

## PROBLEMAS DE UNIDAD DE CONTROL

1. Deseamos añadir la instrucción **j** a la ruta de datos de un sólo ciclo. Añadir los elementos de la ruta de datos y puntos de control necesarios para realizar esta instrucción. El código de operación de la instrucción **j** es: 000010.

j dirección /\* PC=dirección, los 4 bits más significativos son los del PC \*/

¿Cómo habría que modificar la unidad de control?

2. Realizar el problema anterior para la ruta multiciclo.

Mostrar los añadidos a la máquina de estados finitos con 10 estados que se implementó en clase para realizar esta operación.

3. Deseamos añadir la instrucción **jal** a la ruta de datos de un sólo ciclo. Añadir los elementos de la ruta de datos y puntos de control necesarios para realizar esta instrucción. El código de operación de la instrucción **jal** es: 000011.

4.

jal dirección /\* (\$31= PC+4; PC = dirección) \*/

¿Cómo habría que modificar la unidad de control?

5. Realizar el problema anterior para la ruta multiciclo.

Mostrar los añadidos a la máquina de estados finitos con 10 estados que se implementó en clase para realizar esta operación.

6. Añadir los elementos de la ruta de datos y puntos de control necesarios para realizar la instrucción **addiu**, tanto con ruta de datos monociclo como multiciclo. El código de operación de la instrucción **j** es: 001000.

addiu \$i, \$j, inmediato /\* \$i = \$j + inmediato \*/

7. ¿Qué modificaciones hay que realizar en la ruta de datos y la U.C. microprogramada multiciclo para añadir la instrucción **bne** (bifurcar si no igual, con código de operación 000101) que se comporta de manera similar a **beq**.

8. Suponer que los tiempos de operación para las principales unidades funcionales de una implementación monociclo del MIPS son:

Unidades de memoria: 10ns

ALU y sumadores: 10ns

Banco de registros (lectura o escritura): 5ns

Suponiendo que los multiplexores, unidad de control, accesos al CP, unidad de extensión de signo y el cableado no tengan retardo, ¿cuál de las siguientes implementaciones será más rápida y cuánto más?

1. Una implementación en la que cada instrucción opera en un ciclo de reloj de duración fija.

2. Una implementación donde cada instrucción se ejecuta en un ciclo de reloj utilizando un reloj de duración variable, que para cada instrucción tiene la duración que necesita.

3. Una implementación multiciclo

Frecuencia de uso de las diferentes instrucciones para un caso típico (gcc):

? Carga: 22%

? Almacenamiento: 11%

? Aritmético-lógicas: 49%

? Salto condicional, realizando el salto: 4%

? Salto condicional, no realizando el salto: 12%

? Salto incondicional: 2%

9. Los diseñadores de MIPS han comprobado que el camino crítico que determina el ciclo de reloj de las nuevas máquinas en la implementación multiciclo está dado por las operaciones de acceso a memoria en la parte de lectura o escritura de datos. Ello ha determinado que el ciclo de reloj de la nueva máquina sea de 500 MHz frente a los 750 Mhz esperados.

Pero se puede comprobar que se podría usar el reloj de 750 MHz si cada ciclo de acceso a memoria se divide en dos.

Usando las frecuencias de uso de cada tipo de instrucción que aparecen en el problema anterior, calcular cuánto más rápida es esta nueva implementación.

**10.** Suponiendo los siguientes tiempos de ejecución de los diversos elementos de una ruta de datos para el MIPS:

Lectura de memoria: 5ns; Escritura de memoria: 7ns; Lógica de control de la ALU: 2ns; Lógica de control principal: 2 ns; Lectura BR: 1ns; Escritura BR: 2ns; Operación ALU: 5ns; Extensión signo: 1 ns; Multiplicación por 4: 1 ns; Multiplexores: 1 ns; Tsetup=0,1 ns; Thold=0,1 ns; CLK\_to\_Q= 0,1 ns.

a) Calcular cuál es el mejor tiempo de ciclo para una implementación multiciclo.

b) Si el tiempo de escritura en memoria fuese 15 ns, realizar el mismo cálculo. Pensar algún método de mejorar el rendimiento del sistema en este caso, en el que hay una descompensación clara entre unos ciclos y otros.

**11.** Sobre la estructura de la figura 1 adjunta, que permite implementar el control multiciclo de nuestro procesador, indicar los valores de todas las líneas de datos y los registros, en los siguientes casos:

**a.-** Ejecución de la instrucción OR \$1, \$2, \$3, al final de cada una de sus cuatro etapas.

Estado actual de la máquina:

\$1=HEX(00000017)

\$2=HEX(001100FF)

\$3=HEX(FF000345)

PC=HEX(00003400)

**b.-** Ejecución de la instrucción JAL 1024 (1024 en decimal) al final de las tres etapas.

**12.** Dibujar los cronogramas de ejecución de las operaciones sw y beq, usando los datos de tiempo del problema anterior, para una implementación monociclo.

