Práctica 3

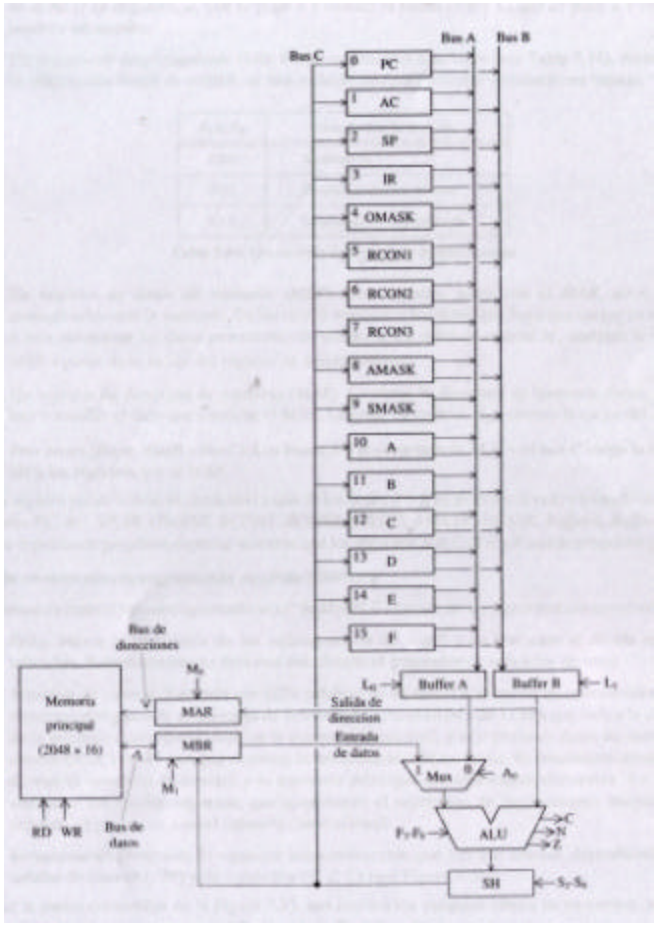
1. Introducción

Para la realización de esta práctica se hace uso de un simulador desarrollado por alumnos de la Escuela de Ingenieria Técnica de Informática de la UNED, basado en el computador pedagógico sencillo SIMPLE2, descrito en el libro de S.Dormido (E.y T. De Computadores, 2000) y que dispone de una unidad de control microprogramable.

El simulador permite ver como funciona el corazón de un computador: su unidad de control. Dado un conjunto de instrucciones máquina, el simulador carga en su memoria de control el conjunto de microinstrucciones que implementan dicho repertorio, de esta forma se puede experimentar desde dentro el funcionamiento de un computador elemental. El simulador nos permite además conseguir un segundo objetivo, ya que una vez microprogramado un repertorio de instrucciones, permite ejecutar programas en su lenguaje máquina. De este modo, también se puede utilizar como un sencillo entrenador de lenguaje máquina sobre el que probar distintos algoritmos..

2. Estructura de SIMPLE2

- 2.1. Camino de datos



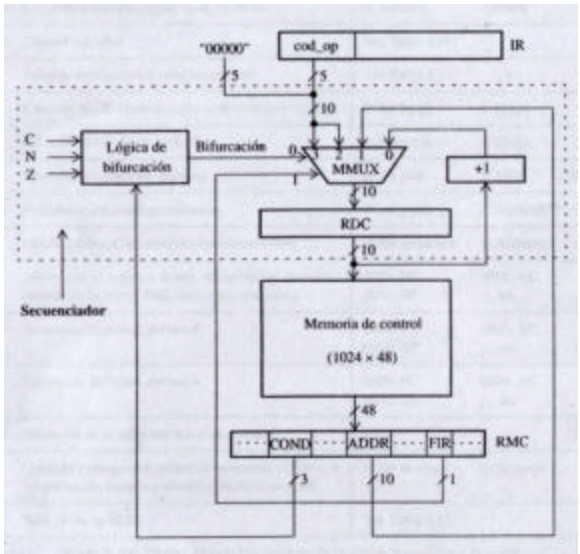
- 2.2. Descripción de registros

Registro	Función del registro
PC:	Contador de programa: contiene la dirección de memoria de la siguiente instrucción a ejecutar
AC:	Acumulador: sirve para cargar datos desde la memoria o hacia la memoria
SP:	Puntero de pila: dirección de la última posición de la pila en memoria
IR:	Registro de instrucción: contiene la instrucción máquina que se está ejecutando
OMASK:	Registro constante cargado con el valor $0000011111111111_2 = 2047_{10}$
RCON1:	Registro constante 1 cargado con el valor $0000000000000000_2 = 0_{10}$
RCON2:	Registro constante 2 cargado con el valor $0000000000000001_2 = +1_{10}$
RCON3:	Registro constante 3 cargado con el valor $1111111111111111_2 = -1_{10}$
AMASK:	Registro constante cargado con el valor $0000111111111111_2 = 4095_{10}$
SMASK:	Registro constante cargado con el valor $0000000011111111_2 = 255_{10}$
A, B, C, D, E, F:	Registros de propósito general

