

PROBLEMAS DE FUNDAMENTOS DE COMPUTADORES (HOJA 2.1)

DISEÑO DEL PROCESADOR (CURSO 2018/19)

- 1.- ¿Qué modificaciones hay que realizar en la ruta de datos y en la máquina de estados para añadir la instrucción **BNE** (bifurcar si no igual) que se comporta de manera similar a BEQ?
- 2.- Añadir los elementos y puntos de control necesarios a la ruta de datos y a la máquina de estados finitos para implementar la instrucción **ADDI**.
***addi** ri, rj, inmediato // ri = rj + SignExt(inmediato)*
- 3.- Deseamos añadir la instrucción **J** a la ruta de datos. Añadir los elementos y puntos de control necesarios para realizar esta instrucción. Mostrar los añadidos a la máquina de estados finitos que se implementó en clase para realizar esta operación.
*j dirección // PC=4*dirección, los 4 bits más significativos son los del PC+4*
El código de operación de la instrucción **J** es 000001.
- 4.- Deseamos añadir la instrucción **JAL** a la ruta de datos multiciclo. Añadir los elementos y puntos de control necesarios para realizar esta instrucción. Mostrar los añadidos a la máquina de estados finitos.
***jal** dirección // (R31= PC+4; PC=4*dir., los 4 bits más significativos son los del PC+4)*
El código de operación de la instrucción **jal** es 0000011.
- 5.- Se está considerando la ampliación del repertorio MIPS con la instrucción **ADDM** Rt, Rs, despl. que suma el contenido del registro Rt con el contenido de la posición de memoria BR[Rs] + SignExt(despl) y deposita el resultado en Rt. Modifica la ruta de datos y la FSM de control para incluir esta nueva instrucción.
***addm** rt, rs, despl //BR[rt] <- BR[rt] + MEM[BR[rs]+despl]*
- 6.- Se desea añadir al procesador multiciclo la instrucción **MOVZ**, de tipo R:
***movz** rd, rs, rt //si (BR[rt]==0) entonces BR[rd] ← BR[rs]*
Se pide:
 - a) Completar la ruta de datos de la figura con los elementos que faltan para que se puedan ejecutar las instrucciones del procesador MIPS estudiadas en clase, incluyendo además los cambios que tendrían que realizarse para poder ejecutar esta instrucción.
 - b) Añadir los estados necesarios a la máquina de estados (ver figura siguiente) para poder implementar el control de esta instrucción.
 - c) Indicar los valores que van tomando todos los registros implicados en la ejecución de la instrucción **MOVZ** R1, R2, R3. El código de operación de la función MOVZ es 000000 y el valor de los campos SHAMT y FUNCT son 00000 y 100110, respectivamente. El estado actual de la máquina es:
 - R1=0x00000017
 - R2=0x00110011
 - R3=0x00000000
 - PC=0x00003400
- 7.- Se desea añadir al procesador multiciclo la instrucción **JALR** Rd, Rs, con el formato tipo R pero código de operación 001001:
***JALR** Rd, Rs // Temp <- BR[Rs], BR[Rd] <- PC+8, PC <- Temp*
Nótese que Temp no es un registro físico, sino que se introduce en la descripción para aclarar qué ocurre en el caso de que Rs=Rd: en PC se escribe el valor antiguo de Rs, antes de que este mismo registro se actualice con el valor PC+8.
 - a) Indica todos los cambios que tendrían que realizarse a la ruta de datos para poder ejecutar esta instrucción.
 - b) Describir los cambios necesarios en la máquina de estados del controlador para poder ejecutar correctamente esta instrucción.

- c) (1 punto) Indica los valores que van tomando todos los registros implicados en la ejecución de la instrucción JALR R1,R3, sabiendo que los campos Rt, SHAMT y FUNCT se dejan a 0. El estado actual de la máquina es:
- R0=0x000000FA
 - R1=0x00000030
 - R2=0x0C001600
 - R3=0x0C00E040
 - PC=0x0C000020

8.- Sobre la ruta de datos vista en clase, indicar los valores de todas las líneas de datos y los registros, en los siguientes casos:

- a) Ejecución de la instrucción OR r1, r2, r3, al final de cada una de sus cuatro etapas

Estado actual de la máquina:

