



**Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú**

Diseño de un computador sencillo. Ejercicios

Sergio Sánchez

Xavier Masip

Departament d'Arquitectura de Computadors

1. Dado el siguiente programa en lenguaje máquina:

4803h
CA04h
1003h

- Indica a qué instrucciones en lenguaje ensamblador corresponde esta secuencia de instrucciones.
- ¿Qué operación realizan?
- Implementa un programa que realice una función equivalente.

2. - Dado el siguiente programa, escrito en algún lenguaje de alto nivel:

si ($R_1 > R_2$) y ($R_3 \leq R_4$) entonces
 $R_1 := R_1 - 1$
Sino
 $R_3 := R_3 + 1$

- Tradúcelo a lenguaje máquina suponiendo que está almacenado a partir de la dirección de memoria 00h.
- Realiza un programa equivalente en el mismo lenguaje de alto nivel.
- Indica la secuencia de estados por los que pasa el programa del apartado a) si se ejecuta la sentencia $R_3 = R_3 + 1$. ¿Cuántos ciclos son necesarios para la ejecución de dicho programa?

3. - Se quiere añadir al conjunto de instrucciones de nuestra máquina, la siguiente instrucción:

MOV R_f, R_d

Pasa el contenido del registro R_f al registro R_d . Activa los indicadores Z y N.

- Indica como será el formato de esta instrucción.
- ¿Es necesario modificar algún bloque de la Unidad de Proceso?
- ¿Es necesario modificar la Unidad de Control?

4. El siguiente programa en ensamblador realiza la función:

$$F = \sum_{i=0}^N (N - i)$$

```

00h  0000h
01h  0004h
02h  xxxxh
03h  LOAD 0(R0),R1
04h  LOAD 1(R0),R2
05h  LOAD 0(R0),R3
06h  .....
07h  BEQ 0Dh
08h  SUB R2,R1,R4
09h  ADD R3,R4,R3
0Ah  ADDI R1,#1,R1
0Bh  .....
0Ch  STORE R3,2(R0)

```

a) Indica cual es la misión de cada registro.

b) Completa el programa para que realice la función indicada.

¿Es posible obtener un programa equivalente pero que tenga menos instrucciones? En caso afirmativo, realiza la modificación.

5. El siguiente programa en lenguaje ensamblador implementa la función **F(n,i,A)** siendo **A** un vector de **n** componentes guardados desde la posición 03h hasta la 03h+(n-1) de memoria, **i** una variable inicializada a 0, **n** un valor guardado en la posición 01h y el valor de **F** se guardará en la posición 02h de memoria. En la posición 00h tenemos el valor 0.

```

LOAD 0(R0),R2
LOAD 0(R0),R4
LOAD 1(R0),R1
Bucle: SUB R4,R1,R0
      BEQ salto
      LOAD 3(R4),R3
      ADD R2,R3,R2
      ADDI R4,#1,R4
      BR bucle

```

Salto: SUBI $R_1, \#1, R_1$
 BLE guarda
 ASR R_2, R_2
 BR salto
 Guarda: STORE $R_2, 2(R_0)$

- b) Escribe el programa, en lenguaje de alto nivel, que corresponde al programa ensamblador anterior.
 - c) Escribe la expresión matemática de **F(n,i,A)**.
6. Se pretende añadir un flag de estado a nuestro procesador que llamaremos P. Este indicador tomará el valor 1 cuando el número de 1's de los 4 bits de menor peso de la salida de la ALU, $O_3O_2O_1O_0$, sea par y tomará el valor 0 en caso contrario.
- a) Indica cual es la expresión lógica de $P(O_3, O_2, O_1, O_0)$.
 - b) Realiza el circuito lógico (con puertas lógicas de 2 entradas) utilizando el MÍNIMO número de puertas posible.
7. El siguiente programa en lenguaje máquina empieza en la posición 02h:

00h 0001h
 01h 0000h
 02h 0801h
 03h 9805h
 04h D110h
 05h C911h

- a) Escribe el código en lenguaje ensamblador.
- b) Completa la siguiente tabla para 5 ciclos teniendo en cuenta que la unidad de control es cableada y ha sido diseñada a partir del grafo más simplificado.