$F_3F_2F_1F_0$	Operación	Descripción de la salida
0000	A + B	Suma aritmética de A y B (A procede de BufferA y B de BufferB)
0001	A ∧ B	AND-lógico entre A y B
0010	A	No hace nada, permite que A pase a través de la ALU (modo transparente)
0011	$\overline{A}$	Complemento a 1 de A
0100	A ∨ B	OR-lógico entre A y B
0101	A ⊕ B	OR-exclusivo entre A y B

$S_2S_1S_0$	Descripción de la salida
000	No desplaza.
001	Desplaza un bit a la derecha.
010	Desplaza un bit a la izquierda.

3. Unidad de control microprogramable



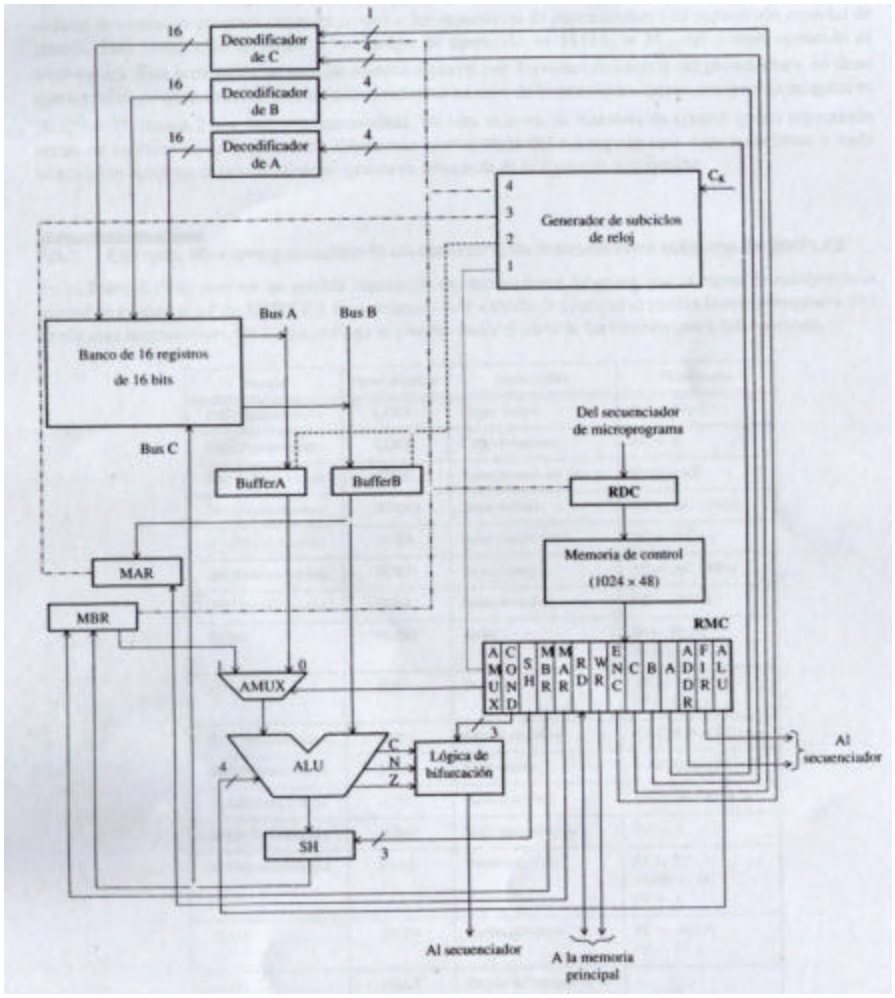
Entradas de control a MMUX		Próxima dirección
1 (FIR)	0 (Bifurcación)	
0	0	*RDC + 1*
0	1	*RMC[ADDR]*
1	0	*IR[cod_op]*
1	1	*IR[cod_op]*

$C_2 C_1 C_0$	Acción
0 0 0	No salta.
0 0 1	Salta si N = 1.
0 1 0	Salta si Z = 1.
0 1 1	Salta si C = 1
1 0 0	Salta siempre

