Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2343 - Arquitectura de Computadores

Tarea 6 - Bonus

12 de octubre de 2018

II semestre 2018

Gabriel Balassa - Cristobal Ilabaca - Diego Pedraza

Para este caso, utilizaremos el esquema completo del computador básico en arquitectura Von Neumann provisto en el enunciado.

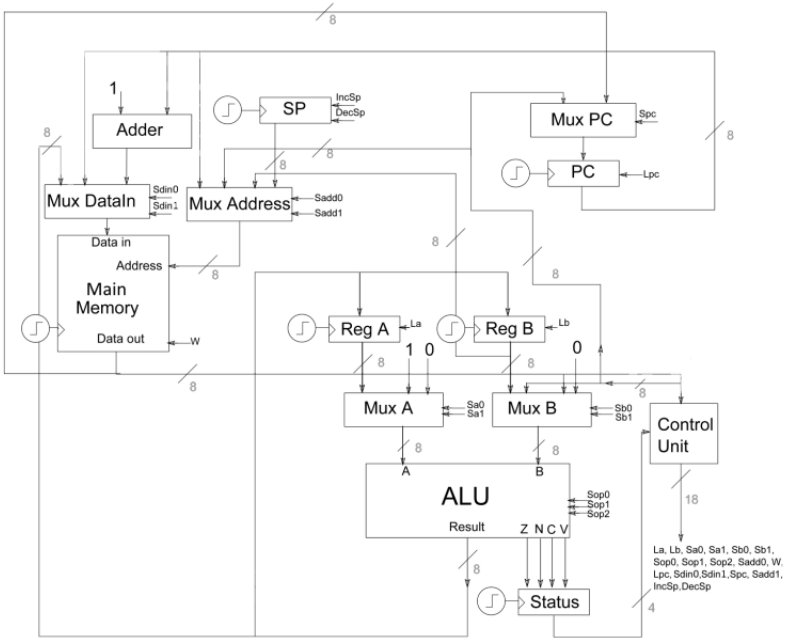

Figura 1: Arquitectura Von Neumann computador básico.

Esto implica que ya no tendremos una memoria de instrucciones y por ende, como es bien sabido, las instrucciones, variables e información irán a una sola memoria principal. Esto implica que ahora, la señal proviniente del PC (program counter) también deberá ser leída por la memoria principal, ergo, el Mux Datain, que selecciona que bus de datos ingresa a la memoria, deberá tener una entrada más y por ende dependerá no de una, sino de dos señales de control para su correcta multiplexación.

Ahora bien, para poder incluir subrutinas en el funcionamiento de nuestro computador, debemos generar las señales de control respectivas con una ISA coherente con la implementada hasta el momento. Básicamente, debemos transformar las señales de control para las funciones CALL, RET, PUSH A,B y POP A,B. Según la descripción de la ISA para un computador básico en arquitectura Harvard, necesitaremos generar las siguientes señales de control:

Instrucción	Operandos	Opcode	Condition	Lpc	La	Lb	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin0	Spc0	W	IncSp	DecSp
CALL	Dir	1000101		1	0	0	-	-	-	SP	PC	LIT	1	0	1
RET		1000110		0	0	0	-	-	-	-	-	-	0	1	0
		1000111		1	0	0	-	-	-	SP	-	DOUT	0	0	0
PUSH	A	1001000		0	0	0	A	ZERO	ADD	SP	ALU	-	1	0	1
PUSH	B	1001001		0	0	0	ZERO	B	ADD	SP	ALU	-	1	0	1
POP	A	1001010		0	0	0	-	-	-	-	-	-	0	1	0
		1001011		0	1	0	ZERO	DOUT	ADD	SP	ALU	-	0	0	0
POP	B	1001100		0	0	0	-	-	-	-	-	-	0	1	0
		1001101		0	0	1	ZERO	DOUT	ADD	SP	ALU	-	0	0	0

Figura 2: ISA Harvard para subrutinas y SP.

Para lo que se sugiere la siguiente ISA:

Instrucción	Operandos	Operación	Condición	Opcode
CALL	Dir	Mem[SP] = PC + 1, SP-, PC = Dir	-	00101111
RET		SP+	-	00111111
		PC = Mem[SP]	-	01000001
PUSH	A	Mem[SP] = A, SP-	-	01000010
PUSH	B	Mem[SP] = B, SP-	-	01000011
POP	A	SP+	-	01000100
		A = Mem[SP]	-	01000101
POP	B	SP+	-	01000110
		B = Mem[SP]	-	01000111

Cuadro 1: ISA Von Neumann para subrutinas y SP.

De esta manera, tenemos que para cada operación:

- CALL: seteamos Lpc = 1, por ende el PC queda en modo de cargam seteamos el valor de la señal W = 1(ya que queremos guardar el valor del PC + 1 en memoria, i.e. 'stackearlo' en memoria), generamos un decremento en el SP y en el PC almacenamos la dirección de la subrutina.

- RET: este llamado se realiza en dos ciclos, (i) primero se realiza un incremento en el PC (haciendo IncSp = 1), luego el direccionamiento de la memoria principal recibe el valor en memoria señalado por SP.
- PUSH X: en este caso se guarda el valor del registro X (A o B) en la posición actual del SP y se genera un decremento en el valor del PC.
- POP X: en este caso, tenemos que generar un aumento en el valor del PC (IncSp = 1) y luego cargamos el registro X (A o B) con el valor proviniente de la posición del SP en memoria.

De esta manera notamos que una de las mayores diferencias es que ahora el MuxIn debe ser entregar el valor del PC en la dirección dada por el SP cuando corresponda, de manera que se agregará una señal de control para tales propósitos, es decir, tendremos las señales de control Sdin0 y Sdin1.