➔ **Tema 1:** Procesadores Multinúcleo e Perspectivas de Escalamento

1. Introducción
2. Tecnoloxías de Fabricación e Escalamento
    a. Escalamento das dimensións
    b. Escalamento da potencia
    c. Escalamento do retardo dos circuítos
3. Escalamento Hardware dos Microprocesadores
4. Escalamento de Prestacións en Microprocesadores
    a. Árbol de Ecuacións
    b. Modelo de Escalamento
    c. CPI vs. Mr
5. Final dos Procesadores cun só Núcleo
6. Arquitecturas alternativas segundo o tipo de carga de traballo
7. Comparación de sistemas: conxuntos de programas de referencia (benchmarks)

➔ **Tema 3:** Núcleos de Procesamento - Segmentación (Pipelining)

1. Introducción
2. Gañancia en velocidade coa segmentación e limitacións
3. Exemplo de pipeline (básico)
    a. Implementación de instrucións tipo ALU
    b. Implementación de instrucións tipo Load/Store
    c. Implementación de instrucións de salto
    d. Melloras do Pipeline
4. Control do Pipeline
5. Reducción da Penalización por Saltos
6. Extensión a Operacións Multiciclo e Varias Unidades
7. Excepcións e Pipelining
8. Exemplo de Pipeline Realista (sinxelo): MIPS R4000

➔ **Tema 4:** Núcleos de Procesamento – Paralelismo a Nivel de Instrucción

1. Introducción
2. Planificación Dinámica Baseada en Marcador
3. Predición de Saltos
    a. Política de cambio de predición
    b. Información Local
    c. Información Global
    d. Predictores Combinados
    e. Predictores Específicos para Lazos
    f. Predición do Enderezo de Destino do Salto
    g. Predición de enderezo de retorno de chamadas
4. Arquitectura dun Núcleo con Paralelismo a Nivel de Instrución
    a. Renomeado de Rexistros
    b. Estacións de Reserva
    c. Búfer de Reordenamento de Instrucións (Reorder Buffer en inglés)
    d. Búfer de Reordenamento de Operacións de Memoria (MOB)
    e. Arquitectura do Núcleo
    f. Arquitectura de Núcleos Industriais
    g. Escalamento dos núcleos
    h. Arquitecturas superscalares alternativas: VLIW

➔ **Tema 5:** Subsistema de Memoria Compartida en Procesadores Multinúcleo

- Variantes MESI
    1. Introducción
    2. Variantes de MESI: MESIF e MOESI
- Protocolos de Coherencia
    1. Introducción
    2. Protocolos de coherencia caché
    3. Protocolos de snooping (arquitecturas UMA)
    4. Protocolos basados en directorios (arquitecturas CC-NUMA)

EN AMARILLO LAS COSAS QUE ESTÁN PENDIENTES

2017_mayo

Asumindo un pipeline básico ¿Qué fases precisa unha instrucción STORE?

d. IF, ID, EX, MEM

❑ Executing a MIPS instruction can take up to five steps.

Step	Name	Description
Instruction Fetch	IF	Read an instruction from memory.
Instruction Decode	ID	Read source registers and generate control signals.
Execute	EX	Compute an R-type result or a branch outcome.
Memory	MEM	Read or write the data memory.
Writeback	WB	Store a result in the destination register.

❑ However, as we saw, not all instructions need all five steps.

Instruction	Steps required				
beq	IF	ID	EX		
R-type	IF	ID	EX		WB
sw	IF	ID	EX	MEM	
lw	IF	ID	EX	MEM	WB

Nun protocolo baseado en directorio (asumimos protocolo MESI nas cachés) suponemos que o procesador do nodo solicitante erra ao escribir na caché. O nodo orixe é un nodo remoto e o estado do bloque no directorio é exclusivo. ¿Cal será o novo estado do bloque no directorio e na caché do nodo solicitante?

Exclusivo y modificado respectivamente (en directorio modificado implica siempre exclusivo).

Nodo solicitante → PtLecEx; Nodo origen genera invalidaciones, estado en directorio a exclusivo. Los nodos con copia responden al nodo solicitante con reconocimientos a las invalidaciones. Nodo solicitante puede proceder; pero debe detener cualquier petición posterior o postecrituras hasta recibir confirmaciones de invalidación.

¿Para que tipos de problemas son adecuadas as tarxetas gráficas (GPUs)?

d. Aplicacións cun elevado número de fíos que aplican kernels de computación sobre streams de datos.