Estado	FETCH				
Ld_IR	1				
Ld_PC	1				
Ld_R@	0				
Ld_RA	0				
Ld_RZ	0				
Ld_RN	0				
Erd	0				
L/E	0				
PC/@	0				
CRf	X				
Operar	X				
M@	02h				
Mout	0801h				
Min	X				
Din	X				
PC	02h				

IR	X				
R@	X				
RA	X				
RZ	X				
RN	X				
R1	X				
R2	X				

- c) Indica cual es la secuencia de microinstrucciones necesarias para ejecutar el programa, si la UC es microprogramada.
8. La capacidad de memoria principal de un computador es de 256 Mbytes y la Unidad de Proceso utiliza palabras de 16 bits.
- a) Diseña la memoria a partir de matrices CUADRADAS de 1 bit por chip utilizando el mínimo número de matrices. Indica como utilizas los bits de dirección para seleccionar cada palabra en la memoria.

Se quiere añadir una memoria cache asociativa por conjuntos de 512 Kbits. Cada línea de cache puede contener un bloque de memoria principal. Un bloque de memoria principal está formado por 64 palabras.

- b) Indica cuantos conjuntos forman la cache si cada conjunto tiene 4 líneas. Realiza el diseño de la cache indicando como se utilizan los bits de dirección.
9. Dado el siguiente programa en lenguaje máquina:

0801h
1002h
4802h
5001h

- a) Indica a qué instrucciones en lenguaje ensamblador corresponde esta secuencia de instrucciones.
- b) ¿Qué operación realizan?
- c) ¿Cuántos ciclos necesita para completar la ejecución, si la unidad de control es cableada?.
10. Dado el siguiente programa, escrito en ensamblador:

03h SUBI R1, #1, R1
04h ADDI R3, #1, R3
05h SUB R1, R2, R0
06h BLE 0Ah
07h SUB R3, R4, R0
08h BG 0Ah
09h BR 03h
0Ah

- a) Tradúcelo a lenguaje de alto nivel utilizando sentencias del tipo: “Si-entonces-sino”, “mientras-hacer”, “repite-hasta que”.
- b) Realiza un programa equivalente en el mismo lenguaje de alto nivel teniendo en cuenta que el bucle se ejecuta más de una vez.
- c) Indica la secuencia de estados por los que pasa el programa si $R1 \leq R2$. ¿Cuántos ciclos son necesarios para la ejecución de dicho programa?

11. Se quiere añadir al conjunto de instrucciones de nuestra máquina, la siguiente instrucción:

ANDI Rf, #num, Rd

La operación que realiza es : **Rd:= Rf AND num**, siendo **num** un valor natural de rango [0,31]. Activa los indicadores Z y N.

- a) Indica como será el formato de esta instrucción.
- b) ¿Es necesario modificar algún bloque de la Unidad de Proceso?
- c) ¿Es necesario modificar la Unidad de Control?

12. Sean 3 vectores A, B, C de 10 elementos cada uno, almacenados en las posiciones 00h, 0Ah y 14h respectivamente. El siguiente programa escrito en lenguaje ensamblador, que realiza una determinada operación con dichos vectores, se encuentra almacenado a partir de la posición 80h.

```

80h  ADD R0, R0, R1
81h  SUBI R1, #10, R0
82h  BGE 89h
83h  LOAD 0(R1), R2
84h  LOAD 10(R1), R3
85h  ADD R2, R3, R3
86h  STORE R3, 20(R1)
87h  ADDI R1, #1, R1
88h  BR 81h
89h

```

- a) Indica cual es la operación que realiza este programa.
- b) Explica cual es la misión de cada registro.

c) ¿En que caso el resultado de la operación sería falso?

13. El siguiente programa en lenguaje ensamblador implementa la función $F(a, cont)$. Las variables a y $cont$ son números enteros guardados en las posiciones $00h$ y $01h$ respectivamente. El resultado de la función se guarda en la posición $02h$. Además, se ha añadido al conjunto de instrucciones de la máquina la instrucción **MUL Rf1, Rf2, Rd** que realiza la operación de multiplicar: $Rd = Rf1 * Rf2$.

```
00h 0008h
01h 0001h
02h 0001h
03h LOAD 0(R0), R1
04h LOAD 1(R0), R2
05h LOAD 2(R0), R3
06h SUB R2, R1, R0
07h BGE 0Bh
08h ADD R2, #1, R2
09h MUL R2, R3, R3
0Ah BR 06h
0Bh STORE R3, 2(R0)
```

- Escribe el programa, en lenguaje de alto nivel, que corresponde al programa ensamblador anterior.
- ¿Qué operación realizan?
- ¿Cuántos ciclos necesita para completar la ejecución, si la unidad de control es cableada y $a=1$?

