

Guía de trabajos prácticos de TD2

M. Estefanía Pereyra

17 de abril de 2020

Transferencia de registros y operaciones de la computadora

Ejercicio 1

Realizar el diagrama de bloque del hardware que ejecute la siguiente instrucción de transferencia de registros.

$$T_3 : R_2 \leftarrow R_1, R_1 \leftarrow R_2$$

Ejercicio 2

Las salidas de cuatro registros R_0 , R_1 , R_2 y R_3 están conectadas a través de multiplexores a las entradas de un quinto registro R_5 . Cada registro tiene 8 bits de longitud. Las transferencias requeridas son:

$$T_0 : R_5 \leftarrow R_0$$

$$T_1 : R_5 \leftarrow R_1$$

$$T_2 : R_5 \leftarrow R_2$$

$$T_3 : R_5 \leftarrow R_3$$

Las variables de sincronización, de la T_0 a la T_3 , son mutuamente excluyentes y sólo una puede ser igual a 1 en cualquier momento mientras las otras son igual a 0.

- Realizar el diagrama de bloque del hardware que ejecute las transferencias de registros.
- Incluir conexiones de las señales de control.

- Realizar el diagrama equivalente utilizando buffers de tres estados.

Ejercicio 3

Utilizando dos registros de 4 bits, R_1 y R_2 , un multiplexor cuádruple de 2 a 1 líneas, con habilitación (E) y cuatro inversores, trace el diagrama del circuito que ejecute las siguientes instrucciones.

$$T_1 : R_2 \leftarrow R_1$$

$$T_2 : R_2 \leftarrow \overline{R_2}$$

$$T_3 : R_2 \leftarrow 0$$

Ejercicio 4

Diseñe el hardware que ejecute las siguientes instrucciones de transferencia de registros para los registros de 4 bits, R_1 y R_2 .

$$XT : R_1 \leftarrow R_1 + R_2$$

$$\overline{XT} : R_1 \leftarrow R_2$$

Ejercicio 5

Diseñe un circuito aritmético con una variable de selección S y dos entradas de datos de n bits, A , B . El circuito genera las cuatro operaciones aritméticas que se muestran en la Tabla 2 junto con el acarreo de entrada C_{in} . Trace el diagrama de lógica para las dos primeras etapas del sumador completo (las etapas correspondientes al bit 0 y bit 1).

Tabla 2: Operaciones aritméticas

S	$C_{in} = 0$	$C_{in} = 1$
0	$F = A + B$ (suma)	$F = A + 1$ (incremento)
1	$F = A - 1$ (decremento)	$F = A + \overline{B} + 1$ (sustracción)

Ejercicio 6

Diseñe un circuito aritmético de 4 bits con dos variables de selección S_0 y S_1 que genere las operaciones aritméticas detalladas en la Tabla 3. Trace el diagrama de lógica de una etapa típica.

Tabla 3: Operaciones aritméticas

$S1$	$S0$	$C_{in} = 0$	$C_{in} = 1$
0	0	$F = A + B$ (suma)	$F = A + b + 1$
0	1	$F = A$ (transferencia)	$F = A + 1$ (incremento)
1	0	$F = \bar{B}$ (complemento)	$F = \bar{B} + 1$ (negación)
1	1	$F = A + \bar{B}$	$F = A + \bar{B} + 1$ (sustracción)

Ejercicio 7

Convierta los siguientes números decimales en binarios con signo con 8 bits cada uno (incluyendo al signo). Realice las operaciones con los números binarios con signo y verifique los dos últimos acarrees. Indique si los acarrees señalan un desbordamiento. Recuerde que los números de 8 bits tienen un rango de $[-128, 127]$.

- $(+65) + (+36)$
- $(+65) - (-90)$
- $(-36) + (-90)$
- $(-65) - (+90)$

Ejercicio 8

Realice las operaciones lógicas AND, OR, y XOR con los números de 8 bits 10011100 y 10101010.

Ejercicio 9

Diseñe los circuitos digitales correspondientes que realicen las operaciones lógicas indicadas en la Tabla 4 y en la Tabla 5.

