

PRUEBA DE EVALUACIÓN A DISTANCIA 2

Nombre: Salvador Moreno Sánchez

DNI: 20077477H

Centro: Las Tablas - Madrid

PRUEBA DE EVALUACIÓN A DISTANCIA 2-1

1.- (2.5 ptos)

Los circuitos de las figuras siguientes representan sumadores a los que se está aplicando un conjunto de impulsos en sus entradas. El de la Figura 1a representa un semisumador binario (*SSB*) y el de la Figura 1b un sumador binario completo (*SBC*).

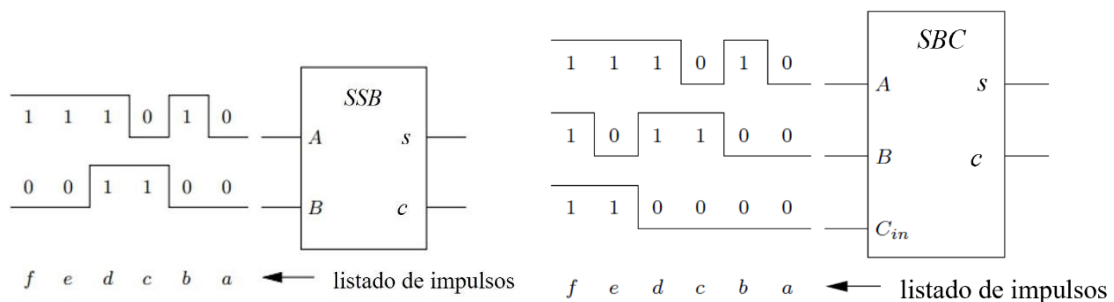


Figura 1a

Figura 1b

- Calcular los distintos valores de las salidas de suma s y acarreo c que se obtienen en la Figura 1a.
- Calcular los distintos valores de las salidas de acarreo s y c que se obtienen en la Figura 1b.
- Si las entradas de un sumador completo con anticipación de arrastre son: $A = 0$ y $B = 1$. Determinar el acarreo generado g y el acarreo propagado p .

a)

Impulsos \ Salidas	A	B	s	c
a	0	0	0	0
b	1	0	1	0
c	0	1	1	0
d	1	1	0	1
e	1	0	1	0
f	1	0	1	0

Tabla 1. Salidas que se obtienen en el SSB de la Figura 1a

b)

Impulsos \ Salidas	A	B	C_{in}	s	c
a	0	0	0	0	0
b	1	0	0	1	0
c	0	1	0	1	0
d	1	1	0	0	1
e	1	0	1	0	1
f	1	1	1	1	1

Tabla 2. Salidas que se obtienen en el SBC de la Figura 1b

c)

Al ser una suma de dos bits en un sumador completo con anticipación de arrastre, con la realización de la misma podemos intuir rápidamente tanto el acarreo generado como el propagado:

$$A + B + C_{in} = 0 + 1 + 0 = 1$$

Sin embargo, y antes de dar una respuesta, acudamos a la teoría, en la que para que haya acarreo generado se debe cumplir la condición de que los dos bits de entrada sean 1, respondiendo a la siguiente fórmula:

$$g = A \cdot B = 0 \cdot 1 = 0$$

Por otro lado, para que exista acarreo propagado, se debe cumplir la condición de que uno de los dos bits de entrada sea 1, respondiendo a la siguiente fórmula:

$$p = A + B = 0 + 1 = 1$$

Así, el acarreo generado (g) es 0 y el propagado (p) es 1.

PRUEBA DE EVALUACIÓN A DISTANCIA 2-2

2.- (2.5 pts) El siguiente algoritmo describe una determinada operación de un sistema digital:

```

1: Declaración de registros       $A [8], B [8], R [8], I [1];$ 
2: Declaración de buses        Bus-Entrada [8], Bus-Salida[8];
3: Inicio:      if  $I = 0$  then go to Inicio;
4:               $A \leftarrow$  Bus-Entrada;
5:               $B \leftarrow$  Bus-Entrada;
6:               $B \leftarrow A - B;$ 
7:              if  $B [8] = 1$  then go to Salto;
8:               $B \leftarrow B + 3;$ 
9:               $B \leftarrow C1(B);$ 
10:             go to Fin;
11: Salto:       $A \leftarrow C2(A);$ 
12: Fin:         $R \leftarrow A - B;$ 
13:             Bus-Salida  $\leftarrow R;$ 
14:             go to Inicio;
```

Donde I es el valor de una señal externa que se activa ($I = 1$) para que el algoritmo comience su ejecución, y se desactiva ($I = 0$) inmediatamente después. $B [8]$ representa el bit más significativo del registro B .

