Arquitectura de Computadoras

Solución segundo parcial - 05 Abril 2019

1. Una computadora tiene una memoria principal de 2 GBytes direccionable por Bytes. La cache es de 512 KB con bloques de 32 Bytes. Llene la siguiente tabla:

   Direcciones físicas: $log_2(2G) = 31$. *Bits para bloque* $log_2(32) = 5$.
   La memoria cache tiene $512kB/32B = 16K$ *bloques.*

	Tamaño Tag	Tamaño Index	Líneas de cache a las que puede mapearse bloque de memoria
Mapeo directo	$31 - (14 + 5) = 12$	$log_2(16K) = 14$	1
Full associative	$31 - 5 = 26$	$--$	2^{14}= *16 K*
8-way set associative	$31 - (11 + 5) = 15$	$14 - log_2(8) = 11$	8

2. ¿Por qué es importante tener muchos registros (≥ 32) en las arquitecturas RISC?

   Porque las arquitecturas RISC no tienen muchos modos de direccionamiento a memoria; la mayoría de las variables deben accederse de manera sencilla, preferentemente con direccionamiento a registros (sobre todo para las operaciones aritmético-lógicas.
   Porque la longitud de las palabras de instrucción deben ser cortas y de longitud fija (en su mayoría). Esto se consigue si los operandos fuente y destino están en registros (Arq. de tres direcciones, direccionamiento de registro).
   Porque se requiere de operaciones más o menos regulares de manera que las instrucciones puedan insertarse en el pipeline. Las operaciones son más sencillas y cortas si sus operandos son registros.

3. Mencione brevemente por qué las arquitecturas VLIW no tuvieron éxito?

   Afectan mucho la dinámica del procesador pipeline No se emiten de manera regular en las distintas unidades funcionales.
   Son muy dependientes del hardware: Es difícil mantener compatibilidad con la evolución tecnológica o con otros procesadores que tengan la misma "arquitectura".
   Dependen enormemente del compilador; un compilador no personalizado difícilmente puede generar código que explote al máximo la existencia de varias unidades funcionales.
   Prácticamente no se puede garantizar la autonomía entre la generación de compilador y su efecto en el desempeño del procesador.
   -

4. ¿Qué hace el siguiente código?

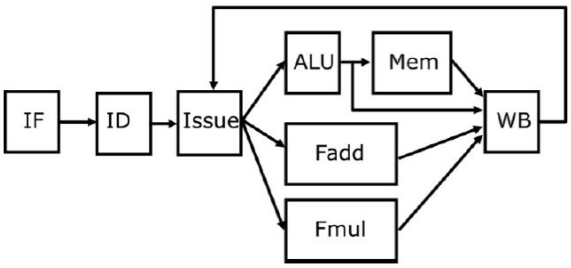
   VL = 01010101
   VM = 01010101
   V3 = V2+V1

Supone que los registros vectoriales son de 85 elementos. Se hace la suma de V2 con V1 pero solamente algunos de sus elementos (los elementos 1, 3, 5, 7, ...) son sumados.

5. Considere un procesador con repertorio de instrucciones (ISA) con extensiones para multimedia (MMX). ¿Cuál es la longitud de palabra para instrucciones regulares si es capaz de realizar cuatro operaciones aritméticas simultáneamente sobre registros de dos bytes?

   *4*2 = 8 bytes = 64 bits*

6. Considere un procesador In-order issue, Out-of-order completition como el que se muestra en la figura. Para simplificar el problema, en el diagrama de tiempos que se solicita, sólo consideraremos las etapas I - Cuando la instrucción se emite; E - cuando la instrucción inicia ejecución en una FU; C - cuando la instrucción se completa y entra a la etapa WB para escribir los resultados.



- Todas las FU tienen pipeline
- Operaciones ALU toman un ciclo; Operaciones Mem dos más (tres en total)
- Fadd toma tres ciclos; Fmul cinco
- No hay renombramiento de registros
- Una instrucción puede ser emitida si no genera WAR o WAW
- Sólo se puede emitir una instrucción a la vez. Si hay varias candidatas, se emite la más antigua

Llene la siguiente tabla. Para su ayuda, se han llenado ya los primeros dos renglons

Instr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
LD F2, 0(R1)	I	E	E	E	C												
ADD F1, F1,F2		I			E	E	E	C									
MUL F4,F3,F3			I	*E*	*E*	*E*	*E*	*E*	*C*								
ADDI R1, R1,8				*I*	*E*	*C*											
LD F2, 0(R1)					*I*	*E*	*E*	*E*	*C*								
ADD F1,F1,F4						*I*	*E*	*E*	*E*	*C*							