Tabla 4: Operaciones lógicas

$S1$	$S0$	Operación	Función
0	0	$A \wedge B$	$A \text{ AND } B$
0	1	$A \vee B$	$A \text{ OR } B$
1	0	$A \oplus B$	$A \text{ XOR } B$
1	1	\bar{A}	complemento

Tabla 5: Operaciones lógicas

$S1$	$S0$	Operación	Función
0	0	$A \oplus B$	A XOR B
0	1	$\overline{A \oplus B}$	A NOR Excluyente B
1	0	$A \vee B$	A NOR B
1	1	$\overline{A \wedge B}$	A NAND B

Ejercicio 10

Dado el valor de 16 bits $A = 0101101011000011$. ¿Qué operación se debe efectuar

- a) para poner a 0 los últimos 8 bits?
- b) para iniciar a 1 los primeros 8 bits?
- c) para complementar los 8 bits centrales?

Ejercicio 11

Comenzando a partir de los 8 bits $R = 10110010$, demuestre los valores que se obtienen después de cada una de las microoperaciones de corrimiento que se presentan en la Tabla 6.

Tabla 6: Microoperaciones de corrimiento

Designación simbólica	Descripción
$R \leftarrow shlR$	Corrimiento a la izquierda del registro R
$R \leftarrow shrR$	Corrimiento a la derecha del registro R
$R \leftarrow rolR$	Rotación a la izquierda del registro R
$R \leftarrow rorR$	Rotación a la derecha del registro R
$R \leftarrow aslR$	Corrimiento aritmético a la izquierda de R
$R \leftarrow asrR$	Corrimiento aritmético a la derecha de R

Ejercicio 12

Diseñe una unidad de corrimiento de 4 bits utilizando circuitos combinatorios (multiplexores) con dos variables de selección $H0$ y $H1$ que genere las operaciones de corrimiento descritas en la Tabla 7.

Tabla 7: Operaciones de corrimiento

$H1$	$H0$	Operación	Función
0	0	$S \leftarrow F$	Transferencia de F a S (no hay corrimiento)
0	1	$S \leftarrow shlF$	Corrimiento a la izquierda de F a S
1	0	$S \leftarrow shrF$	Corrimiento a la derecha de F a S
1	1	$S \leftarrow 0$	Transferencia de ceros a S

Ejercicio 13

Diseñe una unidad de corrimiento de 4 bits de forma tal que extienda las operaciones descritas en la Tabla 7 incorporando las operaciones de rotación que se describen en la Tabla 8, para tal fin agregar una variable al campo H de selección. Es decir, las señales de selección de las operaciones serán H_2 , H_1 y H_0 .

Tabla 8: Operaciones de corrimiento

$H2$	$H1$	$H0$	Operación	Función
0	0	0	$S \leftarrow F$	Transferencia de F a S (no hay corrimiento)
0	0	1	$S \leftarrow shlF$	Corrimiento a la izquierda de F a S
0	1	0	$S \leftarrow shrF$	Corrimiento a la derecha de F a S
0	1	1	$S \leftarrow 0$	Transferencia de ceros a S
1	0	0	—	—
1	0	1	$S \leftarrow rolF$	Rotación a la izquierda de F a S
1	1	0	$S \leftarrow rorF$	Rotación a la derecha de F a S
1	1	1	—	—

Ejercicio 14

Una unidad procesadora (como la de la figura 7-11 del libro) tiene 30 registros. ¿Cuántas líneas de selección se necesitan para cada conjunto de multiplexores y para el decodificador?

Ejercicio 15

Dada una ALU de 8 bits con salidas de la F_0 a la F_7 y acarrees C_7 y C_8 , muestre el circuito lógico para iniciar los cuatro bits de estado, C (acarreo), V (desbordamiento), Z (cero) y S (signo).

Ejercicio 16

Especifique la palabra de control que se debe aplicar al procesador para ejecutar las siguientes microoperaciones:

Tabla 9: Microoperaciones

(a)	$R_2 \leftarrow R_1 + 1$	(e)	$R_1 \leftarrow shl R_1$
(b)	$R_3 \leftarrow R_4 + R_5$	(f)	$R_2 \leftarrow ror R_2$
(c)	$R_6 \leftarrow \overline{R_6}$	(g)	$R_5 \leftarrow R_3 + R_1$
(d)	$R_7 \leftarrow R_7 - 1$	(h)	$R_6 \leftarrow R_7$

Ejercicio 17

Dada las siguientes palabras de control de 16 *bits* de la unidad procesadora, determine la microoperación que se ejecuta con cada palabra.