Si los valores iniciales cargados en los registros A y B desde el Bus de Entrada son: $A = (1\ 0\ 1\ 0\ 1\ 0\ 1\ 1)$ y $B = (0\ 1\ 0\ 0\ 1\ 0\ 0\ 1)$, determinar los valores de A , B y R después de la ejecución del algoritmo. Nota $C1$ denota el complemento a 1 y $C2$ el complemento a 2

Respuesta

Cuando el algoritmo se inicia con la señal externa $I = 1$, el registro A , desde el bus de entrada, se carga con el valor dado de 10101011 y, en la siguiente instrucción sucede lo mismo pero con el registro B , que se carga con el binario 01001001. Seguidamente, se procede a realizar la resta entre A y B , cuyo resultado (01100010) es cargado en B . Llegamos a la instrucción donde debemos realizar la comprobación de si el bit más significativo de B es 1 (saltaríamos a la instrucción 11) o no (se seguiría con la siguiente instrucción del algoritmo): como este es 0, no se cumple la condición, por lo que no realizamos el salto indicado y seguimos con la octava instrucción, donde sumamos el decimal 3 (00000011) a B y el resultado (01100101) lo cargamos en este. A continuación, efectuamos el complemento a 1 de B , que sería 10011010 y lo volvemos a cargar en el registro B . Por último, hacemos la resta de A menos B y el resultado (00010001) lo cargamos en R para que luego este sea llevado al bus de salida.

Por tanto, podemos enunciar que los valores de los registros después de la ejecución del algoritmo son:

- **A = 10101011**
- **B = 10011010**
- **R = 00010001**

En la Tabla 3 podemos observar un resumen de la evolución de los registros A, B y R a lo largo del algoritmo dado.

Nº de instrucción	A	B	R
4 : $A \leftarrow \text{Bus} - \text{Entrada}$	10101011	--	--
5 : $B \leftarrow \text{Bus} - \text{Entrada}$	10101011	01001001	--
6 : $B \leftarrow A - B$	10101011	01100010	--
7: if $B[8] = 1$ then go to Salto	10101011	<u>0</u> 1100010	--
8 : $B \leftarrow B + 3$	10101011	01100101	--
9 : $B \leftarrow C1(B)$	10101011	10011010	--
12 : $R \leftarrow A - B$	10101011	10011010	00010001

Tabla 3. Evolución de los registros a lo largo de la ejecución del algoritmo a estudiar

PRUEBA DE EVALUACIÓN A DISTANCIA 2-3

3.- (2.5 ptos) Considérese el algoritmo del ejercicio anterior (ejercicio 2, PED 2-2) .

- a) Dibuje un esquema de una Unidad de Procesamiento que permita realizar este algoritmo, utilizando los recursos que considere necesarios. **Nota:** Asúmase las operaciones de suma, resta, complemento a 1 y complemento 2 se realizan mediante una *ALU*.
- b) Haga la representación de un diagrama de estados de una unidad de control que ejecute dicho algoritmo para la unidad de procesamiento diseñada en el apartado a).
- c) Indique el número de estados, señales externas, señales de condición y señales de control necesarias para todo ello, y justifique la respuesta.

Respuesta

a)

Para diseñar nuestra Unidad de Procesamiento, se ha comenzado por determinar el tipo y número de componentes necesarios. Dichos componentes son:

- Un registro de 8 bits para el almacenamiento de A.
- Un registro de 8 bits para el almacenamiento de B.
- Un registro de 8 bits para el almacenamiento de R.

En la Tabla 4 podemos observar que se ha escogido un tipo de registros que permiten el desplazamiento, además de la carga, por ello necesitan 2 bits como señales de control (c_1c_0). Aunque para satisfacer nuestro algoritmo no se precisa de desplazamientos, por lo que sólo podríamos haber elegido un tipo de registro con una sola señal de control (c_0) para representar carga o no carga, se ha preferido emplear este tipo de registros por su uso más extendido en la teoría y en los ejercicios del libro de la asignatura.

c_1c_0	$Q(t + 1)$
0 0	$Q(t)$
0 1	Despl. Dcha.
1 0	Despl. Izq.
1 1	Carga

Tabla 4. Módulo elegido para regir los registros a usar

- Una ALU que realice la suma y la resta de dos números binarios de 8 bits, además del complemento a 1 y el complemento a 2.

c_1c_0	Salida
0 0	Suma
0 1	Resta
1 0	C1(B)
1 1	C2(A)

Tabla 5. Módulo elegido para regir la ALU a usar

- Un multiplexor de dos salidas para seleccionar la carga del registro A. A este registro se le pueden cargar datos desde el bus (\bar{c}_0) o desde la ALU (c_0).
- Un multiplexor de dos salidas para seleccionar la carga del registro B. A este registro se le pueden cargar datos desde el bus (\bar{c}_0) o desde la ALU (c_0).
- Un multiplexor de dos salidas para seleccionar entre el contenido del registro A (\bar{c}_0) o el

decimal 3 en binario de 8 bits (c_0) en pos de decidir una de las entradas de la ALU.