- r1=0x00000017
- r2=0x001100FF
- r3=0xFF000345
- PC=0x00003400

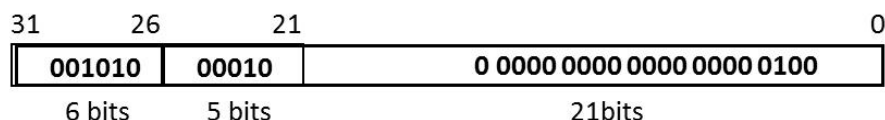
- b) Ejecución de la instrucción JAL 1024 (1024 en decimal) al final de las tres etapas

Estado actual de la máquina:

- r0=0x00000017
- PC=0x00003400

9.- Para el computador MIPS estudiado en clase, responder a las siguientes preguntas:

- a) Qué instrucción/es, del repertorio de instrucciones del procesador MIPS estudiado en clase, se ven afectada/s y no se podrían ejecutar si se elimina de la ruta de datos el registro MDR, la entrada 1 del Mux que selecciona el dato que se escribe en el banco de registros, y por lo tanto dicho Multiplexor.
- b) Partiendo de la ruta de datos completa del MIPS, se desea añadir una nueva instrucción **SMDA** (Store con Direccionamiento Absoluto). Se añade para ello un nuevo formato de instrucción con los siguientes campos (Op, Rs, dirección):



El comportamiento de la instrucción **SMDA** sería el siguiente:

SMDA Rs, dirección // $Mem[ExtCeros(dirección)] \leftarrow BR[rs]$

donde ExtCeros(dirección) extiende la dirección de 21 bits a 32 bits añadiendo ceros por la izquierda. Añade a la ruta de datos los cambios que tendrían que realizarse para poder ejecutar esta instrucción.

- c) Añade los cambios necesarios en el diagrama de transición de estados del controlador para poder ejecutar correctamente esta nueva instrucción.
- d) Indicar los cambios necesarios en la tabla de verdad del controlador (añadir las filas y columnas necesarias).

10.-Se desea añadir al procesador multiciclo la instrucción **BEQAL** Rs, Rt, Offset con código de operación 000001, cuyo comportamiento es:

Si Rs = Rt

R31 \leftarrow PC + 4;

PC \leftarrow PC + 4 + 4*Offset;

Si no

PC \leftarrow PC + 4

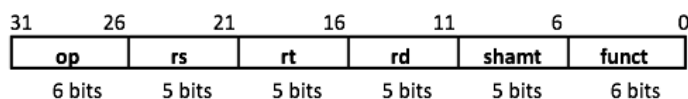
Se pide:

- Indica todos los cambios que tendrían que realizarse a la ruta de datos para poder ejecutar esta instrucción.
- Describir los cambios necesarios en el diagrama de transición de estados del controlador para poder ejecutar correctamente esta instrucción.
- Indicar los cambios necesarios en las tablas de verdad del controlador.

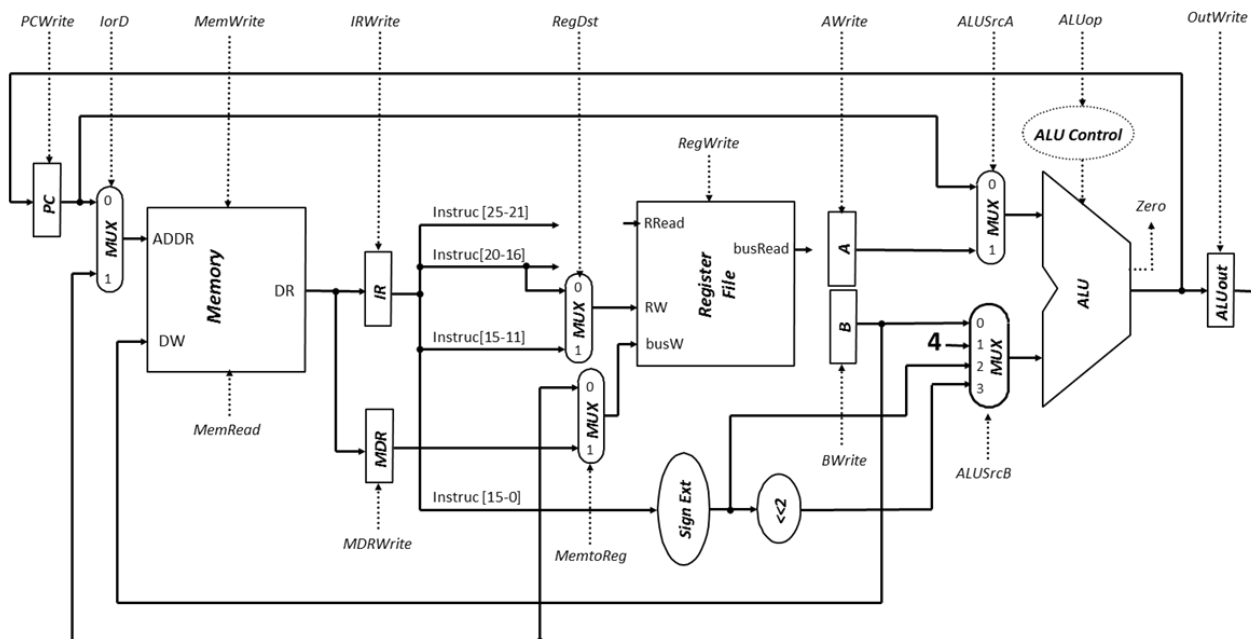
11.-(Junio 2015) Supongamos que se simplifica el Banco de Registros del MIPS estudiado en clase, de modo que sólo disponga de un único puerto de lectura, en lugar de dos, tal como se muestra en la figura de la ruta de datos.

