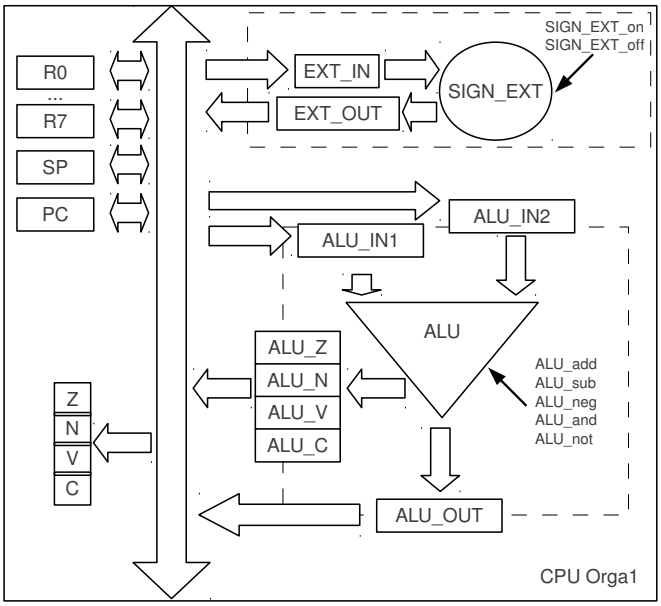
Práctica 4 - Microarquitectura del CPU

Organización del Computador 1

Primer Cuatrimestre 2023

Ejercicio 1 El siguiente esquema muestra algunos de los componentes de la microarquitectura de un modelo del procesador Orga1:



El bus interno tiene 16 bits. Todos los registros son de 16 bits excepto Z, C, V, N, ALU_Z, ALU_C, ALU_V y ALU_N. IR0 es el registro que ocupa la primera palabra de la instrucción a ejecutar. El procesador posee una Unidad Aritmético Lógica (ALU) y un extensor de signo complemento a 2 (SIGN_EXT) de 16 bits. La unidad de control (no dibujada) controla las siguientes señales:

- Microoperaciones de la ALU:
  - ALU_{add}: activa la suma. ALU_OUT := ALU_IN1 + ALU_IN2
  - ALU_{sub}: activa la resta. ALU_OUT := ALU_IN1 - ALU_IN2
  - ALU_{neg}: activa la negación. ALU_OUT := - ALU_IN1
  - ALU_{and}: activa el and-lógico. ALU_OUT := ALU_IN1 AND ALU_IN2
  - ALU_{not}: activa el not-lógico. ALU_OUT := NOT ALU_IN1
- Microoperaciones del SIGN_EXT:
  - $SIGN\_EXT_{on}$: activa la operación de extensión de signo de 8 bits a 16 bits.
  - $SIGN\_EXT_{off}$: copia el contenido de EXT_IN en ALU_IN2.

Indicar cuál es la secuencia de microoperaciones que debe realizar la unidad de control al ejecutar las siguientes instrucciones:

a) MOV R0, R1

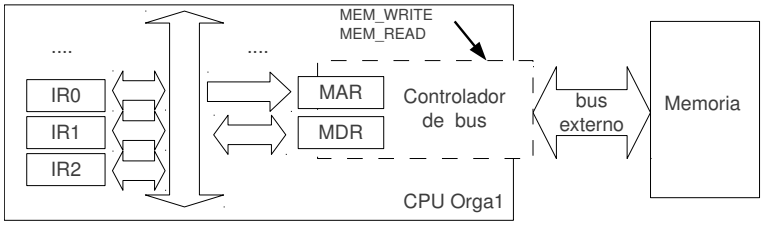
b) ADD R0, R1

c) AND R0, R1

d) NOT R0

e) NEG R0

Ejercicio 2 El esquema del ejercicio anterior se amplía con los siguientes componentes de la computadora:



IR0, IR1 e IR2 son tres registros de 16 bits que almacenan el valor de la instrucción fetcheada. Un controlador de bus, que es el encargado de las comunicaciones con otros dispositivos, cuenta con dos registros: MAR y MDR. MAR es el Memory Address Register. MDR es el Memory Data Register. Además, la unidad de control (no dibujada) controla las siguientes señales del controlador del bus:

- MEM_WRITE: Inicia el protocolo de escritura para escribir el contenido del registro MDR en la dirección de memoria indicada por el MAR.
- MEM_READ: Inicia el protocolo de lectura para obtener del contenido de la dirección de memoria indicada por el MAR, cuyo valor podrá ser leído luego en MDR.

