

Ejercicios Tema 4

1. En la ruta de datos monociclo: Suponer que se ha producido un fallo y todas las señales de control a los multiplexores se encuentran desactivadas. ¿Qué instrucciones se ejecutarían correctamente?

2. Si el fallo que se produce es fiar a 1 las señales de control a los multiplexores. ¿Qué instrucciones se podrían ejecutar correctamente?

3. Construye una ruta de datos monociclo que ejecute únicamente la instrucción lw y la sw.

4. En una implementación monociclo, ¿Qué valor deberán tener las señales de control para que se ejecute la instrucción 0x8D0B0008?

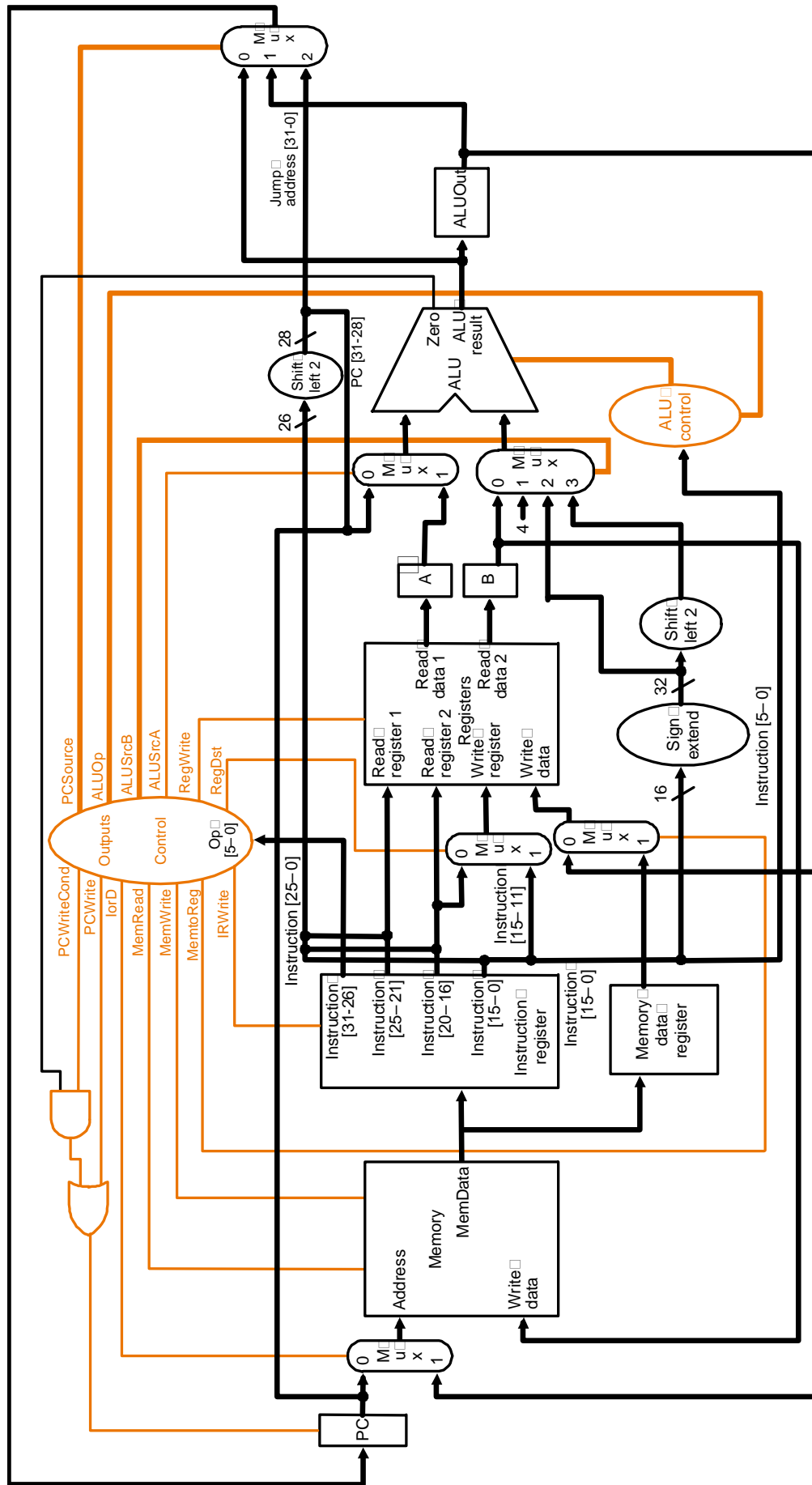
5. Queremos que la ruta de datos monociclo estudiada ejecute una nueva instrucción: addi (suma inmediata). Añade los elementos y señales de control necesarias a la ruta de datos de la transparencia 31 para que se pueda ejecutar la instrucción. ¿Cuál es el valor de las señales de control?