As tarxetas gráficas, no seu camiño de ampliar o eido de computación no que poden operar, contan na actualidade con moitas unidades deste tipo, además do hardware especializado para taréfas de baixo nivel de procesamento gráfico.

¿Cal das seguintes afirmacións sobre o MOB é FALSA?

Según yo todas son verdaderas:

- No MOB as escrituras en memoria fanse cando se dispoña de suficiente ancho de banda

  Verdadera. Según el tema: "Almacenar nunha cola ordenada os resultados que se escribirán en memoria por instrucións Store xa completadas. Cando se dispoña de ancho de banda dispoñible coa memoria, realízanse as escrituras en orde dos Stores completados (retiradas dos Stores)."

- O MOB informa o ROB sobre a especulación nos Loads

  Verdadera. El MOB chequea la dirección donde va a leer el Load y la dirección dónde van a escribir los Stores que están en la cola. Si hay una dependencia, se especuló mal y el Load trajo un valor que no es válido (suele obtener el del propio store). Si se especuló bien la especulación trajo el valor correspondiente.

- O MOB facilita a execución fóra de orde dos Stores.

  Si no hay conflictos, los Stores se pueden evaluar (conseguir la dirección de destino) fuera de orden, pero el MOB ayuda a que los cambios se escriban en memoria siguiendo el orden del programa.

Na planificación baseada en marcador, ¿Qué afirmación é FALSA?

d. A execución despois da etapa de Emisión de Instruccións é fóra de orde.

Emisión de instrucciones (IE): en orden, frenando los WAW
Lectura de operandos (Register Read RR): fuera de orden frenando los RAW

WAR frenados en WB.

Se quero realizar a predicción do enderezo de destino dun salto, utilizarei:

d. Ningunha das anteriores.

O predictor do enderezo de destino do salto (Branch Target Buffer en inglés – BTB), é unha cache indexada pola parte menos significativa do enderezo de salto, e onde as etiquetas (tags en inglés) son a parte máis significativa do devandito enderezo.

¿Cal das seguintes características non se corresponde cos procesadores x86 de Intel?

b. Teñen un xogo de instruccións RISC.

Arquitecturas x86 usan el juego de instrucciones CISC, que se descomponen en microoperaciones RISC.

Os tempos de acceso a memoria principal son da orde de:

b. Centenas de ciclos

A memoria principal, externa ao chip do procesador, ten na actualidade uns tempos de acceso da orden de centos de ciclos de reloxo do procesador. -Tema 1

Teño un PC na casa con dúas CPUs multicore (8 núcleos cada unha) de Intel de última xeración. Polo tanto a miña máquina:

a. Cunha alta probabilidade soporte o protocolo MOESI.

Falso. *Asociados a estes sistemas de interconexión que permiten mensaxes de coherencia están as variantes do algoritmo MESI: MOESI (Hypertransport – AMD) e MESIF (QPI – Intel).*

b. Os procesadores usan un sistema de conexión coherente HyperTransport.

Falso. *Actualmente os sistemas de interconexión coherente a nivel industrial máis relevantes son QPI (Quick Path Interconnect – asociado a Intel) e HT (Hypertransport – asociado a AMD).*

c. É un sistema NUMA (con respecto á memoria principal).

Na implementación do algitmo MESI convencional nun sistema NUMA con conexións punto a punto, cando se produce un fallo de lectura na cache dun procesador, este retransmite a todos os procesadores a petición de lectura.

d. Ningunha das anteriores.

¿Cal das seguintes afirmacións sobre as Estacións de Reserva é VERDADEIRA?

a. O tamaño das Estacións de Reserva ten influencia directa no CPI do núcleo.

Se seguirán despachando instrucciones y la estación de reserva las tendrá en cola hasta enviarlas a ejecución, por lo que las etapas anteriores no están esperando a que acabe la etapa EX para mandar instrucciones.

b. O despacho de instruccións realízase fóra de orde.

Falso. Se despachan intrucciones en orden.

c. Cada entrada na estación de reserva correspóndese a unha instrucción que xa ten os seus operandos disponibles.

No, está en la estación de reserva esperando a que los operandos estén disponibles.

d. As Estacións de Reserva centralizadas son pouco flexibles.

Las centralizadas son más flexibles pero más complicadas

¿Cal é a función do estado O no protocolo MOESI?