Sean las siguientes instrucciones:

i) MOV R0, R1

ii) MOV R0, 0xFF00

iii) MOV [0xAA00], 0xFF00

iv) MOV [0xAA00], [[0xFF00]]

v) MOV [0xAA00],[R0]

vi) MOV [0xAA00],[R0+0xFF00]

a) ¿Cuál es el contenido de los registros IR0, IR1 e IR2 al finalizar la decodificación de cada una de ellas?

b) Indicar cuál es la secuencia de microoperaciones que debe realizar la unidad de control al ejecutarlas.

Ejercicio 3 Suponga que se cuenta con la microarquitectura del ejercicio 2 extendida con una señal que permite colocar el valor 0x0001 en el registro ALU_IN2. Indicar cuál es la secuencia de microoperaciones que debe realizar la unidad de control para:

- JMP R0
- JE 0xFF
- JG 0xFF
- CALL R0
- RET

Ejercicio 4 Se desea extender la arquitectura de la máquina ORGA1 con una nueva instrucción SHL1 con el siguiente comportamiento:

Operación	Efecto
SHL1	Dest := Dest $\ll$ 1 Shift hacia la izquierda de un 1 bit Modifica los flags, deja C y V en 0

a) Extender la microarquitectura del ejercicio 3 para implementar la nueva instrucción.

b) Indicar la secuencia de microoperaciones que se necesitan para ejecutar: SHL1 R1.

c) Se restringe el modo de direccionamiento para la instrucción SHL1, limitándola a decalar sólo el registro R1. ¿Qué cambios puede introducir en la microarquitectura para aprovechar esta limitación?

Ejercicio 5 Escriba la secuencia de microoperaciones realizadas por la unidad de control para efectuar las siguientes acciones. La microarquitectura de la computadora ORGA1 es la dada en el ejercicio 3:

a) Instrucciones tipo 3 y 4:

```
1) Fetch primer palabra de la instrucción a IR0
2) Incrementar PC
```

b) Instrucciones tipo 2a:

```
1) Fetch primer palabra de la instrucción a IR0
2) Incrementar PC
3)   Si Dest es Inmediato, Directo, Indirecto o Indexado
4)      Fetch segunda palabra de la instrucción a IR1
5)      Incrementar PC
```

Ejercicio 6 La computadora MARIE posee el siguiente set de instrucciones

Instrucción	CodOp	Significado
LOAD X	0001	AC := [X]
STORE X	0010	[X] := AC
ADD X	0011	AC := AC + [X]
SUBT X	0100	AC := AC - [X]
JUMP X	1001	PC := X
SKIPCOND	1000	Saltea la próxima instrucción de acuerdo a la condición

El formato de instrucción de MARIE es el que sigue:

CodOp	Dirección/Valor
4 bits	12 bits

La condición de salto se codifica en los bits 10 y 11 del campo Dirección/Valor. Todas las condiciones inspeccionan el valor actual del registro AC, utilizando la codificación complemento a 2 de 16 bits:

Codificación	Condición
00	Ignora la próxima instrucción si AC es negativo
01	Ídem si AC es 0
10	Ídem si AC es mayor o igual que 0

a) Definir el camino de datos y la microarquitectura del CPU de MARIE para soportar la implementación de al menos estas instrucciones.

b) Implementar las siguientes instrucciones

   i) LOAD X
   ii) ADD X
   iii) JUMP X
   iv) SKIPCOND

c) Modificar la microarquitectura del CPU de MARIE en caso que se desee incorporar la siguiente instrucción:

Instrucción	CodOp	Significado
LROT X	1010	Rotación hacia la izquierda Se rotan tantos bits como indica el valor X

Ejercicio 7 Indicar cuál es la secuencia de microoperaciones que debe emitir la unidad de control para realizar el fetch de una instrucción en la máquina del ejercicio 6.

Ejercicio 8 Una computadora cuenta con 128 registros de uso general, direcciones de 32 bits, direccionamiento a byte y tamaño de palabra de 32 bits. Su conjunto de instrucciones es el que sigue:

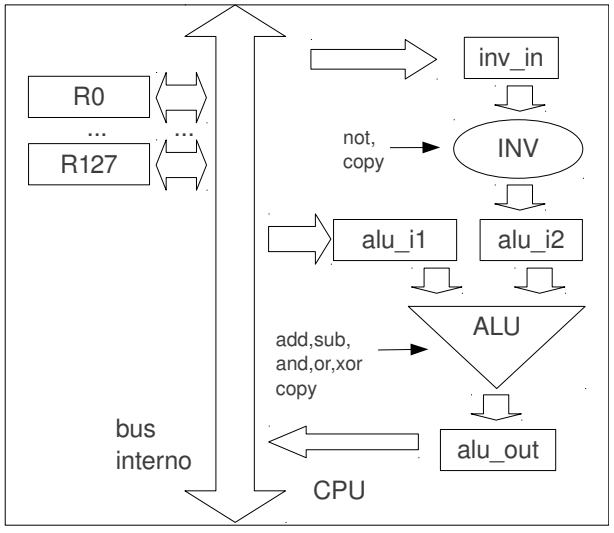
Instrucciones	Operandos	Efecto
add/sub	reg1,reg2,reg3	reg3 := reg1 +/- reg2
and/or/xor	reg1,reg2,reg3	reg3 := reg1 and/or/xor reg2
andn/orn/xorn	reg1,reg2,reg3	reg3 := reg1 and/or/xor (not reg2)
load/store	reg,addr	carga/guarda el valor del registro desde/hacia memoria
be/bg/bl	reg,addr	PC := addr si reg=0/reg>0/reg<0

El formato de la instrucción es:

0	codOp	reg1	reg2	reg3
1 bit	10 bits	7 bits	7 bits	7 bits

1	codOp	sin usar	reg	addr
1 bit	3 bits	21 bits	7 bits	32 bits

Sea la siguiente microarquitectura parcial del CPU:



a) Completar la microarquitectura de la computadora para soportar la implementación del conjunto de instrucciones.

b) Describir las acciones de la unidad de control para ejecutar la siguiente instrucción:

```
xorn R0,R1,R2
```

c) Describir la secuencia de microoperaciones que utiliza la unidad de control para realizar el fetch de una instrucción

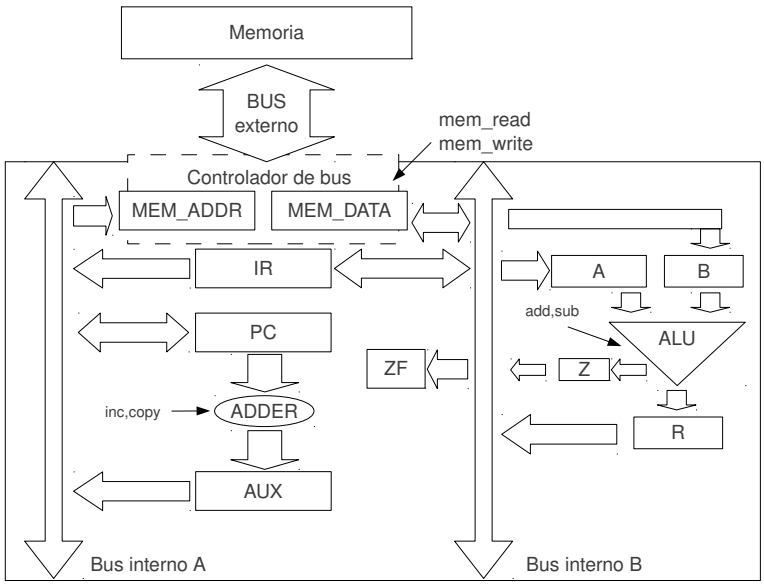
Ejercicio 9 Una computadora cuenta con direcciones de 7 bits, direccionamiento a byte y tamaño de palabra de 16 bits. Su conjunto de instrucciones es el que sigue:

Instrucciones	Operandos	Efecto
MOV	dir1,dir2	[dir1] := [dir2]
ADD	dir1,dir2	[dir1] := [dir1] + [dir2]. Modifica ZF
CMP	dir1,dir2	ZF :=1 sii [dir1]-[dir2]=0
BEQ	dir1	si ZF=1 entonces PC:=dir1

El formato de la instrucción es:

codOp	dir1	dir2
2 bits	7 bits	7 bits

La microarquitectura de la computadora se muestra en el gráfico. La unidad aritmético-lógica (ALU) sólo cuenta con las operaciones ALU_ADD y ALU_SUB. El incrementador (ADDER) posee 2 operaciones que operan sobre su entrada: ADDER_inc incrementa 1, y ADDER_copy sólo lo copia.



a) Describir las acciones de la unidad de control para ejecutar las siguientes instrucciones:

   i) ADD dir1, dir2

   ii) BEQ dir1

b) Describir la secuencia de microoperaciones que utiliza la unidad de control para realizar el fetch de una instrucción.

c) ¿Cuál es el tamaño mínimo de cada bus interno?