Facultad de Informática
Universidad Complutense de Madrid

PROBLEMAS DE FUNDAMENTOS DE COMPUTADORES II

TEMA 5

Problemas básicos:

1. Suponiendo que las instrucciones al ejecutar un programa en un RISC-V monociclo se reparten de la siguiente manera:

	tipo **`add`**	tipo **`addi`**	**`lw`**	**`sw`**	**`beq`**	**`jal`**
frecuencia	24%	28%	25%	10%	11%	2%

Indique:

a) El porcentaje de instrucciones que usan la memoria de datos.
b) El porcentaje de instrucciones que usan la memoria de instrucciones.
c) El porcentaje de instrucciones que usan el valor calculado por el extensor de signo.
d) El porcentaje de instrucciones que usan el valor calculado por la ALU.

2. En el procesador monociclo, justifique los valores que toman las señales de control al ejecutar instrucciones **`lw`**.

3. En el procesador monociclo, justifique los valores que toman las señales de control al ejecutar instrucciones **`andi`**.

4. En el procesador monociclo, justifique los valores que toman las señales de control al ejecutar instrucciones **`beq`**.

5. Suponga que el procesador monociclo lee de la dirección 0x00001000 de la memoria de instrucciones la palabra 0x00c6aa23, que todo registro xi del procesador contiene el valor xi (i.e. el registro x1 contiene 0x0000001, el registro x2 contiene 0x00000002, etc.) y que en todas las direcciones de la memoria de datos hay almacenado 0x00000000. Indique:

a) El valor que toma la señal ALUctr generada por el DEC ALU.
b) El valor que toma la señal ImmSrc generada por el DEC sExt.
c) El nuevo valor del PC tras ejecutar la esta instrucción.
d) Los valores que tienen en las entradas y salidas de datos los multiplexores.
e) Los valores en las salidas del banco de registros.
f) Los valores a las entradas de la ALU y los sumadores.

Problemas adicionales:

6. Indique qué instrucción está ejecutando el procesador monociclo sabiendo los valores que toman las señales de control:

a) MemWr = 1
b) ALUSrc = 0 y ALUctr = 010
c) BRwr = 0 y ALUctr = 001
d) PCsrc = 1 y Jump = 1
e) ALUSrc = 1 y ALUctr = 000
f) ResSrc = 10
g) ResSrc = 00
h) Branch = 1

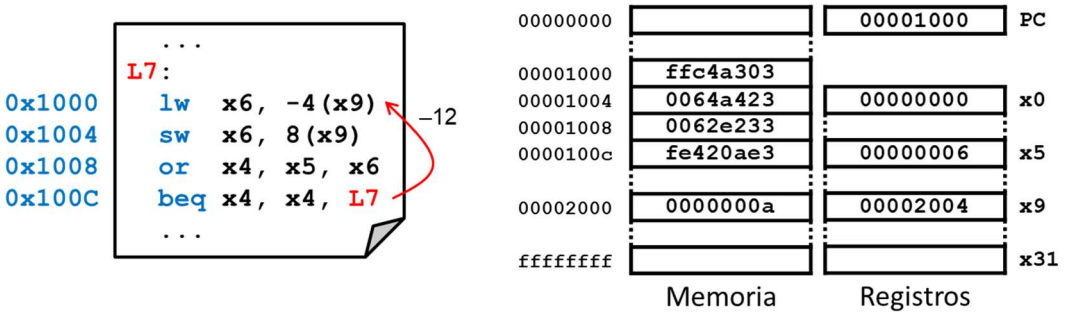
7. Un pequeño porcentaje de los chips que se fabrican son defectuosos por errores de manufactura. Es común que los defectos hagan que el circuito se comporte como si una de sus interconexiones estuviera permanentemente a valor 0 (*stuck-at-0*) o a valor 1 (*stuck-at-1*). Indique qué instrucciones fallarían en el RISC-V monociclo de arquitectura

reducida si una de las siguientes señales de control tuviera un defecto de tipo *stuck-at-0*.

a) BRWr
b) $ALUop_1$
c) $ALUop_0$
d) MemWr
e) $ImmSrc_1$
f) $ImmSrc_0$
g) $ResSrc_1$
h) $ResSrc_0$
i) PCSrc
j) ALUSrc

8. Repetir el ejercicio anterior en el caso de que el error fuera de tipo *stuck-at-1*.

9. Supongamos que sobre el procesador monociclo se quiere ejecutar el programa que se muestra a continuación, siendo el contenido inicial de los registros y la memoria el mostrado en la figura. Represente los diagramas de ejecución para los registros y la memoria, así como para las señales de estado y de control, de modo que se visualicen sus valores en cada ciclo de reloj correspondiente a la ejecución del programa.