O algoritmo MOESI incorpora un novo estado: O (Owned). Este estado indica que a liña cache foi modificada e logo requirida para lectura por outras caches. Así se unha copia está en estado M, e chega unha mensaxe cun requirimento de lectura por parte doutra cache, a liña pasa de estado M a O, e envíaselle unha copia da liña á cache que o requiriu. As peticións posteriores de lectura tamén serán satisfeitas pola cache que ten a liña en estado O. Se unha liña en estado O é desaloxada, entón é necesario actualizar a memoria, xa que a copia existente en memoria está desfasada. O que se pretende evitar é que cando hai unha copia en estado M, sexa necesario facer a actualización de memoria de xeito inmediato (este proceso ten unha latencia moi elevada). Polo tanto, deste xeito, evítase a actualización da memoria e envíase directamente unha copia a outras caches, facendo que a transacción completa teña menor latencia.

En cristiano: Si alguien pide leer una línea, quien tenga el estado M se la ofrece y pasa a estado O. Posteriores lecturas serán servidas por la línea en estado O. Si esta es desalojada, entonces se debe actualizar la memoria.

¿Qué explica a Ley de Moore?

La ley de Moore expresa que aproximadamente cada dos años se duplica el número de transistores en un microprocesador.

¿Cal é a razón pola que resulta máis interesante anticipar o antes posible a execución dos Load fronte aos Store?

Son más frecuentees, suelen inciar un bloque de compilación que luego tendrá dependencias en los valores que se cargan y pueden preservar elevada latencia por fallos en la caché y búsqueda en memoria, por lo que hay que saber lo antes posible la dirección en la que se va a leer.

¿Qué modela R no modelo de escalamento de prestacións en mciroprocesadores?

Factor en el que se reduce la cache total visto por cada hilo en cada núcleo, puede ir desde 1 hasta Núm de núcleos x Hilos por núcleo.

Cando medimos a gañancia en velocidade (S) que resulta de usar segmentación (pipelining) fronte unha CPU non segmentada, o retardo dunha etapa do pipeline descomponse como a suma de varios tempos, ¿cales son?

Tiempo de registro, tiempo de ejecución y tiempo de retardo para balancear las etapas.

T_R : retardo do rexistro (inclúe a sobrecarga por sincronización, etc).
T_S : é o retardo por etapa ideal correspondente á parte combinacional. Polo tanto para un pipeline de N etapas, este valor é T_S =T_COMB / N
T_OV : é o tempo adicional do ciclo de reloxio debido a un balanceo non ideal das etapas do pipeline.

¿De que xeito o Buffer de Reordenamento de Instruccións (ROB) facilita a especulación?

O ROB facilita a especulación (por exemplo por dependencias de control: saltos), de tal xeito que se incorpora información ás entradas para indicar as instrucións que están en estado especulativo. Unha entrada especulativa non pode ser retirada/completada até que pase a ser non especulativa (por exemplo cando se resolve o salto). **Se o resultado da acción especulativa foi negativo, invalídanse as entradas do ROB correspondentes as instrucións especulativas dependentes desa acción (co cal nunca serán completadas/retiradas, e non producirán cambios no estado da máquina).**

En canto á especulación, debe recibir información de si a acción especulativa tivo éxito ou non. É necesario que reciba información do enderezo da instrución a partir do cal as instrucións son especulativas, para pasalas a estado non especulativo se a especulación tivo éxito ou para invalidalas en caso contrario.

Indicar as acción que ocorren nun procesador multicore con 4 núcleos, co último nivel de caché compartido e un primeiro nivel de caché privado, con interconexión entre as cachés privadas e a compartida en bus co protocolo de coherencia snooping (MESI), cando se realizan as siguientes referencias de memoria a datos compartidos (suponer que os datos están inicialmente na memoria principal, e que A e B están na mesma liña caché mentres que o resto de datos están en liñas distintas). As referencias ocorren na orde indicada (indicamos o núcleo e a acción que realiza), e con tempo suficiente entre elas para evitar conflictos nas transaccións.

P0 (Load A), P1 (Load B), P2 (Load C), P3 (Load D), P2 (Store A), P1 (Store C), P1 (Load D), P3 (Store B), P0 (Load C).

