UCSP_black.pdf

Examen Final de Arquitectura de Computadores 2018-02

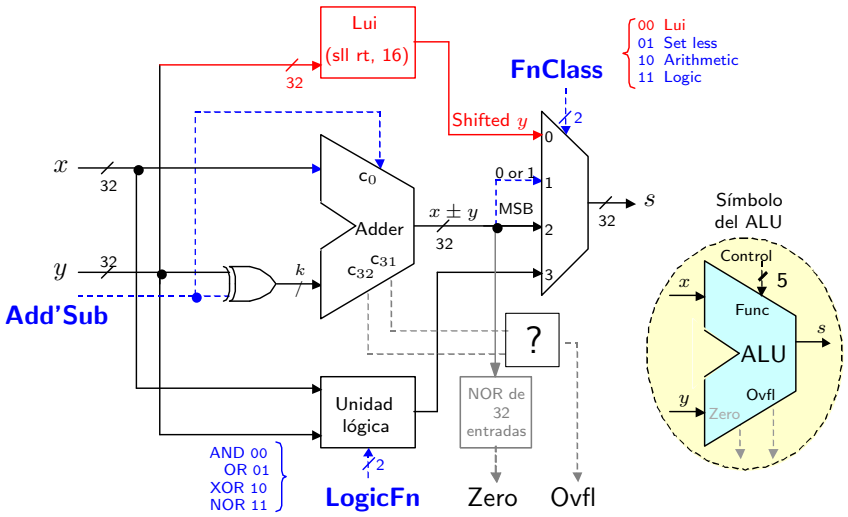
Escuela Profesional de Ciencia de la Computación: CCOMP3-1, CCOMP3-2

04 diciembre 2018

ARCHITECTURE IN SINGLE CYCLE

Pregunta 1: **[4pt, Habilidades: b2]**

Sea la unidad aritmético-lógica de la figura, la cual ha sido adecuada para implementar la instrucción [`lui rt, imm`] en vez de las instrucciones de corrimiento estudiadas..



Haga el diseño lógico (dibujo de puertas lógicas) faltante de:

a) El bloque que implementa la instrucción [`lui rt, imm`]

b) El bloque que implementa las instrucciones lógicas [`and`, `or`, `xor`, `nor`]

ARCHITECTURE IN MULTICYCLE

Pregunta 2: **[4pt, Habilidades: b2]**

Dados los siguientes porcentajes y cantidades de ciclos de ejecución de instrucciones MIPS en arquitectura multiciclo:

Aritméticas	50%	4 ciclos
Lectura en memoria	23%	5 ciclos
Guardar en memoria	10%	4 ciclos
Bifurcaciones	15%	3 ciclos (*)
Saltos incondicionales	2%	3 ciclos (*)

a) Calcule el porcentaje de reducción del CPI (Ciclos de reloj por instrucción) que se obtendría si la arquitectura multiciclo deja de usar el ALU para resolver bifurcaciones y saltos incondicionales (lo cual toma 3 ciclos de reloj) y pasa a utilizar la lógica de próxima dirección existente en la arquitectura de ciclo único, la cual promete resolver todo en el 2do ciclo de reloj.

b) Calcule la cantidad de instrucciones por segundo que se ejecutarían, antes y después de cambiar la lógica de cálculo de bifurcaciones y saltos incondicionales, si el reloj del procesador es de 2.5GHz

ARCHITECTURE IN PIPELINING

Pregunta 3: **[4pt, Habilidades: b2,i3]**

Sea el siguiente trecho de código microMIPS que busca el máximo valor en una lista A, a ser ejecutado en una arquitectura en *pipeline* de 5 etapas:

```
      lw $t0,0($s1)       # initialize maximum to A[0]
      addi $t1,$zero,0    # initialize index i to 0
loop: add $t1,$t1,1       # increment index i by 1
      beq $t1,$s2,done    # if all elements examined, quit
      add $t2,$t1,$t1     # compute 2i in $t2 (**)
      add $t2,$t2,$t2     # compute 4i in $t2 (**)
      add $t2,$t2,$s1     # form address of A[i] in $t2
      lw $t3,0($t2)       # load value of A[i] into $t3
      slt $t4,$t0,$t3     # maximum < A[i]?
      beq $t4,$zero,loop  # if not, repeat with no change
      addi $t0,$t3,0      # if so, A[i] is the new maximum
      j loop              # change completed; now repeat
done: ...                 # continuation of the program
```

a) Identifique las instrucciones donde existe dependencia de datos tipo I y II, y dependencias de control

b) Si el *pipeline* soporta *data forwarding*, reordenamiento de instrucciones y previsión de bifurcación simple:

- ¿Es posible reordenar instrucciones en este código?
- ¿En qué instrucciones se insertarán efectivamente burbujas?
- ¿Cuántos ciclos de reloj se emplearán en total, si la lista de datos a buscar es `A=4,6,1`?

c) **(bonificación extra)** Dé alguna sugerencia de cambio para que las instrucciones marcadas como (**) se ejecuten en menos ciclos de reloj

MEMORY SYSTEMS

Pregunta 4: **[3pt, Habilidades: b2,i3]**

Sea un procesador con 3 niveles de caché: $L1, L2, L3$ cuyo CPI sin fallo de caché es 1.0 y en cada instrucción se asume 1.12 accesos a memeoria en promedio, y los parámetros de caché son:

- $L1$ con acierto $h_1 = 90\%$, penalización por falla 9 ciclos de reloj
- $L2$ con acierto $h_2 = 87\%$, penalización por falla 15 ciclos de reloj
- $L3$ con acierto $h_3 = 75\%$, penalización por falla 55 ciclos de reloj

Calcular el CPI efectivo considerando los fallos de caché y hallar la tasa de acierto si se ve el sistema de caché $L1 + L2 + L3$ como un caché de un único nivel.

I/O SYSTEMS

Pregunta 5: **[3pt, Habilidades: b2,i3]**

Sea un procesador de 2GHz, verificar el porcentaje de tiempo perdido atendiendo los llamados de un mouse de bolita durante el movimiento, si la consulta más el costo del llamado a interrupción cuestan 1000 ciclos de reloj, y cuando se mueve la bolita del *mouse*, éste dispara 100 interrupciones por segundo.

ALL TOPICS

Pregunta 6: **[2pt, Habilidades: b2,i3]**

Responda con verdadero o falso a las siguientes afirmaciones:

a) En el ALU estudiado, la función que determina la ocurrencia de *overflow* es la siguiente: $ov = c_{31} \wedge c_{32}$. [ ]

b) Al analizar el cumplimiento de la condición de diversas bifurcaciones, se determina que las bifurcaciones cuyo salto es "hacia atrás" ocurren con más frecuencia que aquellas cuyo salto ocurre "hacia adelante" [ ]

c) Las instrucciones extendidas MMX permiten sumas, multiplicaciones y bifurcaciones con vectores de 64bits que contienen varios operandos enteros [ ]

d) Es estrictamente necesario que un sistema QPU esté en el cero absoluto ($0^{\circ}K$) para que funcionen la propiedades de la mecánica cuántica y funcione el sistema [ ]