- El registro R debe volcar su contenido al bus de salida, por lo que se debe utilizar una puerta triestado de conexión unidireccional con control de 8 bits.

c_0	Salida
0	-
1	Entrada

Tabla 6. Módulo elegido para regir el triestado a usar

- Una puerta AND, que reciba como entradas dos bits: un 1 y el bit más significativo del registro B ($B[8] = B_7$), por lo que la salida de la puerta será 1 en el caso de que el bit más significativo del registro B sea 1, siendo la única condición (señal de condición s_0) que debemos tener en cuenta en nuestro algoritmo.

La Figura 1 muestra una posible Unidad de Procesamiento que puede ejecutar el algoritmo propuesto.

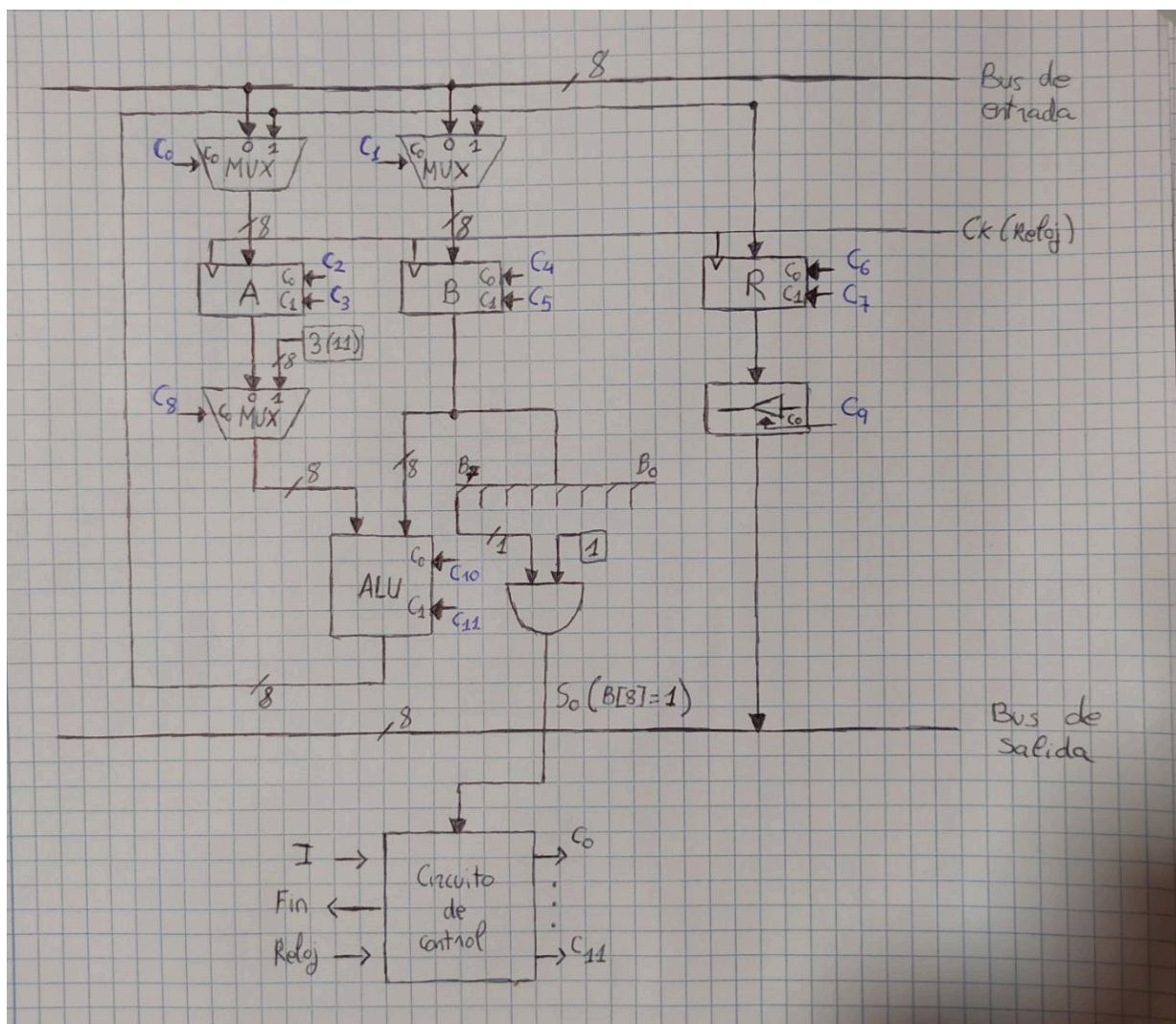


Figura 1. Esquema de la Unidad de Procesamiento asociada al algoritmo dado (enlazada con la Unidad de Control)

b)

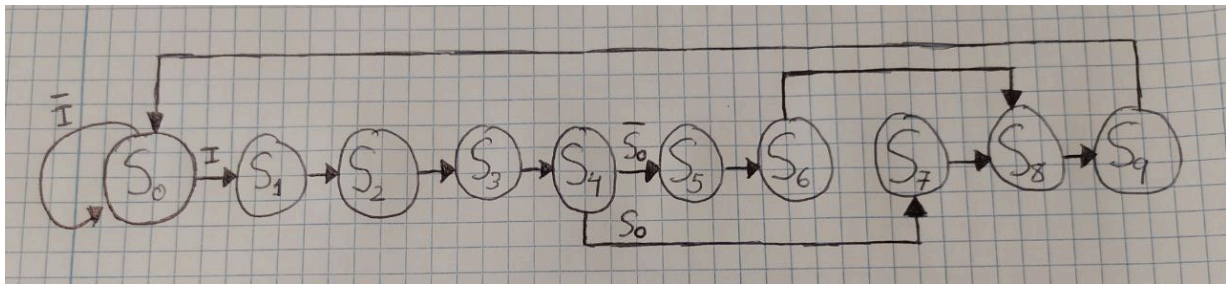


Figura 2. Diagrama de estados de la Unidad de Control que ejecuta el algoritmo dado para la Unidad de Procesamiento diseñada

c)

Como observamos en la Figura 2, el diagrama tiene 10 estados. Se comienza en el estado S_0 , estado en el que sólo se comprueba si la señal externa I que entra en nuestro circuito de control está activada (1) o no (0). Si recibe un 0, nos mantendríamos en el estado S_0 ; en caso contrario, se pasa al estado S_1 , en el que se transfiere el contenido del bus de entrada al registro A. Seguidamente, se pasa al estado S_2 , en el que se transfiere el contenido del bus de entrada al registro B. A continuación, se pasa al estado S_3 , donde se realiza la operación de resta de los registros A y B, cuyo resultado se almacena en el registro B. Llegamos al estado S_4 , donde sólo se va a comprobar si el bit más significativo del registro B es 1 o 0. Este condicionante (s_0) nos va a presentar dos escenarios:

- En el caso de que sea 1: se pasa al estado S_7 , en el que se realizaría el complemento a 2 del registro A para almacenarlo también en A. Luego, se pasaría al estado S_8 , donde se realiza la resta de los registros A y B para almacenar el resultado en el registro R; para, por último, pasar al estado S_9 , en el que se produce la transferencia desde el registro R al bus de salida y se vuelve al estado S_0 .
- En el caso de que sea 0: se pasa al estado S_5 , en el que se realizaría la suma del registro B y el decimal 3. A continuación, se pasaría al estado S_6 , donde se haría el complemento a 1 del registro B para almacenarlo también en B y, seguidamente, desde el estado actual S_6 se pasa directamente al estado S_8 , en el que se realiza la resta de los registros A y B para almacenar el resultado en el registro R; para, por último, pasar al estado S_9 , en el que se produce la transferencia desde el registro R al bus de salida y se vuelve al estado S_0 .

Respecto a las señales de condición, y como se observa en la Figura 1, nuestra Unidad de Procesamiento posee una salida que sirve de entrada al circuito de control de la Unidad de Control. Se trata de la señal de condición s_0 , la cual valdrá 1 cuando el bit más significativo del registro B sea 1. Como ya hemos comentado, esta condición entrará en juego en nuestra Unidad de Control una vez que se esté en el estado S_4 .

Por otro lado, vemos en el esquema de nuestra Unidad de Procesamiento de la Figura 1 que, en color azul, se especifican las señales de control. En la Tabla 7 se describen las 12

señales de control necesarias junto con la función que debe realizar cada una de ellas.