- A y B en la misma línea (ojo al false sharing), el resto en distintas
- primer nivel privado, último nivel compartido
- protocolo MESI

➔ P0 (Load A)
Está todo en memoria principal, se carga A en la caché compartida y en la privada del núcleo P0 donde se marca el bloque como Exclusivo.
➔ P1 (Load B)
B está en la misma línea que A, se lee desde la caché compartida, en las privadas de P0 y P1 se marca como Shared.
➔ P2 (Load C) y P3 (Load D)
Está todo en memoria principal, se carga C y D en la caché compartida y en las privadas de los núcleos P2 y P3 donde se marca los bloques como Exclusivo.
➔ P2 (Store A)
P2 solicita acceso exclusivo de A y lo marca Modificado, se invalida la copia de P0.
➔ P1 (Store C)
P1 solicita acceso exclusivo de C y lo marca Modificado, se invalida la copia de P2.
➔ P1 (Load D)
P1 solicita lectura de D se marca el bloque en P1 y P3 como compartido.
➔ P3 (Store B)
P3 solicita acceso exclusivo de B y lo marca modificado, se invalida el bloque en P1 y como A está en el mismo bloque también se invalida el bloque que está en P2 (false sharing).
➔ P0 (Load C)
P1 solicita lectura de C y se lee desde P1 que lo tiene modificado, en ambos se marca el bloque como compartido.

Un programa corre un procesador segmentado presentando un CPI de 1.6. O sistema admite ata unha instrucción por ciclo. O programa executa un 20% de instruccións de salto e a taxa de acerto na predicción é do 75%. O custo dun salto mal predito é de 6 ciclos, e o custo dos saltos ben preditos é cero.

1.6 ciclos/instruccion
0.2 saltos/instruccion
0.25 fallos/salto
6 ciclos/fallo

- **Calcula o CPI se melloramos o preditor de saltos para que teña unha taxa de acertos do 90%.**

Suponiendo que en el CPI anterior están incluído la penalización por fallos primero hay que quitársela:

0.2 saltos/instruccion * 0.25 fallos/salto * 6 ciclos/fallo = 0.3 ciclos/instrucción → CPI = 1.6 - 0.3 = **1.3 = CPI**

Ahora calculamos con la tasa de aciertos de 90%

0.2 saltos/instruccion * 0.1 fallos/salto * 6 ciclos/fallo = 0.12 ciclos/instruccion → CPI = 1.3 + 0.12 = **1.42 = CPI**

- **Calcular o CPI se además se aplica a técnica do salto retardado supoñendo que, no código, o compilador sempre atopa unha instrucción para aplicala, e consegue o máximo beneficio que esta permite.**

Según el tema 3:
Salto retardado ("delayed branch" en inglés):O compilador colocará logo da instrución de salto certo número de instrucións (tantas coma sexa necesario até que se resolva o salto e se coñeza o destino do salto) que permitan facer traballo (executar instrucións) que probablemente resulte útil. Polo tanto, logo dun salto, nunca parará a execución, e pode ser necesario converter estas instrucións en NOP antes de que cambien o estado (se a transformación non é compatible co resultado real do salto durante a execución).

Respecto das instrucións de salto, nesta arquitectura transcorren tres ciclos dende o inicio da instrución até que se coñece o destino do salto. (computado no ciclo EX). Un dos ciclos serve de "slot" para un salto retardado, intentando incorporar unha instrución no primeiro ciclo logo da instrución de salto, segundo as técnicas estudadas anteriormente. Nos outros dous ciclos séguese a estratexia de predicir o salto coma non tomado.

Basicamente cambian los ciclos de penalización por fallo, lo que no sé es si es 0 o 1.
Corrección hecha por Dora:
Hay que considerar a mayores que en este caso el coste de cada salto será un ciclo menos porque uno de ellos se aprovecha con el salto retardado con una ranura de retardo. Serán 5 ciclos por cada salto mal predicho. Esto nos da un CPI = 1,4

Un procesador acada unha frecuencia de reloxo de 4 GHz nunha tecnoloxía CMOS de 32 nm. Supoñer que o mesmo procesador (so alterando potencialmente a ubicación dos rexistros do pipeline) é implementando nunha nova tecnoloxía de 14 nm e cunha frecuencia de reloxo de 3GHz. Cos datos proporcionados, facer unha estimación de primeira orde do ratio entre as profundidades do pipeline de ambas implementacións, argumentando as suposicións realizadas (o cálculo exacto non é posible facelo cos datos que se proporcionan).
DATOS:

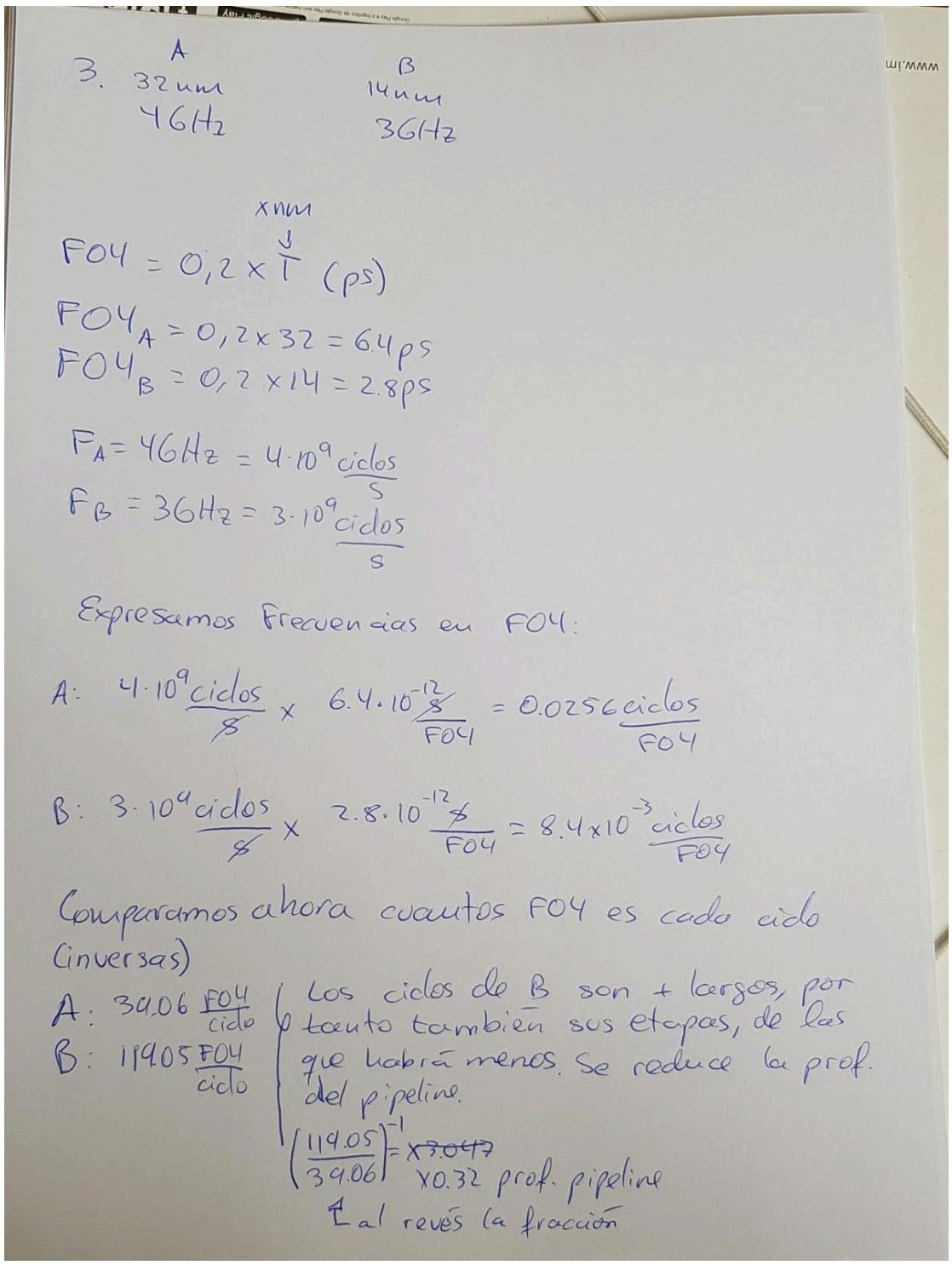
F_1 = 4 GHz
T_1 = 32 nm

F_2 = 3 GHz
T_2 = 14 nm

FORMULAS:

FO4 = 0.2 x T
TFO4 = número de retardos de FO4 por etapa do pipeline

Tc = TFO4 x FO4 = segundos/ciclo
F = 1 / Tc = ciclos/segundo



2017_julio

El escalamiento real de un procesador de un nodo al siguiente:

D. Ninguna de las anteriores

Según el tema 1: *Podemos esperar factores de escalamento de área moi próximos a x0.5 para arranxos de memoria cache, pero para o caso dos núcleos de procesamento os factores de escalamento poden ser de x0.6 ou x0.7.*

¿Qué mide LM en el modelo de escalamiento de prestaciones en microprocesadores?

D. A latencia media por acceso á memoria principal

$$CPI_{core} = CPI_{core|} + Mr \times (LM+QE)$$

Ciclos por instruccion en cada core, con dos partes: suponiendo que no hay fallos en últ. caché y la penalización de ir a memoria (Mr: tasa fallos y QE: latencia por esperar en cola)

As unidades nas que se mide a Intensidade Operacional

D. Ningunha das anteriores

Es instrucciones/bytes. Según el tema 1: A intensidade operacional dun programa atópase dividindo o número de instrucións por segundo (F (ciclos/segundo) / CPI (ciclos/instrución)) polo ancho de banda demandado BW (bytes/segundo) resultando as unidades instrucións/byte = (instrucións/segundo) / (bytes/segundo).