6. ¿Cuántos ciclos de reloj habrán transcurrido tras la ejecución de la siguiente secuencia de instrucciones en una implementación multiciclo?

Sw \$5,4(\$2)
Add \$6,\$3,\$1
Lw \$2 4(\$8)

7. Realiza las modificaciones necesarias sobre la ruta de datos multiciclo estudiada en clase para que se pueda ejecutar la instrucción Jal. ¿Qué señales se activarán en cada fase de ejecución de la instrucción? (se proporciona la ruta de datos en la siguiente página).

8. Se desea añadir a la ruta de datos una nueva instrucción Sm (saltar a memoria): $PC \leftarrow M[Rf1 + Rf2 + Despl]$ El formato de esta instrucción es de tipo I. Se propone modificar la ruta de datos incluyendo un multiplexor a la entrada del PC que acepte como segunda entrada el dato almacenado en MDR y ampliar el **primer** multiplexor de la ALU para que acepte además como entrada la salida del registro Sal_ALU. Obtener las fases de ejecución de esta nueva instrucción y las señales de control que se activan en cada ciclo de reloj.



8. Dado el siguiente programa escrito en lenguaje ensamblador del MIPS:

```
# Suponer que A, B y C son posiciones de memoria
# como se muestra a continuación:
# 0($a0) A
# 4($a0) B
# 8($a0) C
#
main:
    addi $t2, $zero, 1
    lw $t1, 4($a0)
    lw $t3, 0($a0)
    slt $t0, $t3, $t1
loop   beq $t0, $t2, fin
    sub $t3, $t3, $t1
    slt $t0, $t3, $t1
    jump loop
fin:   sw $t3, 8($a0)
```

a) ¿Qué hace el programa?

b) Suponiendo que el programa comienza en la posición de memoria 40_{10} , obtén la codificación en lenguaje máquina de la instrucción `beq $t0, $t2, fin`. Expresa este valor en binario especificando cada campo y el valor total en hexadecimal. (Recuerda que esta instrucción sigue el formato tipo I y que su código de operación es el 4_{10} , además, los registros $\$t0-\$t7$ corresponden a los registros temporales 8 al 15)

9. Supongamos una implementación del MIPS simple capaz de ejecutar instrucciones aritmético-lógicas, instrucciones de transferencia con memoria: `lw` y `sw`, y las instrucciones de salto: `beq` y `j`. Asumir que los tiempos de operación para las principales unidades funcionales son:

Unidades de memoria: 4 ns;

ALU y sumadores: 2 ns;

Banco de registros (lectura o escritura): 1 ns

Suponer que no hay retrasos debidos a los multiplexores, unidad de control, cables, etc.