- 3.1. Formato de las Instrucciones

Campo de la μI		Acción
AMUX:	Controla la entrada izquierda de la ALU	0: BufferA 1: MBR
COND:	Control del salto	Ver Tabla 7.16
SH:	Función del registro de desplazamiento	Ver Tabla 7.14
MBR:	Carga de MBR desde el registro de desplazamiento	0: No carga 1: Carga
MAR:	Carga de MAR desde BufferB	0: No carga 1: Carga
RD:	Petición de lectura de memoria	0: No pide 1: MBR ← M[MAR]
WR:	Petición de escritura en memoria	0: No pide 1: M[MAR] ← MBR
ENC:	Habilita el bus C (control de almacenamiento)	0: No almacena 1: Almacena
C:	Selecciona el registro donde almacenar el dato procedente de la ALU. ENC debe estar activada a 1.	0000: PC 0001: AC 0010: SP .... etc
B:	Selecciona la fuente del bus B	0000: PC 0001: AC 0010: SP .... etc
A:	Selecciona la fuente del bus A	0000: PC 0001: AC 0010: SP .... etc
ADDR:	Dirección de la siguiente microinstrucción	
FIR:	Controla la carga del código de operación (5 bits) de la instrucción máquina almacenada en IR en RDC	0: No se carga 1: Se carga
ALU:	Función de la ALU	Ver Tabla 7.13

- 3.2. Fases de la ejecución de una instrucción



4. Instrucciones máquina

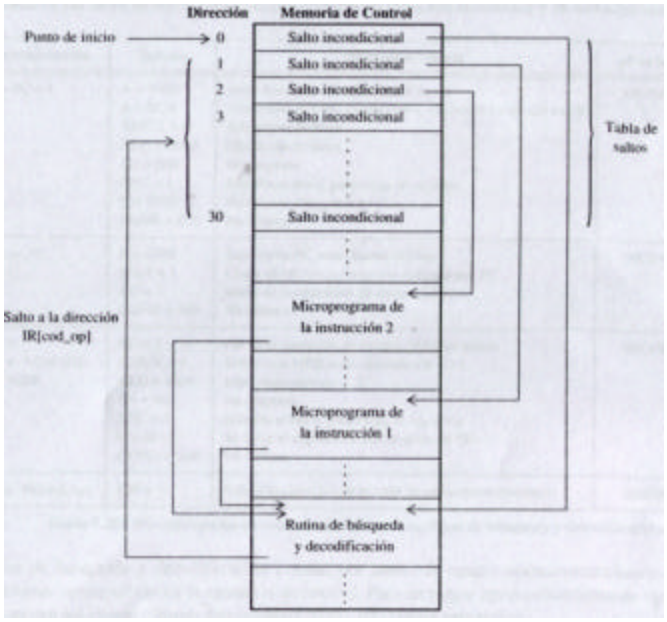
La unidad de control de un computador es una máquina secuencial y de forma cíclica realiza una serie de operaciones:

- *Búsqueda de la siguiente instrucción máquina*
- *Decodificación*
- *Búsqueda de los operandos*
- *Ejecución*
- *Almacenamiento de resultados*
- *Determinación de la dirección de la siguiente instrucción máquina*

4.1. Búsqueda de la siguiente instrucción máquina

Microinstrucción	Señales	Microoperaciones	µI en hexadecimal
PC ← PC + 1	A = 0000 B = 0110 AMUX = 0 ALU = 0000 SH = 000 ENC = 1 C = 0000 COND = 000	Selecciona PC como fuente del bus A Selecciona RCON2 (contenido = +1) como fuente del bus B Selecciona BuffferA Operación de suma No desplaza Habilita el bus C para carga de registro Selecciona almacenar en PC No bifurca	0610 6000 0000
MAR ← PC; RD = 1	B = 0000 MAR = 1 RD = 1 COND = 000	Selecciona PC como fuente del bus Carga MAR con la dirección indicada por PC Inicio de la operación de lectura No bifurca	00C0 0000 0000
RD = 1; MBR ← M[MAR]; IR ← MBR	RD = 1 AMUX = 1 ALU = 0010 SH = 000 ENC = 1 C = 00 1 COND = 000	Fin de la operación de lectura. MBR es válido Selecciona MBR como entrada a la UAL Modo transparente No desplaza Habilita el bus C para carga de registros Se carga el registro IR con la salida de SH No bifurca	8053 0000 0400
MPC ← IR[cod_op]	FIR = 1	Salto a la dirección de la tabla de saltos correspondiente	0000 0000 2000