10. Discuta las modificaciones que sería necesario realizar en la ruta de datos del RISC-V monociclo de arquitectura reducida para ampliar su repertorio con cada una de las instrucciones siguientes:

a) **`xor` / `xori`**
b) **`sll` / `slli`**
c) **`bne` / `blt` / `bge`**
d) **`slt` / `slti`**
e) **`lb` / `lh` / `lbu` / `lhu`**
f) **`sb` / `sh`**
g) **`jarl`**
h) **`lui`**
i) **`auipc`**

11. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador monociclo para añadir una nueva instrucción de tipo-I que lea una palabra de memoria pre-incrementando el registro base, **`lwpreinc` *`rd, imm(rs1)`***:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{rs1} + \text{sExt(imm)}\ ],\ \text{rs1} \leftarrow \text{rs1} + \text{sExt(imm)}\ \}$$

12. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador monociclo para añadir una nueva instrucción de tipo-I que lea una palabra de memoria post-incrementando el registro base, **`lwpostinc` *`rd, imm(rs1)`***:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{rs1}\ ],\ \text{rs1} \leftarrow \text{rs1} + \text{sExt(imm)}\ \}$$

13. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador monociclo para añadir una nueva instrucción de tipo-R que intercambie el valor contenido en 2 registros, **`swap` *`rs1, rs2`***:

$$\{ \text{rs1} \leftarrow \text{rs2},\ \text{rs2} \leftarrow \text{rs1}\ \}$$

14. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador monociclo para añadir una nueva instrucción de tipo-U que cargue en un registro un dato leído de memoria que ocupa una dirección dada, **`lwa` *`rd, imm`***:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{zExt(imm)}\ ]\ \}$$

15. Si pudiera reducir a la mitad el retardo de solo uno los módulos del procesador

monociclo diseñado con la biblioteca CMOS de 90 nm, ¿cuál elegiría para obtener un tiempo de ciclo menor? ¿cuál sería el nuevo tiempo de ciclo?

16. Si se redujera un 30% el retardo de la ALU, un 10% el retardo del extensor de signo y 20% los retardos de lectura/escritura del banco de registros del procesador monociclo diseñado con la biblioteca CMOS de 90 nm, calcule el retardo del camino crítico de cada tipo de instrucción y cuánto tardaría este procesador en ejecutar 100 millones de instrucciones.

17. Suponga que añadir la operación de multiplicación a la ALU del procesador monociclo duplica su retardo, pero reduce el número de instrucciones ejecutadas un 5% (debido a que no hay que emularlas por software). Se pide:

a) Calcular a frecuencia de reloj del nuevo procesador.
b) Discutir si tiene sentido la mejora.
c) Calcular el incremento de retardo de la ALU máximo a partir del cual no tiene sentido el cambio.
d) El porcentaje de reducción de instrucciones mínimo requerido para que tenga sentido el cambio.

18. En el procesador monociclo, la instrucción **`lw`** es la más lenta porque hace uso simultáneo de todos los recursos de la ruta de datos. Si el comportamiento de las instrucciones **`lw`** y **`sw`** se modifica de manera que la dirección que envían a memoria sea la almacenada en el registro base rs1 (y no su suma con el desplazamiento inmediato), se puede reducir el tiempo de ciclo. Sin embargo, esto obliga a reemplazar en todos los programas las antiguas **`lw`** y **`sw`** por pares **`lw`**/**`addi`** y **`sw`**/**`addi`**. Suponiendo que el procesador original ejecuta un programa de 10^8 instrucciones repartidas de la siguiente manera:

	tipo **`add`**	tipo **`addi`**	**`lw`**	**`sw`**	**`beq`**	**`jal`**
frecuencia	24%	28%	25%	10%	11%	2%

Se pide:

a) Calcular el número total de instrucciones que tendrá el nuevo programa.
b) Calcular el tiempo de ciclo del nuevo procesador. I
c) Calcular la relación entre los tiempos de ejecución del programa en ambos procesadores (speedup).
d) Discutir si tiene sentido la mejora.

19. Suponga que doblando el número de registros de propósito general de 32 a 64 el procesador monociclo reduce el número de instrucciones **`lw`** y **`sw`** ejecutadas un 12% pero a costa de aumentar el retardo de acceso al banco de registros un 25% y su coste un 50%. Suponiendo que el procesador original ejecuta un programa de 10^8 instrucciones repartidas de la siguiente manera:

	tipo **`add`**	tipo **`addi`**	**`lw`**	**`sw`**	**`beq`**	**`jal`**
frecuencia	24%	28%	25%	10%	11%	2%

Se pide:

a) Calcular el número total de instrucciones que tendrá el nuevo programa.
b) Calcular el coste y tiempo de ciclo del nuevo procesador.
c) Calcular la relación entre los tiempos de ejecución en ambos procesadores.