Cal das seguintes afirmacións sobre o renomeado de rexistros é falsa?

D. Elimínanse as dependencias WAW, WAR e RAW.

Las dependencias puras RAW no se pueden eliminar. El renombreado sí que elimina WAW y WAR.

Dun nodo tecnolóxico ao seguinte, a área dun transistor escala nun factor:

D. x0.5

Según el tema 1: *Dun nodo tecnolóxico ao seguinte (aproximadamente cada dous anos aparece un novo nodo) tradicionalmente as dimensións lineais dos transistores e conexións escalaban por un factor 0.7. Isto implicaba un escalamento ideal na área que ocupaban os elementos de x0.5. Polo tanto un microprocesador cunha área de 2cm² na tecnoloxía de 65 nm ocuparía idealmente unha área de 1cm² na tecnoloxía de 45 nm (= 65 x 0.7).*

Cal das seguintes NON se corresponde coa planificación baseada en marcador?

A.Soporta excepcións precisas

Segun el tema 1, sobre la planificación con marcador:

- *Non tivemos en conta a posible presencia de instrucións de salto. Só analizamos bloques de instrucións consecutivas.*
- *A efectividade da planificación dinámica, depende do paralelismo dispoñible entre instrucións (dependencias), da ventá de instrucións consideradas, e do tipo e número de unidades funcionais.*
- *O estado da máquina e da memoria actualizase fora de orde, co que non soporta execcións precisas.*

Cal das seguintes afirmacións sobre o MOB é falsa?

Según yo todas son verdaderas:

- No MOB as excrituras en memoria fanse cando se dispoña de suficiente ancho de banda

Verdadera. Según el tema: "Almacenar nunha cola ordenada os resultados que se escribirán en memoria por instrucións Store xa completadas. Cando se dispoña de ancho de banda dispoñible coa memoria, realízanse as escrituras en orde dos Stores completados (retiradas dos Stores)."

- O MOB informa o ROB sobre a especulación nos Loads

Verdadera. El MOB chequea la dirección donde va a leer el Load y la dirrección dónde van a escribir los Stores que están en la cola. Si hay una dependencia, se especuló mal y el Load trajo un valor que no es válido (suele obtener el del propio store). Si se especuló bien la especulación trajo el valor correspondiente.

- O MOB facilita a execución fóra de orde dos Stores.

Si no hay conflictos, los Stores se pueden evaluar (conseguir la dirección de destino) fuera de orden, pero el MOB ayuda a que los cambios se escriban en memoria siguiendo el orden secuencial del programa.

P é a probabilidade de obter o resultado dunha instrucción por ciclo. Cal é falsa?

A. Si hay más dependencias y saltos, P aumenta

Falsa, es al revés. Hay menos probabilidad de obtener un resultado

Canto máis pequeno é o ciclo de reloxo

D. Ninguna de las anteriores.

Descartas A y C pq puede que otros factores hayan hecho más pequeño el ciclo de reloj (mejor predicción de saltos por ejemplo).
B pone la penalicación en ciclos, puede que la penalización sea menor porque el ciclo es más corto pero el núm de ciclos será el mismo.

Nun protocolo baseado en directorio (asumimos protocolo MESI nas cachés) suponemos que o procesador do nodo solicitante erra ao escribir na caché. O nodo orixe é un nodo remoto e o estado do bloque no directorio é exclusivo. ¿Cal será o novo estado do bloque no directorio e na caché do nodo solicitante?

> Exclusivo y modificado respectivamente (en directorio modificado implica siempre exclusivo).
>
> *Nodo solicitante → PtLecEx; Nodo origen genera invalidaciones, estado en directorio a exclusivo. Los nodos con copia responden al nodo solicitante con reconocimientos a las invalidaciones. Nodo solicitante puede proceder; pero debe detener cualquier petición posterior o postecrituras hasta recibir confirmaciones de invalidación.*

En qué consiste y cuándo se produce False Sharing?

Consiste en que se almacenen en una misma línea cache datos de dos variables sin relación entre ellas, toda la línea se mueve aunque solo se esté accediendo a una de las variables, esto puede provocar que se invalide toda la línea al modificar solo una de las variables.

Cando falamos dunha caché unificada a que nos estamos a referir?

Que dentro de ella se almacenan tanto datos como instrucciones. OJO no confundir con compartida.