Se pretende comparar el rendimiento de distintas implementaciones contrastando el tiempo de ejecución de la CPU. Dicho tiempo se obtendrá de la siguiente manera:

Tiempo ejecución CPU = N° de instrucciones \times CPI \times tiempo ciclo de reloj

CPI= Número de ciclos de reloj por instrucción

a) En una implementación **monociclo** de la máquina en la cual cada instrucción se ejecuta en un ciclo de reloj de longitud fija. ¿Cuál es la duración del ciclo de reloj?. Identifica el tipo de instrucción que requiere ese tiempo y muestra como lo has calculado.

b) Suponer que la máquina ejecuta el programa de la pregunta 1, con el siguiente valor de las variables: $A=11$ y $B=4$. ¿Cuál es el tiempo que tarda la CPU en ejecutarlo?.

c) Considera ahora una implementación **monociclo** donde cada instrucción se ejecuta en un ciclo de reloj de longitud variable, el cual tiene para cada instrucción la longitud necesaria para

ejecutarse. Obtén la duración del ciclo de reloj para cada tipo de instrucción, justifícalo y muestra como lo has calculado.

d) Realiza las mismas suposiciones que en la cuestión b. ¿Cuál es el tiempo que tarda la CPU en ejecutar el programa?. Compara este resultado con el de la cuestión b. ¿Qué conclusiones obtienes?

e) Supongamos ahora una implementación **multiciclo** en la cual cada instrucción se ejecuta en el mínimo número de ciclos de reloj para su clase. Obtén para cada tipo de instrucción el tiempo necesario de ejecución, justifica ese tiempo y muestra como lo has calculado.

f) Considera que el programa de la pregunta 1 con las suposiciones hechas en la cuestión b, se ejecuta en la implementación **multiciclo** de la cuestión e. Calcula el número medio de ciclos de reloj por instrucción (CPI) para el programa dado. Compáralo con el CPI de una implementación multiciclo en la cual todas las instrucciones se ejecutan en el mismo número de ciclos de reloj, siendo éste el valor necesario para que se ejecute la instrucción más lenta.

Más Ejercicios Tema 4

1. Ruta de datos del SuperAbacus.

Esta ruta de datos posee un banco de registros de uso general, que pueden ser utilizados como registros aritméticos, o dependiendo de las condiciones de direccionamiento como registros base o de índice. Todos los cálculos se realizan en un operador, que a la vez actúa como unidad aritmético - lógica y como unidad de cálculo de direcciones. Como SuperAbacus no tiene PC, se le asigna al registro R0 el que tiene que hacer estas funciones, cuyo incremento se realizará a través del operador. El tiempo de lectura y escritura en memoria es de un periodo de reloj.

NOTAS:

- En las instrucciones STORE, LOAD, ADD, SUS, Rn representa cualquiera de los registros R2 a R15
- En la instrucción MOVE, Rn y Rm , representan cualquiera de los registros R2 a R15
- Que el Código de Operación de la instrucción debe incluir el registro que se referencia
- El tiempo de acceso a la memoria es de 50ns.
- Considerar todas la señales de carga de un registro por flanco de subida, y las de salida a un bus o los ciclos de L/E de nivel.
- Las señales de control más significativas son:
 - o **NOP**: Permite la transferencia de la información entre una de las entradas de la unidad y la salida (NOP1 y NOP2).
 - o **INCR**: Transferencia desde una de las dos entradas, con incremento en 1, de la información (INC1 y INC2).
 - o **SUM**: Suma de los datos presentes en las entradas (SUM12), o de un dato de una de las entradas con el contenido del acumulador (SUM1 y SUM2).
 - o **SUS**: Resta de los datos de las entrada 2 y 1.
 - o **CMP**: Complemento de los datos de una de las dos entradas (CMP1 y CMP2).

Calcular:

- Cronograma de todas las instrucciones.
- Grafo de estados mínimo
- Tabla del estado siguiente.
- Tabla de las funciones de salida.
- Contenido de la memoria para la instrucción STORE R1. Si su código de operación es 00h

Super Ábacus

STORE R,dir ; $M(\text{dir}) \leftarrow R$
 LOAD R,dir ; $R \leftarrow M(\text{dir})$
 MOVE R_n, R_m ; $R_m \leftarrow R_n$
 ADD R,dir ; $R \leftarrow R + M(\text{dir})$
 SUS R,dir ; $R \leftarrow R - M(\text{dir})$