**13.** Dibujar los cronogramas de ejecución de las instrucciones lw, sw, beq, add, jal en una implementación multiciclo con buses como la vista en clase.

**14.** Diseñar la U.C. completa con lógica discreta para las instrucciones lw, sw, beq, add, jal en una implementación multiciclo con buses como la vista en clase.

Diseñar la U.C. microprogramada con formato horizontal, vertical y vertical por campos.

**15.** Diseñar una ruta de datos y una unidad de control cableada para implementar el siguiente algoritmo que calcula el máximo común divisor de 2 números. Los números llegan por un bus de entrada y el resultado se manda a un bus de salida:

Program Mcd;

/\* Num1 y Num2 son los dos números de los que se calcula el mcd.

Resul almacena los resultados parciales y el final \*/

BEGIN

Leer Num1;

Leer Num2;

Result = 0;

IF Num1 <> 0 AND Num2 <> 0 THEN BEGIN

WHILE Num1 <> Num2 DO

IF Num1 > Num2 THEN Num1 := Num1 - Num2;

ELSE Num2 := Num2 - Num1;

Result := Num1;

END;

END.

**16.** Sobre la ruta de datos de la figura.

a) Dar la secuencia de pasos para implementar las siguientes instrucciones máquina de un computador, indicando sobre la figura qué puntos de control se necesitarían y explicando su significado.

LOAD X (AC<--M(X))

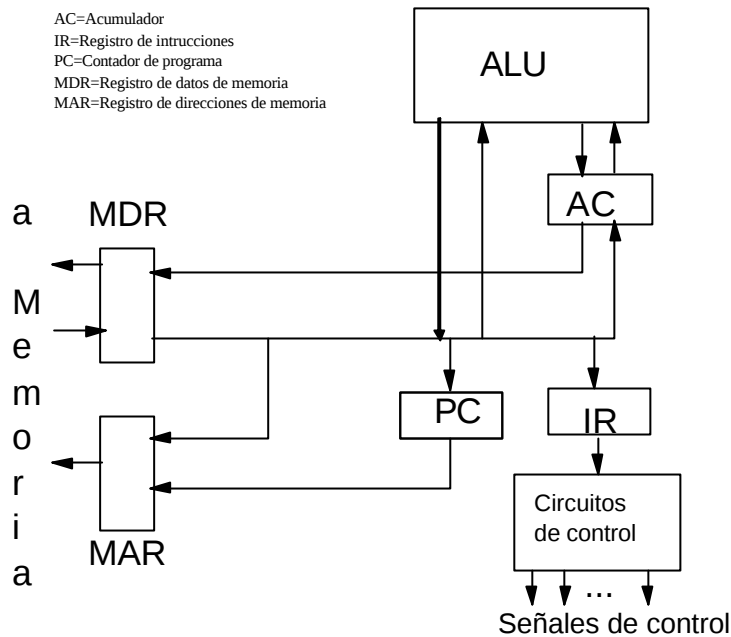
ADD X (AC<--AC+M(X))

SUB X (AC<--AC-M(X))

JUMP X (PC <--X) Salto incondicional

Suponer que la instrucción tiene un campo con el valor de X, que puede pasarse directamente a PC y a MAR.

b) Implementar, a nivel de puertas y los módulos necesarios, la unidad de control cableada para realizar estas operaciones.



17. El siguiente fragmento de código se ejecuta en un MIPS con segmentación:

```
sub    $1,$2,$3
add    $4,$5,$6
sub    $5,$4,$8
add    $7,$2,$3
add    $9,$7,$3
lw     $1,10($6)
add    $3,$1,$4
sub    $6,$7,$8
```

Suponiendo que un dato se puede escribir en un banco de registros y leer su nuevo valor en el mismo ciclo:

- Calcular el número de ciclos necesarios para ejecutar este código si cada vez que existe un conflicto se detiene el pipeline.
- Tratar de reordenar el código para conseguir que el número de ciclos de ejecución sea mínimo.

18. Sobre la estructura del computador MIPS segmentado, se ejecuta la siguiente secuencia de instrucciones:

```
ADD $1, $2, $3
SUB $4, $2, $3
AND $5, $2, $3
OR  $6, $2, $3
```

Si la instrucción ADD está colocada en la dirección de memoria 002000 (Hex), y el contenido de los registros es:

```
$1=000005
$2=000004
$3=000002
PC=002000
```

- Señalar el contenido de los siguientes registros al cabo de 4 ciclos de reloj: pc1, pc2, pc3, a, b, out1, out2, rd1, rd2, rd3, SMDR.
- Indicar también el contenido de: WB1, M1, EX1, WB2, M2, WB3, explicando brevemente su significado.

19. Tratar de averiguar lo máximo posible sobre las instrucciones que se están ejecutando en la ruta de datos de la figura 2 adjunta.

20. Idear algún método para reducir al máximo el tiempo de detención del pipeline en los saltos.