- Completar la ruta de datos para que las transferencias " $A \leftarrow BR(rs)$ " and " $B \leftarrow BR(rt)$ " se sigan pudiendo realizar en todas las instrucciones.
- Modificar el diagrama de estados para que permita la ejecución de todas las instrucciones del MIPS sobre la nueva ruta de datos. Añadir los nuevos estados que sean necesarios, y explicar qué señales de control deben añadirse al diseño (si es el caso).
- En este procesador queremos ejecutar la instrucción **ADD** r1, r2, r3. Mostrar los valores de todos los registros modificados al final de cada fase de ejecución, asumiendo que el estado inicial del computador es: R1=0x00000017, R2=0x00110011, R3=0x00000000 y PC=0x00003400.

Recordar que el formato de las instrucciones de tipo R es:



en donde el contenido del campo "op" es $(000000)_2$, y que los valores de los campos "shamt" y "funct" para la instrucción ADD son $(00000)_2$ y $(100000)_2$, respectivamente.



12.-(Septiembre 2015) Para incluir la multiplicación **MUL** en el repertorio de instrucciones del MIPS estudiado en clase, añadimos a la ruta de datos un multiplicador y dos nuevos registros (HI y LO), tal como muestra la figura. La instrucción de multiplicación se define como sigue:

MUL *rs, rt* // $HI \leftarrow (rs * rt) [63:32]$; $LO \leftarrow (rs * rt) [31:0]$ (*rd no se usa*)

i.e. se multiplican los contenidos de los registros seleccionados mediante *rs* y *rt*, y los 32 bits más significativos del resultado se guardan en HI, mientras que los bits restantes se guardan en LO. Para poder opiar el resultado de una multiplicación al Banco de Registros, añadiremos también dos nuevas instrucciones:

MHI *rd* // $rd \leftarrow HI$ (*rs y rt no se usan*)

MLO *rd* // $rd \leftarrow LO$ (*rs y rt no se usan*)

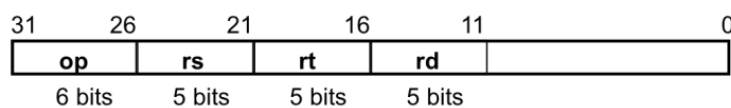
Por ejemplo, usando estas instrucciones, la secuencia:

MUL *r1, r2*

MHI *r5*

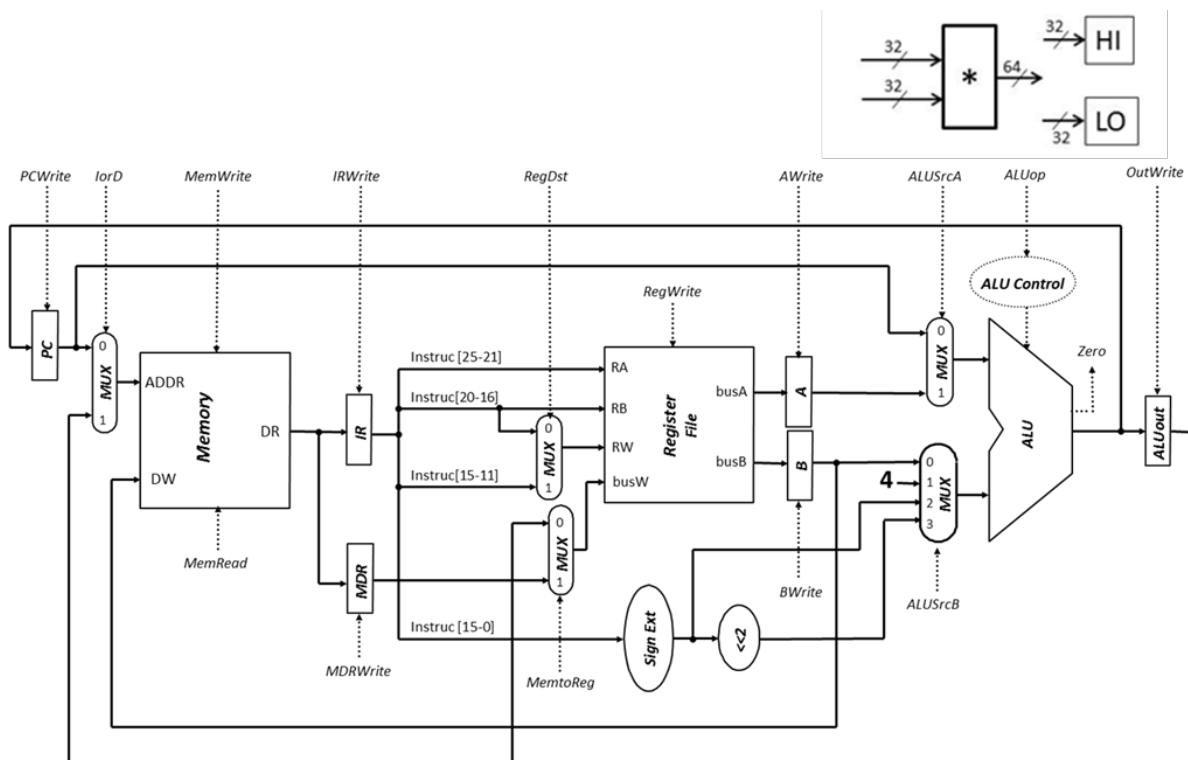
MLO *r6*

Guardaría en el registro *r5* los 32 bits más significativos del producto $r1 * r2$, mientras que los menos significativos se guardarían en *r6*. Estas tres instrucciones tienen el siguiente formato:



con códigos de operación: MUL (001010), MHI (001011) and MLO (001100).

- Modificar la ruta de datos estudiada para que se puedan ejecutar estas tres nuevas instrucciones.
- Modificar el diagrama de estados para incluir estas nuevas instrucciones.
- Modificar la tabla de verdad de la UC para implementar las nuevas instrucciones.



13.-(Junio 2016) Deseamos extender el repertorio de instrucciones del MIPS básico con la nueva instrucción **LUI** (*Load Upper Immediate*):

LUI rx, Inmediato // $RF(rx) \leftarrow (Inmediato \ll 16), PC \leftarrow PC + 4$

que se comporta como sigue: el contenido del campo Inmediato se desplaza 16 bits a la izquierda, llenándose con ceros por la derecha, y el resultado se escribe en el registro Rx del Banco de Registros. LUI se codifica con formato de instrucción de tipo I, tal como muestra la figura::

6	5	5	16
110011	not used	Rx	Inmediato

- Modificar la ruta de datos para permitir la ejecución de *Load Upper Immediate*.
- Modificar el diagrama de estados para incluir la nueva instrucción.
- Modificar la tabla de verdad, añadiendo las filas y columnas necesarias, para poder controlar la nueva ruta de datos..

NOTA: para cada parte de ejercicio, deben explicarse las razones de cada decisión.

14.-(Septiembre 2016) En una revisión del repertorio de instrucciones, los diseñadores del MIPS incluyeron algunas instrucciones nuevas que permitan cálculos relativos al PC. Una de ellas es **ADDIUPC** (*Add Immediate and PC*), con el comportamiento mostrado:

ADDIUPC rs, Inmediato // $RF(rs) \leftarrow (PC+4) + (SignExt(Inmediato))$, $PC \leftarrow PC + 4$

i.e. el contenido del campo inmediato de 21 bits se extiende en signo a 32 bits, y se suma con el valor de $PC+4$. El resultado de la suma se guarda en el registro indicado por el campo Rs. La instrucción se codifica como se muestra aquí:

6	5	21
111011	Rs	Inmediato

Para ampliar el repertorio de instrucciones del MIPS básico estudiado en clase con la nueva instrucción **ADDIUPC**, completar los siguientes apartados:

- Modificar la ruta de datos para que permita la ejecución de la nueva instrucción.
- Modificar el diagrama de estados para incluir la nueva instrucción.
- Modificar la tabla de verdad, añadiendo las filas o columnas requeridas para poder controlar la nueva ruta de datos.