14. Se quiere añadir la instrucción **MPC Rf, Rd** al conjunto de instrucciones de nuestra máquina. Esta instrucción realiza la operación $Rd = Rf * 4$.

- Realiza el bloque combinacional que implementa esta operación.
- Rediseña la ALU para que la UP pueda realizar esta operación.
- Indica cual podría ser el formato de esta instrucción.

15. El siguiente programa en lenguaje máquina empieza en la posición $02h$:

```
00h 0001h
01h 0000h
02h 0800h
```

03h D026h
04h 5001h

- a) Escribe el código en lenguaje ensamblador.
b) Completa la siguiente tabla para 5 ciclos teniendo en cuenta que la unidad de control es cableada y ha sido diseñada a partir del grafo más simplificado.

Estado	FETCH				
Ld_IR	1				
Ld_PC	1				
Ld_R@	0				
Ld_RA	0				
Ld_RZ	0				
Ld_RN	0				
Erd	0				
L/E	0				
PC/@	0				
CRf	X				
Operar	X				
M@	02h				
Mout	0800h				
Min	X				
Din	X				
PC	02h				
IR	X				
R@	X				
RA	X				
RZ	X				
RN	X				
R1	X				
R2	X				

- c) Indica cual es la secuencia de microinstrucciones necesarias para ejecutar el programa, si la UC es microprogramada.

16. La capacidad de memoria principal de un computador es de 2^X Mbytes y la Unidad de Proceso utiliza palabras de 2^Y bits. Se quiere añadir una memoria cache con correspondencia directa de 2^M Kbits. Cada línea de cache puede contener un bloque de memoria principal. Un bloque de memoria principal está formado por 2^Z palabras.

- a) Indica cuantos bits de dirección debe tener el bus de direcciones.
b) Indica cuantos bits son necesarios para direccionar las líneas de cache.
c) Indica cuantos bloques de memoria principal se le asignarán a cada línea de cache.

17. Una matriz $M[2 \times 5]$ se encuentra almacenada en la memoria de la siguiente manera: en la posición 00h se encuentra la componente $M[1,1]$, en la dirección 01h se encuentra la componente $M[1,2]$, y así hasta la posición 09h en donde se encuentra la componente $M[2,5]$. El siguiente programa realiza el intercambio de las filas de la matriz, utilizando la zona de memoria comprendida entre las direcciones 0Ah y 0Eh como zona auxiliar:

A0h	ADDI R0,#0,R1	ABh	LOAD 5(R1), R5
A1h	ADDI R0,#5,R2	ACh	STORE R5,0(R1)
A2h		ADh	ADDI R1,#1,R1
A3h	BEQ A8h	A Eh	
A4h	LOAD 0(R1),R5	AFh	ADDI R0,#0,R1
A5h	STORE R5, 10(R1)	B0h	SUB R1,R2,R0
A6h	ADDI R1,#1,R1	B1h	BEQ B6h
A7h	BR A2h	B2h	
A8h	ADDI R0,#0,R1	B3h	
A9h	SUB R1,R2,R0	B4h	ADDI R1,#1,R1
AAh		B5h	BR B0h

- Completa el programa.
- ¿Cuál es el proceso que se realiza para hacer el intercambio?