Soluciones

1. a) 35% b) 100% c) 76% d) 98%

2. Branch = 0, Jump = 0, BRwr = 1, ALUsrc = 1, ALUop = 00, MemWr = 0
ResSrc = 00, ALUctr = 000, ImmSrc = 11

3. Branch = 0, Jump = 0, BRwr = 1, ALUsrc = 1, ALUop = 10, MemWr = 0
ResSrc = 01, ALUctr = 010, ImmSrc = 00

4. Branch = 1, Jump = 0, BRwr = 0, ALUsrc = 0, ALUop = 01, MemWr = 0
ResSrc = −, ALUctr = 001, ImmSrc = 10

5. a) ALUctr = 000 b) ImmSrc = 01 c) PC = 0x1004
d) Mux-PC: E_0 = 0x00001004, E_1 = 0x00001014, S = 0x00001004
Mux-ALU: E_0 = 0x0000000d, E_1 = 0x00000014, S = 0x00000014
Mux-WB: E_0 = 0x00000000, E_1 = 0x00000021, E_2 = 0x00001004, S = −
e) RD1 = 0x0000000d, RD2 = 0x0000000c
f) ALU_{up} = 0x0000000d, ALU_{down} = 0x00000014
Add_{PC} = 0x00001000
Add_{up} = 0x00001004, Add_{down} = 0x00001000

6.
a) **`sw`**
b) **`and`**
c) **`beq`**
d) **`jal`**
e) **`addi`**
f) **`jal`**
g) **`lw`**
h) **`beq`**

7.
a) **`lw`**, tipos R/I, **`jal`**
b) tipos R/I
c) **`beq`**
d) **`sw`**
e) **`jal`**, **`beq`**
f) **`sw`**, **`jal`**
g) **`jal`**
h) tipos R/I
i) **`jal`**, **`beq`**
j) **`lw`**, **`sw`**, tipo I

8.
a) **`sw`**, **`beq`**
b) **`lw`**, **`sw`**, **`beq`**
c) **`lw`**, **`sw`**, tipos R/I
d) **`lw`**, **`beq`**, **`jal`**, tipos R/I
e) **`lw`**, **`sw`**, tipo I
f) **`lw`**, **`beq`**, tipo I
g) **`lw`**, tipos R/I
h) **`lw`**, **`jal`**
i) **`lw`**, **`sw`**, tipos R/I
j) **`beq`**, tipo R

9. Véanse las transparencias.

10. a) Ampliar la funcionalidad de la ALU.
b) Ampliar la funcionalidad de la ALU.
c) Ampliar la funcionalidad de la ALU.
d) Ampliar la funcionalidad de la ALU.
e) Añadir a la Memoria de Datos las líneas de selección de anchura de acceso.
Añadir a la salida de la Memoria de Datos un extensor de signo/ceros.
f) Añadir a la Memoria de Datos las líneas de selección de anchura de acceso.
g) Conectar la salida de la ALU a una nueva entrada del MUX-PC.
h) Ampliar la funcionalidad del Extensor de Signo.
Conectar la salida del Extensor de Signo a una nueva entrada del MUX-WB.

i) Ampliar la funcionalidad del Extensor de Signo.

11. Añadir al BR otra entrada de datos de escritura y conectarla a la salida de la ALU.
Añadir al BR otra entrada de dirección de escritura y conectarla los bits 19:15 de la salida de la Memoria de Instrucciones.
Añadir al BR otra entrada de control de escritura.

12. Añadir al BR otra entrada de datos de escritura y conectarla a la salida de la ALU.
Añadir al BR otra entrada de dirección de escritura y conectarla los bits 19:15 de la salida de la Memoria de Instrucciones.
Añadir al BR otra entrada de control de escritura.
Añadir un MUX a la entrada de dirección de la Memoria de Datos y conectar sus entradas a la salida de la ALU y a la salida RD1 del BR.

13. Conectar la salida RD2 del BR a una nueva entrada del MUX-WB.
Conectar en la entrada WA un nuevo MUX con 2 entradas conectadas a los bits 11:7 y 19:15 de la salida de la Memoria de Instrucciones.
Añadir al BR otra entrada de datos de escritura y conectarla a la salida RD1 del BR.
Añadir al BR otra entrada de dirección de escritura a los bits 24:20 de la salida de la Memoria de Instrucciones.
Añadir al BR otra entrada de control de escritura.

14. Ampliar el Extensor de Signo.
Añadir un MUX a la entrada de dirección de la Memoria de Datos y conectar sus entradas a la salida de la ALU y a la salida del Extensor de Signo.

15. Memoria de instrucciones, 23366 ps

16. **`lw`**: 24921 ps **`sw`**: 24061 ps tipo-I: 16421 ps tipo-R: 16134 ps
`beq`: 16421 ps **`jal`**: 9932 ps t_{clk} = 24921 ps t_{ejec} = 2,49 s

17. a) 27,8 MHz b) no c) 1453 ps d) 23,2%

18. a) $135 \cdot 10^6$ b) 19116 ps c) 1,07 d) si

19. a) $95{,}8 \cdot 10^6$ b) 27792 ps, 84883 μm^2 c) 1,04