15.-Suponer que los tiempos de operación para las principales unidades funcionales de una implementación MIPS son:

- Lectura unidad de memoria: 10 ns
 - Escritura unidad de memoria: 5 ns
 - ALU: 10 ns
 - Banco de registros (lectura o escritura): 5 ns
 - Multiplexores: 2 ns
 - Extensor de signo: 1 ns
 - Desplazador: 0.5 ns
- Suponiendo que el resto de elementos no tengan retardo (no hay retardo de la unidad de control, de la lectura o escritura de los registros temporales, del cableado, etc.), calcula la frecuencia de reloj máxima a la que podría trabajar este procesador.
 - Calcula el tiempo de ejecución en el procesador multiciclo de un programa con la siguiente frecuencia

de uso de las diferentes instrucciones y que conste de 1000 instrucciones:

- Carga: 22%
- Almacenamiento: 11%
- Formato R: 49%
- Salto condicional: 18%, donde la mitad de los saltos se toman.

16.-Un programa tiene 140 instrucciones de las cuales 70 tardan en ejecutarse en un determinado procesador cuatro ciclos, 35 tardan cinco ciclos, 20 tardan tres ciclos y las 15 restantes tardan siete ciclos. Calcule el CPI promedio para dicho programa. Si el procesador funciona a una frecuencia de 2.0 GHz, determine el tiempo de ejecución del programa.

17.-Se dispone de los siguientes datos de dos procesadores y de su rendimiento en la ejecución de una determinada tarea:

- PowerPC que funciona a una frecuencia de 1.8 GHz y obtiene 700 MIPS.
- Pentium 4 que funciona a 1.6 GHz y 850 MIPS.

Calcule el CPI de cada procesador.

18.-Considere los dos procesadores del ejercicio anterior. En la ejecución de un determinado programa los procesadores obtienen un CPI de 5.5 (PowerPC) y 7 (Pentium 4). El compilador genera un código máquina para dicho programa que tiene 9 millones de instrucciones (PowerPC) y 7.2 millones de instrucciones (Pentium) ¿Qué computador ejecutará más rápidamente la tarea?

19.-Ejercicio 4 Un mismo programa se ejecuta en dos computadores A y B que tienen frecuencias de reloj de 1 GHz y 1.5 GHz, respectivamente. Para ejecutar el programa en A es necesario ejecutar un cierto número de instrucciones repartidas de la siguiente manera:

	Aritmética	Load	Store	Salto tomado	Salto no tomado
Frecuencia	50%	25%	10%	10%	5%
Ciclos	4	5	4	4	3

- Calcula el CPI del programa en el computador A.
- En el computador B el número de instrucciones ejecutadas es el 60% de las ejecutadas en A y el tiempo de ejecución es la mitad que en A. ¿Cuál es el CPI obtenido en la ejecución del programa en el computador B?

20.-(Septiembre 2016) El fragmento de programa mostrado abajo se ejecuta en un computador ARM con una frecuencia de reloj de 1 GHz, con un tiempo de ejecución de 2898 ns.

```
mov r4, #0
ldr r5, =A
mov r6, #0
mov r4, #127
L1: ldr r0, [r5, r4, lsl #2]
add r6, r6, r0
sub r4, r4, #1
cmp r4, #0
bge L1
L2:
... (resto del programa)
```

- Hallar el valor del CPI para este fragmento de programa.

b) Hallar las MIPS obtenidas.

21.-(Junio 2015) El programa P se ha ejecutado en un computador C, con un rendimiento de 400 MIPS. La frecuencia de reloj de C es de 1.66 GHz.

a) Hallar el número de ciclos promedio por instrucción.

b) Asumiendo que P está compuesto por tres tipos de instrucciones, cuyas frecuencia de ejecución y duración en ciclos de reloj son, respectivamente, las siguientes:

- Acceso a memoria: 30%, 5 ciclos.
- Aritméticas: 50%, 4 ciclos.
- Bifurcación condicional: 20%, (4 ciclos si se toma, 3 ciclos si no se toma)

De acuerdo a los datos del ejercicio, hallar el número promedio de bifurcaciones tomadas, por cada 100 instrucciones de bifurcación ejecutadas.