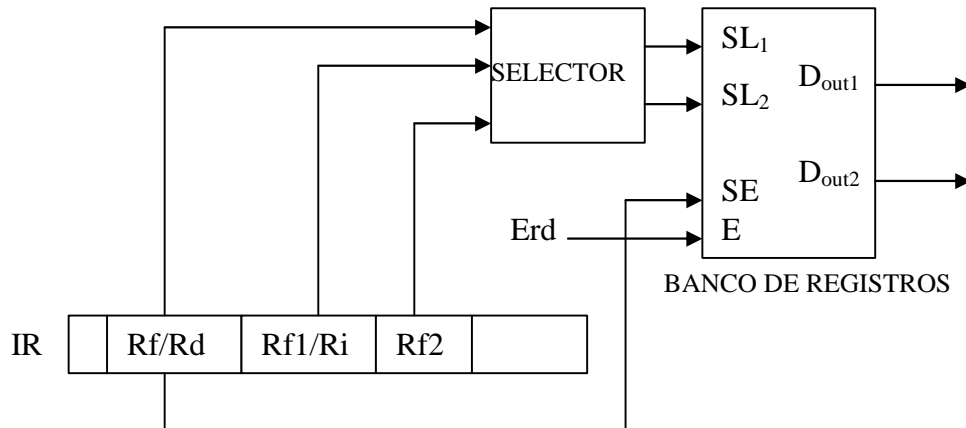
18. Dado el siguiente programa, escrito en ensamblador:

05h	SUB R1,R2,R0
06h	BLE 0Ch
07h	SUB R3,R4,R0
08h	BG 0Dh
09h	ADDI R3,#1,R3
0Ah	ADDI R2,#1,R2
0Bh	BR 07h
0Ch	ADDI R4,#1,R4
0Dh	ADDI R5,#1,R5

- Tradúcelo a lenguaje de alto nivel utilizando sentencias del tipo: “Si-entonces-sino”, “mientras-hacer”, “repite-hasta que”.
- Realiza un programa equivalente en el mismo lenguaje de alto nivel teniendo en cuenta que el bucle se ejecuta más de una vez.

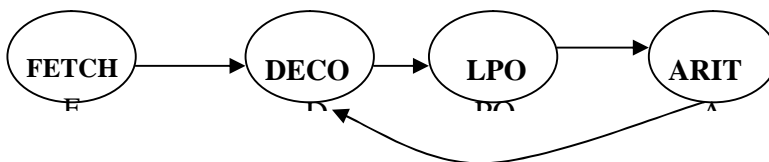
- c) Indica la secuencia de estados por los que pasa el programa si $R1 \leq R2$. ¿Cuántos ciclos son necesarios para la ejecución de dicho programa, si la unidad de control es cableada?

19. Se quiere rediseñar el banco de registros de la UP para que permita leer 2 registros a la vez. Para ello se tiene que añadir un sistema combinacional SELECTOR y se tiene que rediseñar el Banco de Registros:



- a) ¿ Que instrucciones podrán realizar la lectura simultanea en el BR y en que estado se realizará?
 b) Diseña el sistema combinacional SELECTOR, incluyendo las señales de control.
 c) Rediseña el Banco de Registros.

20. Una versión no simplificada utiliza cuatro estados para ejecutar las instrucciones Aritmético-Lógicas, tal como indica la siguiente parte del grafo de estados:



- a) Indica los bits que formaran el vector D si la búsqueda del primer operando se realiza en el estado LPO.
 b) Indica los bits que formaran el vector PO para que se realice la búsqueda del primer operando en la UP.

- c) Indica que microinstrucciones se deben añadir a la UC microprogramada y en que microdirecciones debes estar situadas.

21. Dado el siguiente programa

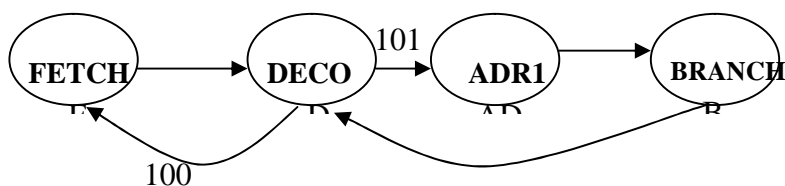
00h	ADDI R0,#0,R4
01h	ADD R0,R0,R1
02h	ADDI R0,#6,R2
03h	SUB R1,R2,R0
04h	BGE 07h
05h	ADDI R1,#1,R1
06h	BR 03h
07h	ADDI R4,#1,R4
08h	SUBI R4,#5,R0
09h	BLE 03h

- a) Tradúcelo a lenguaje de alto nivel utilizando sentencias del tipo: “Si-entonces-sino”, “mientras-hacer”, “repite-hasta que”.
- b) Realiza un programa equivalente en el mismo lenguaje de alto nivel teniendo en cuenta que el bucle se ejecuta más de una vez.