Señal de control	Operación que controla
c_0	Selección del Bus (0) o ALU (1)
c_1	Selección del Bus (0) o ALU (1)
c_3, c_2	Control del registro de desplazamiento A
c_5, c_4	Control del registro de desplazamiento B
c_7, c_6	Control del registro de desplazamiento R
c_8	Selección del registro A (0) o decimal 3 (1)
c_9	Si $c_9 = 1$. se realiza la microoperación $\text{Bus-Salida} \leftarrow R$
c_{11}, c_{10}	Control de la ALU: suma (00), resta (01), $C1[B]$ (10), $C2[A]$ (11)

Tabla 7. Señales de control de la Unidad de Procesamiento

Por último, y para poseer una fotografía final y esquemática del empleo de nuestra Unidad de procesamiento y Unidad de Control, en la Tabla 8 se muestran las microoperaciones y las señales de control asociadas a cada uno de los estados que se han definido.

Estado de la Unidad de Control	Microoperaciones realizadas	Señales de control a activar
S_0	--	--
S_1	$A \leftarrow \text{Bus} - \text{Entrada}$	\bar{c}_0, c_2, c_3
S_2	$B \leftarrow \text{Bus} - \text{Entrada}$	\bar{c}_1, c_4, c_5
S_3	$B \leftarrow A - B$	$\bar{c}_8, c_{10}, \bar{c}_{11}, c_1, c_4, c_5$
S_4	--	--
S_5	$B \leftarrow B + 3$	$c_8, \bar{c}_{10}, \bar{c}_{11}, c_1, c_4, c_5$
S_6	$B \leftarrow C1(B)$	$\bar{c}_{10}, c_{11}, c_1, c_4, c_5$
S_7	$A \leftarrow C2(A)$	$c_{10}, c_{11}, c_0, c_2, c_3$
S_8	$R \leftarrow A - B$	$\bar{c}_8, c_{10}, \bar{c}_{11}, c_6, c_7$
S_9	$\text{Bus} - \text{Salida} \leftarrow R$	c_9

Tabla 8. Microoperaciones y señales de control

PRUEBA DE EVALUACIÓN A DISTANCIA 2-4

4.- (2.5 pts) Calcular la máxima memoria direccionable que tiene un computador con una longitud de palabra de 32 bits, con un repertorio de 16 instrucciones distintas y con un banco de registros de 8 registros, en el en los dos casos siguientes:

- a) Si el computador es de una dirección (procesador con acumulador).
- b) Si el computador es de dos direcciones, con uno de los operandos en memoria principal y el otro en un registro del banco de registros.

Nota: Se supone que las instrucciones de operaciones diádicas están definidas en una palabra de memoria.

Respuesta

a)

Al ser un computador de una dirección, la longitud de palabra del mismo va a estar fraccionada en dos partes: aquella destinada a cubrir el repertorio de instrucciones (código operación) y la otra para apuntar directamente a memoria, la cual representaría el único operando e indicaría la capacidad de la memoria direccionable.

Cuantificando lo dicho:

- Tenemos un computador de 32 bits de longitud de palabra.
- Tenemos un repertorio de 16 instrucciones distintas, por lo que vamos a necesitar 4 bits de código operación para poder representar cada uno de ellos ($2^4 = 16$).
- Así, para calcular el operando debemos restar los 32 bits de longitud de palabra del computador menos los 4 bits del código operación, obteniendo 28 bits para apuntar a memoria.
- De esta forma, **la máxima memoria direccionable será de 2^{28} posiciones en memoria.**

b)

En este caso, al ser un computador de dos direcciones, en el que una va a estar destinada a cubrir el banco de registros y la otra la memoria principal; la longitud de palabra del computador va a estar fraccionada en tres partes: aquella para el código operación, tal y como hemos nombrado en el apartado anterior; y las dos direcciones ya mencionadas como operandos, interesándonos por la que va a apuntar a memoria.

Cuantificando lo dicho:

- Tenemos un computador de 32 bits de longitud de palabra.
- Tenemos un repertorio de 16 instrucciones distintas, por lo que vamos a necesitar 4 bits de código operación para poder representar cada uno de ellos ($2^4 = 16$).
- Además, tenemos un banco de registros de 8 registros, por lo que vamos a necesitar 3

bits para tener la capacidad de apuntar a cada uno de ellos ($2^3 = 8$). Este sería un operando.

- Así, para calcular el operando restante debemos restar los 32 bits de longitud de palabra del computador menos los 4 bits del código operación y los 3 bits del operando que destinamos para señalar al banco de registros, obteniendo 25 bits para apuntar a memoria.
- De esta forma, **la máxima memoria direccionable será de 2^{25} posiciones en memoria.**