7. Para cada una de las afirmaciones, indique si es cierta o falsa

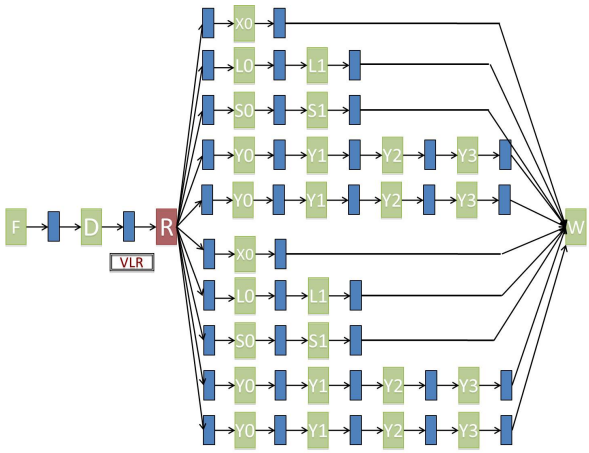
   CIERTO El renombramiento de registros elimina bloqueos por dependencias WAW

   FALSO El renombramiento de registros elimina bloqueos por dependencias RAW

8. El procesador de la siguiente figura tiene dos lanes; cada una con dos FU de punto flotante (Y) una para carga (LD), una para guardar (S) y una FU escalar (X). Puede hacer chaining.

¿En cuánto tiempo se ejecuta el siguiente código suponiendo que los registros vectoriales son de 32 elementos? (Puede ignorar los detalles, únicamente considere cuántas instrucciones pueden estar ejecutándose simultáneamente)

```
LV V1, 0(R1)
MULV V2, V1, R2
ADDV V3, V1, R3
```



En 16 ciclos. Se pueden hacer seis operaciones simultáneas: Dos de Ld, dos de Mul y dos de Add. Sin varios lanes ni varias unidades funcionales, tomparía 32(3) = 96 ciclos.

9. Para los siguientes segmentos de código indique qué característica arquitectónica (Superescalar, Predicción de Saltos, Especulativa con renombramiento de registros) mejora el desempeño. Justifique BREVEMENTE su respuesta

Código	Justificación

ADD F0,F1,F8
ADD F2,F3,F8
ADD F4,F5,F8
ADD F6,F7,F8
	uperescalar: No hay dependencias ni saltos

loop: ADD R3, R4,R0 LD R4, 8(R4) ; cache hit BNEQZ R4, LOOP	Predicción de saltos. Hay muchos conflictos WAR, RAW y muchos saltos Renaming resuelve WAR pero no RAW

LD R1, 0(R2) ;cache miss
ADD R2,R1,R1
LD R1, 0(R3) ; cache hit
LD R3, 0(R4) ; cache hit
ADD R3,R1,R3
ADD R1,R2,R3
	Especulativa con renombramiento.
	Permite que I3 e I4 se ejecuten esperando el miss.
	Hay demasiados conflictos de datos para que se pueda
	aprovechar Arq. superescalar

10. Seleccione las respuestas correctas

- Which of the following statements are true with regard to compute capability in CUDA

___ Code compiled for hardware of one compute capability will not need to be re-compiled to run on hardware of another

X Different compute capabilities may imply a different amount of local memory per thread

___ Compute capability is measured by the number of FLOPS a GPU accelerator can compute

- Which of the following correctly describes a GPU kernel

___ A kernel may contain a mix of host and GPU code

X All thread blocks involved in the same computation use the same kernel

___ A kernel is part of the GPU's internal micro-operating system, allowing it to act as in independent host

- What strategy does the GPU employ if the threads within a warp diverge in their execution?

___ Threads are moved to different warps so that divergence does not occur within a single warp

___ Threads are not allowed to diverge

X All possible execution paths are run by all threads in a warp serially so that thread instructions do not diverge

- Shared memory in CUDA is accessible to:

X All threads in a single block

___ Both the host and GPU

___ All threads associated with a single kernel

- Which of the following correctly describes the relationship between Warps, thread blocks, and CUDA cores?

___ A warp is divided into a number of thread blocks, and each thread block executes on a single CUDA core

X A thread block may be divided into a number of warps, and each warp may execute on a single CUDA core

___ A thread block is assigned to a warp, and each thread in the warp is executed on a separate CUDA core

11. ¿Para qué se utilizan las siguientes variables en CUDA?

gridDim . Define las dimensiones del grid en bloques. Tiene dos campos: La dimensión en X y la dimensión en Y.

blockIdx Es el identificador del bloque dentro del grid

blockDim Define las dimensiones de un bloque en threads. Tiene tres campos: La dimensión en X, la dimensión en Y y la dimensión en Z

threadIdx Es el identificador del hilo dentro del bloque