22. Se pretende almacenar los datos y las instrucciones en memorias separadas. Para ello, se ha utilizado una memoria que llamaremos **MEMdatos** para almacenar los datos y una memoria **MEMinst** para almacenar las instrucciones

- a) Dibuja las modificaciones que se tendrán que realizar en la UP para utilizar estas dos memorias.
- b) ¿Cómo afectará esta modificación en el grafo de estados de la UC?

23. Una versión no simplificada utiliza cuatro estados para ejecutar las instrucciones de salto, tal como indica la siguiente parte del grafo de



estados:

- a) Indica los bits que formaran el vector D si el cálculo de la dirección de salto se realiza en el estado ADR1
- b) Indica los bits que formaran el vector AD para que se realice el cálculo de la dirección de salto en la UP.
- c) Indica que microinstrucciones se deben añadir a la UC microprogramada, en que microdirecciones debes estar situadas y que microinstrucciones quedan modificadas.

24. Se quiere realizar un programa que compruebe si se ha producido o no *overflow* en una operación A-L. Este programa se ejecutará después de cada instrucción A-L Rf1,Rf2,Rd. Si se ha producido *overflow* el programa debe finalizar (saltar al final del programa). En caso contrario el programa debe continuar.

- a) Indica en que instrucciones A-L se puede producir *overflow* y en que casos.
- b) realiza el programa en lenguaje de alto nivel utilizando las sentencias que se han utilizado en clase.
- c) Traduce el programa a lenguaje ensamblador.

25. Se quiere añadir la instrucción **BO** *direccion_salto* (*Branch if Overflow*) al conjunto de instrucciones de nuestra máquina. La instrucción realiza un salto a *direccion_salto* si en la operación anterior se ha producido *overflow*. En caso contrario el programa sigue en secuencia. Además, será necesario incluir un indicador de condición (O) en la UP de tal forma que si la operación realizada en la ALU produce *overflow* entonces O=1. En caso contrario O=0.

- a) Indica el formato de la nueva instrucción
- b) Realiza las modificaciones oportunas en la UP
- c) ¿Qué modificaciones necesita la UC?

26. Suponiendo que el tiempo de respuesta de los distintos elementos de la UP es el siguiente:

- 20 ns para cada multiplexor y decodificador.
- 60 ns para el sumador de direcciones y el incrementador conectado al PC
- 50 ns para el bloque que realiza la evaluación de la condición de salto
- 30 ns para leer un registro del BR.
- 120 ns tiempo de respuesta de la ALU
- 200 ns tiempo requerido para leer o escribir un dato de la memoria RAM

- a) Calcula el tiempo requerido para ejecutar cada uno de los estados del grafo simplificado de la UC cableada (FETCH, DECO, LOAD, STORE, ARIT, BRANCH).
- b) Indica cual será el tiempo de ciclo de nuestro procesador.
- c) Calcula el tiempo requerido para ejecutar cada una de las instrucciones si la UC es microprogramada.

27.a Se quiere realizar una operación entre dos matrices cuadradas A[mxm] y B[mxm]. La matriz A está almacenada a partir de la posición 00h de memoria y la matriz B a partir de la posición 10h. El resultado de la operación se almacenará a partir de la posición 20h. El valor de m se encuentra almacenado en la posición 30h. La operación consiste en :

ejemplo: para m=2.

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \text{oper} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} (a_{1,1} + b_{1,1}) - (a_{1,2} + b_{2,1}) & (a_{1,1} + b_{1,2}) - (a_{1,2} + b_{2,2}) \\ (a_{2,1} + b_{1,1}) - (a_{2,2} + b_{2,1}) & (a_{2,1} + b_{1,2}) - (a_{2,2} + b_{2,2}) \end{pmatrix}$$

- a) Realiza un programa en lenguaje de alto nivel utilizando las sentencias que se han utilizado en clase
- b) Traduce el programa a lenguaje ensamblador

28. Se quiere cambiar la forma de obtener el segundo operando en las instrucciones A-L. Antes el segundo operando se encontraba en un registro del BR o era un operando inmediato. Ahora este operando, **K**, se encuentra en memoria. Las instrucciones A-L serán:

A-L Rf,K,Rd

Siendo:

Rf: Un registro del BR donde se encuentra almacenado el primer operando (número entero de 16 bits).