La potencia de un circuito CMOS depende de la frecuencia y del voltaje. Si quieres reducir el consumo, ¿qué parámetro modificas? Pq?

Se puede aumentar la latencia aunque serís muy contraproducente (^3). Se modifica el voltaje:

O efecto do escalamento da voltaxe no consumo de potencia dinámico está dado polo produto Kf x Kv 2 , xa que o consumo de potencia ten dependencia cuadrática coa voltaxe e lineal coa frecuencia. Para o exemplo anterior, a potencia escalaría por un factor aproximado de x0.2 (80% de redución da potencia dinámica). Os procesadores actuais explotan esta propiedade de xeito masivo. Por un lado, para rebaixar o consumo de potencia nos circuítos que non están nos camiños críticos, a por outra, mediante o escalado dinámico da voltaxe a cada núcleo de procesamento cando non demandan computación intensiva.

Qué es el FO4? Qué mide?

Es el fan-out-of-four: *No canto de expresar os retardos dos circuítos de xeito absoluto, en picosegundos por exemplo, resulta máis interesante expresalo en termos da unidade de retardo FO4 (fanout-of-four). Un FO4 é o retardo que ten un inversor cunha carga de catro inversores. Tense verificado que a expresión do retardo dos circuítos normalizados ao retardo dun FO4 é practicamente independente do nodo tecnolóxico.*

Cuál sería el tamaño del subdirectorio si hablamos de un directorio de vector de bits asignados a grupos? Supón que N é o número de procesadores (cun só core).

(N/TAM_GRUPO)/(Tam_memoria_nodo/Tam_linea_cache)

Explica, en caso de poder darse, en qué situación se producen las dependencias del tipo WAW suponiendo que no hay ejecución fuera de orden.

Cuando una instrucción escriba en un registro antes de que otra instrucción anterior aún no haya escrito su valor, dada la ejecución concurrente.

Cál é a función do estado F no protocólo MESIF?

“” O algoritmo MESIF introduce un novo estado: F (Forward). Na implementación do algitmo MESI convencional nun sistema NUMA con conexións punto a punto, cando se produce un fallo de lectura na cache dun procesador, este retransmite a todos os procesadores a petición de lectura. Os procesadores que teñan na súa cache a liña en estado SH proporcionarán a súa copia (así coma o controlador de memoria correspondente á ubicación da liña en memoria). Isto supón un gasto innecesario de ancho de banda de interconexión, xa que só se require enviar unha copia. Para evitar isto, incorpórase o estado F, que marca a copia que será enviada cando haxa peticións de lectura. Polo tanto, cando haxa varias copias en diferentes procesadores, todas estarán en estado SH excepto unha que estará en estado F. O procesador que ten a copia en estado F será o encargado de enviar copia ao procesador que fai a petición. Por motivos de localidade e eficiencia, o estado F sempre se incorpora a última copia realizada, pasando a anterior do estado F a SH. “”

En cristiano: cuanda haya una petición de lectura, en vez de que todos los procesadores que tengan esa línea caché proporcionen su copia (sería costoso); hay una de las copias marcada como F que indica la que será enviada, esta es la última copia realizada.

De los problemas, el primero que se ve cortado:

- **CPI sin tener en cuenta fallos caché = 1.54**
- **Emite una instrucción por ciclo**
- **18% instrucciones de salto** → 0.18 saltos/instruccion
- **tasa de acierto en la predicción de saltos de 90%** →0.1 fallos/salto
- **Coste del fallo 10 ciclos** → 10 ciclos/fallo
- **Penalizaciones por conflictos estructurales, dependencias y saltos aumentan x1.20**
- **fabricante quere manter o cpi, polo que mellora a predicción de saltos**

Cuál es la tasa de acierto que debe tener el nuevo predictor de saltos para conservar el mismo CPI?

Primero obtenemos el CPI con los fallos antes de modificar el pipeline:

0.18 saltos/instruccion * 0.1 fallos/salto = 0.018 fallos/instruccion
0.018 fallos/instruccion * 10 ciclos/fallo = 0.18 ciclos/instruccion (cpi)

CPI con fallos = 1.54 + 0.18 = 1.72

Modificando el pipeline se aumentan penalizaciones x1.2

1.54 * 1.2 = 1,848 → quitándole los fallos con esa tasa de acierto 1.848 - 0.18 = 1.668

Para mantenerse en 1.72 los fallos deben perjudicar en 1.72 - 1.668 = 0.052 al CPI

0.18 saltos/instruccion * X fallos/salto * 10 ciclos/fallo = 0.052
1-X = tasa de acierto necesaria = 97,11 %