4.2. Decodificación



4.3. Búsqueda de los operandos, ejecución, almacenamiento de resultados y eterminación de la dirección de la siguiente instrucción máquina

Microinstrucción	Señales	Microoperaciones	µI en hexadecimal
AC ← IR[operando]	A = 0011 B = 0100 AMUX = 0 ALU = 0001 SH = 000 ENC = 1 C = 0001 COND = 100 ADDR = 1020_{10}	Selecciona IR como fuente del bus A Selecciona OMASK como fuente del bus B Selecciona BufferA como entrada a la ALU Operación AND No desplaza Habilita el bus C para carga de registros Se carga el registro AC con la salida de SH Salto incondicional A la dirección 1020_{10}	4011 43FF 0200

Microinstrucción	Señales	Microoperaciones	µI en hexadecimal
SP ← SP - 1	A = 0010 B = 0111 AMUX = 0 ALU = 0000 SH = 000 ENC = 1 C = 0010 COND = 000	Selecciona registro SP como fuente del bus A Selecciona registro RCON3 como fuente del bus B Selecciona BufferA como entrada a la ALU Operación suma No desplaza Habilita bus C para carga de registros Se carga el registro SP con la salida de SH No bifurca	0012 7200 0000
MAR ← SP	B = 0010 MAR = 1	Selecciona SP como fuente del bus B Carga registro MAR con la dirección indicada en SP	0080 2000 0000
MBR ← AC	A = 0001 AMUX = 0 ALU = 0010 SH = 000 MBR = 1	Selecciona registro AC como fuente del bus A Selecciona bufferA como entrada a la ALU Modo transparente No desplaza Carga registro MBR con la salida de SH	0100 0100 0400
MP[SP] ← AC	WR = 1	Primer ciclo de escritura (iniciar)	0020 0000 0000
MP[SP] ← AC	WR = 1 COND = 100 ADDR = 1020_{10}	Segundo ciclo de escritura (finalizar) Salto incondicional A la dirección 1020_{10}	4020 00FF 0000

5. Repertorio de Instrucciones de SIMPLE2

Binario	Mnemotécnico	Instrucción	Significado
	Carga-Almacen.		
00001xxxxxxxxxx	LODD	Carga directa	AC ← m[x]
00010xxxxxxxxxx	LODI	Carga inmediata	AC ← x
00011xxxxxxxxxx	STOD	Almacenamiento directo	m[x] ← AC
	Aritméticas		
00100xxxxxxxxxx	ADDD	Suma directa	AC ← AC + m[x]
00101xxxxxxxxxx	ADDI	Suma inmediata	AC ← AC + x
00110xxxxxxxxxx	SUBD	Resta directa	AC ← AC - m[x]
00111xxxxxxxxxx	SUBI	Resta inmediata	AC ← AC – x
	Manejo Pila		
01000xxxxxxxxxx	PUSH	Apilar	SP ← SP - 1 m[SP] ← AC
01001xxxxxxxxxx	POP	Desapilar	AC ← m[SP] SP ← SP + 1
	Saltos		
01010xxxxxxxxxx	JNEG	Salta si negativo	if AC < 0 then PC ← x
01011xxxxxxxxxx	JZER	Salta si cero	if AC = 0 then PC ← x
01100xxxxxxxxxx	JCAR	Salta si acarreo	if C =1 then PC ← x
01101xxxxxxxxxx	JUMP	Salto incondicional	PC ← x
01110xxxxxxxxxx	CALL	Subrutina	SP ← SP - 1 m[SP] ← PC PC ← x
01111xxxxxxxxxx	RETN	Vuelta subrutina	PC ← m[SP] SP ← SP + 1
	Lógicas		
10000xxxxxxxxxx	AND	And lógica directa	AC ← AC (and) m[x]
10001xxxxxxxxxx	ANDI	And lógica inmediata	AC ← AC (and) x
10010xxxxxxxxxx	OR	Or lógica directa	AC ← AC (or) m[x]
10011xxxxxxxxxx	ORI	Or lógica inmediata	AC ← AC (or) x
	Parada		
11111xxxxxxxxxx	HALT	Parada del programa	

6. Ejercicio propuesto

Completar mediante microprogramación, el repertorio de instrucciones propuesto en la sección (al menos una instrucción de cada grupo) , de manera que podamos ejecutar un programa propuesto como ejemplo por el alumno en el simulador gráfico de SIMPLE2.

En la Web esta disponible el enlace con la plantilla modelo para microprogramar.

(una descripción más detallada del funcionamiento y forma de operar con el simulador se puede encontrar en la ayuda que acompaña al programa simulador).