K: un número entero de 16 bits almacenado en la posición de memoria consecutiva a la posición de memoria de la instrucción A-L que estamos ejecutando.

Rd: un registro del BR donde guardaremos el resultado de la operación.

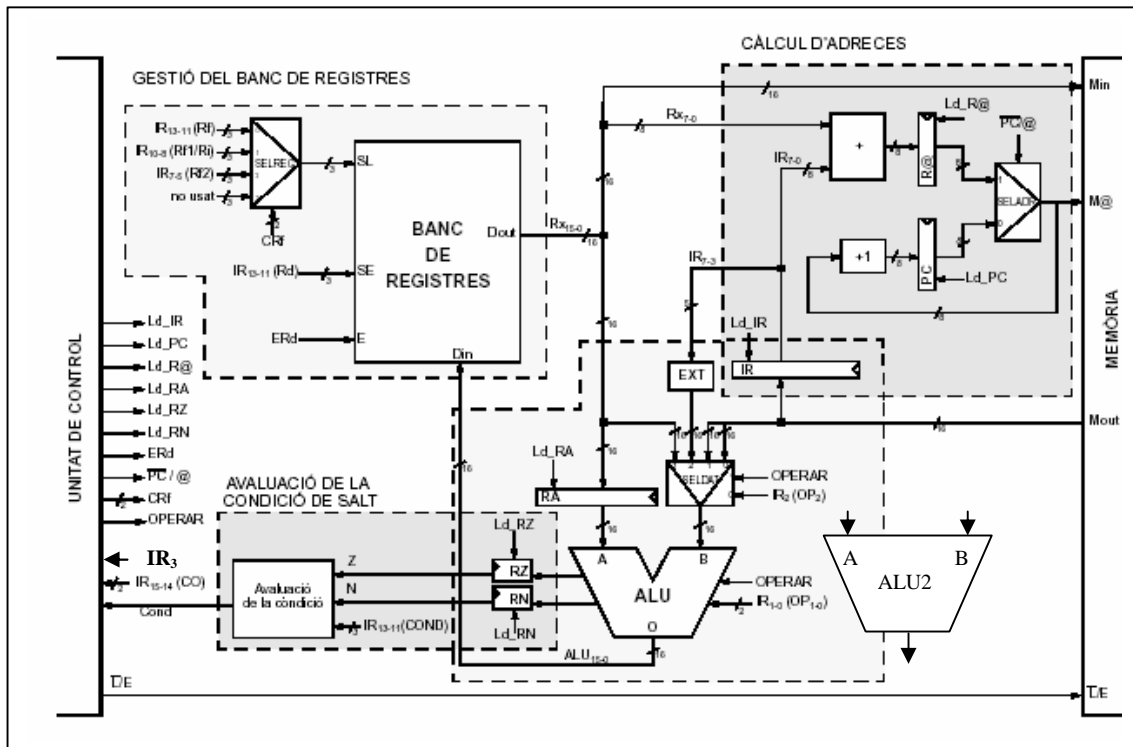
- a) Indica el formato de las instrucciones A-L.
- b) Realiza las modificaciones oportunas en la UP, intentando aprovechar al máximo los componentes ya existentes.
- c) Indica que modificaciones se deben realizar en la UC.

29. La capacidad de memoria de un procesador es de 2^X Mbytes y utiliza palabras de 2^Y bits. Se pretende diseñar la memoria a partir de matrices CUADRADAS (2^P filas x 2^P columnas) de 1 bit por chip, teniendo en cuenta que $2P < 23+X-Y$. Indica de forma JUSTIFICADA:

- a) La cantidad de matrices CUADRADAS que se utilizarán.
- b) Como se utilizarán los bits de la dirección para seleccionar cada palabra en la memoria.

30. Se quiere realizar las siguientes modificaciones:

- Suprimir las instrucciones SUBI y ADDI del conjunto de instrucciones.
- Añadir 4 instrucciones nuevas que realicen operaciones en coma flotante: suma (ADDFP), resta (SUBFP), desplazamiento (ASRFP) y multiplicación lógica (ANDFP). Estas instrucciones modifican los bits de condición N y Z.
- Añadir una nueva ALU (ALU 2) para que realice las operaciones en coma flotante (ver figura).
- El bit IR_3 se utiliza para diferenciar entre operaciones con números enteros y operaciones en coma flotante.