De los problemas, el segundo que se ve cortado (lo resuelve su puta madre):

- **Escalar al siguiente nodo tenolóxico.**
- **Área de caché escala de manera ideal.** → x0.5
- **Área de los núcleos escala x0.65**
- **El área que ocupan los núcleos é igual a los que ocupa la cache de último nivel**
- **Estimar factor de aumento de áreea en el nuevo nodo teniendo en cuenta:**
  - **Frecuencia escala x1.25**
  - **núcleos no alteran la latencia de memoria** → LM es la misma en ambas
  - **Factor de ganancia en [...] con la raíz cuadrada del factor de aumento del número de núcleos**
  - **Ancho de banda con memoria debe escalar como mucho x0.5**

2014_mayo

Un fabricante quere escalar un procesador dun nodo tecnolóxico ao seguinte (escalado lineal dos patróns xeométricos por un factor x0.7) de tal xeito que a área da memoria cache de último nivel escala de forma ideal, mentres que a área dos núcleos escala por un factor x0.65. No nodo actual, a área que ocupan os núcleos é igual a área que ocupa a cache de último nivel. Estimar o factor de aumento de área na implementación no novo nodo tecnolóxico (correspondente á área total de núcleos e cache de último nivel), tendo en conta o seguinte escenario de escalamento: a frecuencia escala por un factor x1.125, os núcleos non alteran a súa arquitectura, a latencia de memoria medida en ns é a mesma, e suponse que o factor de ganancia en velocidade escala coa raíz cadrada do factor de aumento do número de núcleos. O fabricante impón unha restricción no escalamento: o ancho de banda coa memoria debe escalar como moito por un factor x2. Supoñer p=0.5.

Un programa que corre en un determinado procesador presenta un CPI (do núcleo ideal, sen ter en conta fallos cache) de 1.54 ciclos por instrucción. Este procesador emite como máximo unha instrucción por ciclo. O programa presenta un 18% de instruccións de salto e a taxa de acerto dos saltos é do 90% (o custe dun salto mal predecido é de 10 ciclos, e custe adicional dos saltos ben predecidos é cero). Este procesador sufre unha modificación no seu pipeline (máis etapas), de tal xeito que as penalizacións por conflictos estructurais, dependenzas e saltos aumentan por un factor x1.20. O fabricante quere manter o CPI do procesador modificado igual que o procesador orixinal, para o cal modifica a lóxica de predición de saltos. Cal é a taxa de acerto que debe ter o novo preditor de saltos para conservar o mesmo CPI?

Calcula unha cota superior do ancho de banda e capacidade de memoria para unha canle de memoria GDDR5 que ten un sistema de interconexión de 384 bits e que utiliza chips do tipo 4Gbit (x32). Cantas canles DDR3 estándar serían precisas para acadar este ancho de banda, utilizando chips PC3-14928? (1.0p)5- Indicar e explicar o contido (unha das posibles opcións válidas) da táboa de renomeado de rexistros, supoñendo que un núcleo superscalar executa o seguinte código (os operandos inmediatos non se especifican por simplicidade). Indicar e explicar o contido da táboa nun instante no que todas as instruccións están no ROB. Rexistros ISA: EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP; rexistros non ISA: incorporados a cada entrada do ROB, ás cales nos referimos coma slot 1, slot 2, etc, segundo a posición que ocupe a instrucción correspondente no ROB. Supoñer que a primeira instrucción (LD EAX EBP) ocupa o slot 1.

LD EAX EBP
FX-ADD EBX EAX EAX
FX-MUL ESI EAX EBX
FX-DIV EDX ESI EAX
LD EAX ESP
FX-MUL EBX EAX EAX
STORE EBX ESP
FX-SUB ESI ESP EBX

Determinar o CPI do seguinte programa na arquitectura do procesador da Figura 9 to Tema 3 das notas do clase (pipeline con bypass, instruccións de 4 bytes). Descontar os ciclos necesarios até que se enche o pipeline. Supoñer que para os saltos se utiliza a estratexia de supoñer salto non tomado

DADD R2,R1,R5
LD R6 45(R2) // Supoñer que o contido en 45(R2) é o valor 13
DSUB R4, R2, R9 // Supoñer R2=-18 e R9=6
DADD R8, R6,R4
BR R8 16 // Salto ao enderezo PC+20
DADD R3 R6 R7
ST R1, 400(R3)
ST R5, 404(R3)
BR R0 12 // Supoñer R0=0
DADD R6 R2,R4
DSUB R7, R2,R8
NOP // fin do programa, non operación.