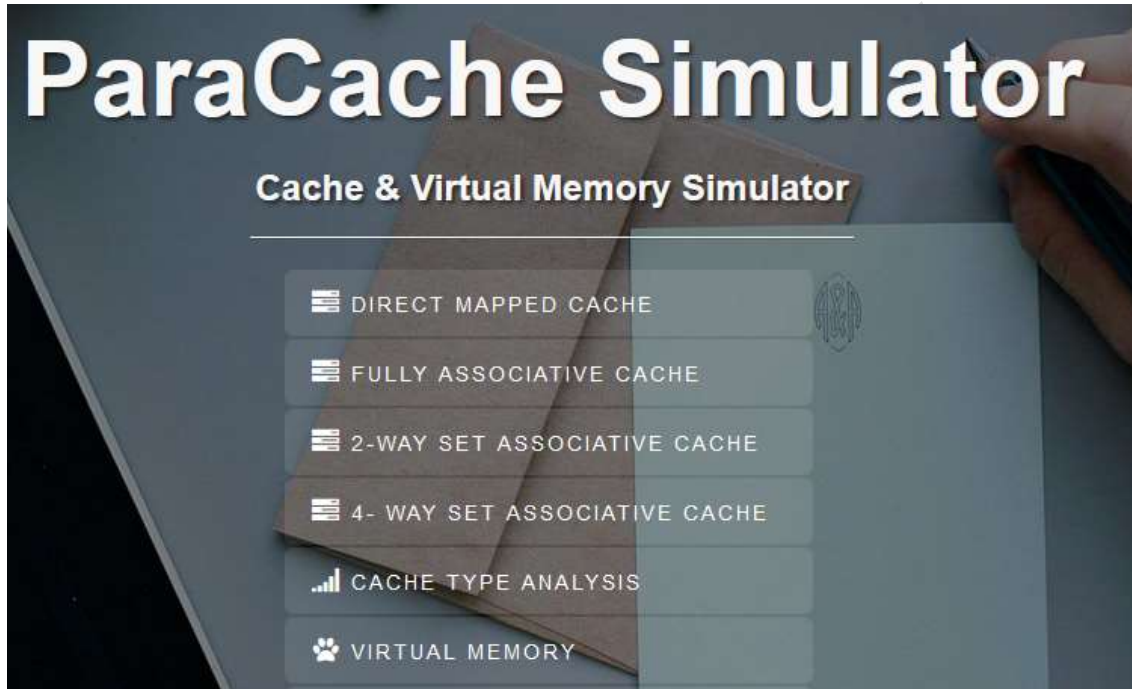
الغاية من الجلسة:

تعريف الطالب بقنيات الCaching المستخدمة

مقدمة:

يوجد نماذج مختلفة من أنماط عمل الذاكرة المخبئية مثل أنماط n -way , fully associative , Direct mapping associative وغيرها من الأنماط , سنقوم بالتدريب على هذه الأنماط اعتماداً على برمجية مخصصة لهذه المهمة.

عند فتح البرنامج سيكون لدينا النافذة التالية:



نختار منها نمط التخزين المطلوب.

النمط الأول: Direct Mapped Cache:

بفرض لدينا ذاكرة بحجم 512 كيلوبايت , وذاكرة كاش بحجم 256 بايت وعدد بتات الإزاحة هو 2. والمطلوب:

- 1- حدد قيم TAG,INDEX,OFFSET في هذا النظام.
- 2- عند إدخال القيمة 2FF في الذاكرة , المطلوب تحديد مكان التخزين لهذه القيمة.
- 3- ماهو حجم الجدول Table الخاص بهذه الذاكرة؟

الحل:

يجب إدخال قيم الذاكرة بالبايت , أي القيمة المحددة بالبايت تدخل على شكل بايت , والقيمة بالكيلوبايت يجب أن تحول إلى بايت عن طريق ضربها بالقيمة 1024.

ملاحظة (عندما تكون القيم المعطاة لا تحقق القيم الثنائية المشهورة عندها يجب أن نأخذ القيمة الثنائية التالية, على سبيل المثال إذا كانت ذاكرة الكاش 302 وهي قيمة غير مشهورة عندها نأخذ القيمة الثنائية التالية وهي 512.).

نقوم بتحويل قيمة الذاكرة الرئيسية $512 * 1024 = 524288$ main memory

ذاكرة الكاش 256 بايت

بتات الإزاحة 2

ندخل هذه القيم وننقر على زر submit.

تظهر القيم التالية في الذاكرة:

TAG	INDEX	OFFSET
11 bit	6 bit	2 bit

إذا أخذنا القيم مرفوعة للقوة عندها سنجد أن الذاكرة الرئيسية $2^{10} * 2^9$ وبالتالي عدد البتات الكلي 19 يحجز منها اثنان للإزاحة.

Offset = 2 bits

Index bits = $\log_2(256/4) = 6$ bits

Instruction Length = $\log_2(524288) = 19$ bits

Tag = 19 bits - 2 bits - 6 bits = 11 bits

Block = 11 bits + 6 bits = 17 bits

2- نقوم بإدخال القيمة 2FF في الموقع المخصص instruction ومن ثم submit وبعد ذلك fast forward من أجل القيام بعملية التنفيذ. يتم فك هذه التعليمية وتحويلها للنمط الثنائي ووضعها في الجدول السابق لتحديد القيم الرئيسية. نلاحظ أن القيمة تم تخزينها في الموقع الأخير من ذاكرة الكاش.

3- يتعلق حجم الجدول بقيمة index وبما أنها في حالتنا هذه 6 وبالتالي فإن حجم الجدول 64 تتراوح من الموقع 0 وحتى الموقع 63.

ماهو HIT,MISS:

تحدد هذه العبارتين إمكانية تواجد القيمة في الذاكرة أو أننا هل استخدمنا موقع الذاكرة المحدد أم لا , فإذا وجدت القيمة بالموقع أو أن هذا الموقع قد استخدم سابقاً بالتالي لدينا HIT وإلا فإننا سنكون في حالة MISS.

الحالة الثانية :

طبق نفس المثال السابق على ذاكرة من نمط Fully associative cache.

سنقوم بإدخال نفس القيم وسينتج لدينا نموذج الجدول التالي:

BLOCK	OFFSET
17 bit	2 bit

حيث نتجت هذه القيم من الحسابات التالية.

Offset = 2 bits

Instruction Length = $\log_2(524288) = 19$ bits

Block = 19 bits - 2 bits = 17 bits

عند إدخال القيمة 2FF سيتم تحويلها بنفس الطريقة وتحديد موقع التخزين. تم التخزين في الكتلة BF والتي نتجت من فك تشفير القيمة كما في الجدول التالي.

BF	
0000000010111111	11
17 bit	2 bit

الحالة الثالثة 2-WAY SA:

بنفس الخائص المقدمة في المثال سيكون لدينا الجدول التالي:

TAG	INDEX	OFFSET
12 bit	5 bit	2 bit

والتي نتجت أيضاً من الحسابات التالية.

Offset = 2 bits

Index bits = $\log_2(256/4/2) = 5$ bits

Instruction Length = $\log_2(524288) = 19$ bits

Tag = 19 bits - 2 bits - 5 bits = 12 bits

Block = 12 bits + 5 bits = 17 bits

وعند إدخال القيمة 2FF سيكون لدينا:

00000000101	1111	11
12 bit	5 bit	2 bit

ولكن كيف يمكننا استنتاج موقع التخزين؟



سيتم التخزين في الكتلة BF. في حين تقوم القيم الأولى TAG بتحديد موقع الجدول المعتبر لعملية التخزين.

جامعة المنارة