Realiza las siguientes modificaciones en la UP:

- Indica el formato de cada una de las nuevas instrucciones.
- Diseña la ALU2, conectando cada uno de los bloques que la forman (sumador, restador, desplazador y multiplicador lógico en coma flotante) e indicando las entradas de selección de las operaciones (no diseñar el cálculo de N y Z).
- Conecta la ALU2 con el resto de los bloques de la UP.

Realiza las siguientes modificaciones en la UC:

- Añade las nuevas señales de control indicando que función realizan.
- Modifica el grafo de estados de la UC para que se puedan ejecutar las nuevas instrucciones (el grafo debe estar completo: estados, salidas y entradas). Indica cual es la función de cada estado nuevo.
- Indica el valor de los nuevos vectores de salida que la UC envía en cada estado nuevo hacia la UP.

31. Se quiere calcular la serie de Fibonacci para n términos. Esta serie se caracteriza porque cada término es la suma de los dos anteriores. Ejemplo:

0 1 1 2 3 5 8 13...

- a) Realiza un programa en lenguaje de alto nivel que calcule la serie de Fibonacci para n términos y los guarde en un vector V de n componentes. Indica claramente cuál es el significado de cada una de las variables del programa.
- b) Realiza un programa en ensamblador que calcule n términos de la serie de Fibonacci y los almacene en las últimas posiciones de la memoria. Considera que el término 0 y el término 1 (los 2 primeros términos de la serie) se encuentran almacenados en las posiciones FFh y FEh respectivamente. El valor de n se encuentra almacenado en la posición 00h de memoria. Relaciona cada parte del programa en ensamblador con las partes del programa del apartado a).

32. Dado el siguiente programa en ensamblador:

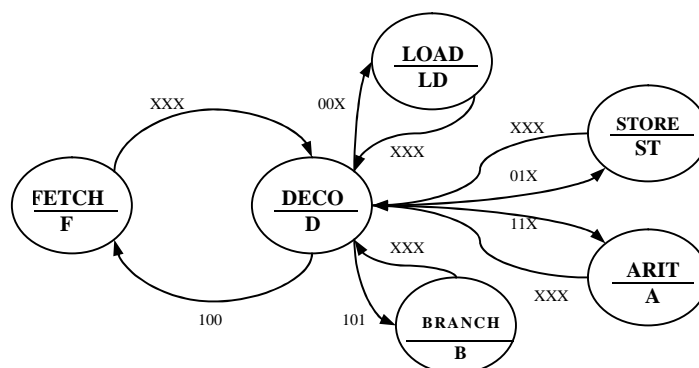
```

SUB R2,R3,R0
BG bucle2
bucle1: SUB R5,R6,R0
BG fin
ADDI R5,#1,R5
BR bucle1
bucle2: SUB R4,R7,R0
BL fin
SUBI R4,#1,R4
BR bucle2
fin:

```

- a) Realiza un programa en lenguaje de alto nivel que corresponda a este programa.
- b) Realiza un programa en lenguaje de alto nivel que sea equivalente al del apartado a).

33. Una UC optimizada ejecuta el siguiente grafo:



- a) Indica que modificaciones se deben realizar en la UP para poder ejecutar todas las instrucciones de la máquina según este grafo. Dibuja la nueva Unidad de Proceso y explica las modificaciones que se han realizado.
- b) Indica el valor de las señales de control correspondientes a los nuevos vectores LD y ST. Explica el significado de cada uno de los bits que forman dichos vectores

34. La capacidad de memoria principal de un computador es de 2 Gbytes y la Unidad de Proceso utiliza palabras de 64 bits.

- c) Diseña la memoria a partir de matrices CUADRADAS de 1 Mbits. Indica como utilizas los bits de dirección para seleccionar cada palabra en la memoria.

Se quiere añadir una memoria cache asociativa por conjuntos de 1 Mbits. Cada línea de cache puede contener un bloque de memoria principal. Un bloque de memoria principal está formado por 128 palabras.

- d) Indica cuantos conjuntos forman la cache si cada conjunto tiene 16 líneas. Realiza el diseño de la cache indicando como se utilizan los bits de dirección.