Tabla 11: Palabra de control

(a)	0010100110101000	(d)	0000010001000000
(b)	1100000010000001	(e)	1110000110000110
(c)	0100100101100000		

Solución del ejercicio 1

La instrucción condicional $T_3 : R_2 \leftarrow R_1, R_1 \leftarrow R_2$, denota una transferencia del contenido del registro R_1 al registro R_2 y a la vez una transferencia del contenido del registro R_2 al registro R_1 . Es decir, la operación genera un intercambio del contenido entre los registros. Esta operación está condicionada por la señal de sincronización T_3 generada por la sección de control del sistema digital.

Toda microoperación realizada por el sistema digital se encuentra sincronizada con el reloj del sistema. En la Fig.1 se muestra el diagrama de sincronización correspondiente a esta operación. En un instante de tiempo inicial $t = t_0$, previo a la ocurrencia de la señal T_3 , el contenido del registro R_1 se supone igual a d_1 y el contenido del registro R_2 se supone igual a d_2 . Luego en el instante de tiempo $t = t_1$, con el flanco ascendente del pulso de reloj se activa la señal de sincronización T_3 . En la siguiente transición positiva del reloj, $t = t_2$, la señal de sincronización se encuentra activa ($T_3 = 1$) con lo cual los datos en las entradas de R_1 (el contenido a la salida del registro R_2 , d_2) se cargan en R_1 , y los datos en las entradas de R_2 (el contenido a la salida del registro R_1 , d_1) se cargan en R_2 .

Esta operación simultánea es posible con registros que tengan flip-flops activados por flanco. En la Fig.2 se muestra el diagrama de bloque que representa la transferencia de R_1 a R_2 y de R_2 a R_1 .

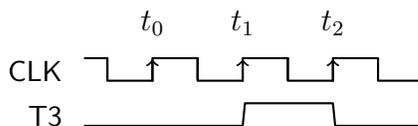


Figura 1: Diagrama de sincronización

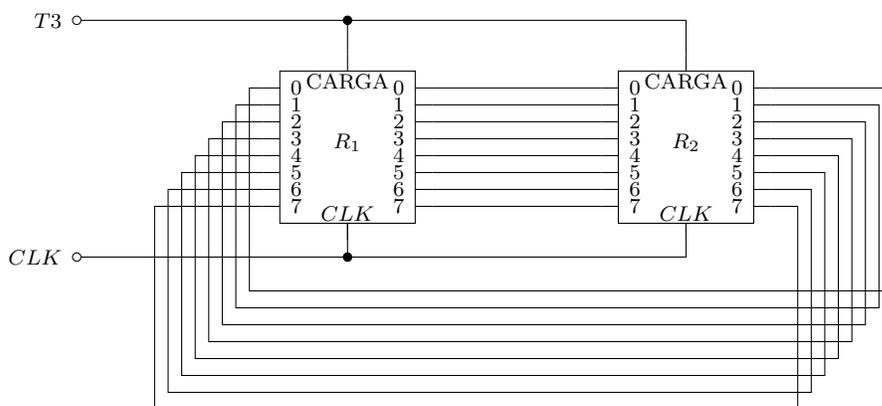


Figura 2: Diagrama del hardware.

Solución del ejercicio 2

El registro R_5 recibe información de cuatro fuentes diferentes en tiempos distintos. Este tipo de operación requiere de multiplexores para seleccionar entre los registros fuentes de acuerdo con los valores de las variables de sincronización.

La activación de cualquiera de las señales de sincronización esta condicionada a la activación de una única señal a la vez pero no tiene restricciones en el orden en que se activan.

Como en el ejercicio anterior, las señales de sincronización se activan con el flanco ascendente de un ciclo de reloj y en el siguiente flanco ascendente de un ciclo se produce la carga del registro correspondiente.

En la Fig.4 se presenta el diagrama de bloques del hardware que realiza las operaciones de transferencia entre registros. Se utilizan tantos multiplexores como número de bits contengan los registros, con registros de $8bits$ se necesitan 8 multiplexores. Cada multiplexor tendrá tantas entradas como número de registros fuente existan, en este caso son cuatro registros fuente $R_1 - R_4$. Entonces tendremos multiplexores de 4 entradas y 1 salida, con lo cual necesitamos dos señales de selección $S1$ y $S0$.

La selección del registro fuente se hace por medio de un circuito codificador que realiza la lógica descrita en la tabla que se muestra en la Tabla 1.

En la Fig.3 se muestra a modo de ejemplo el diagrama de sincronismo correspondiente a la activación de $T0$, $T2$, $T1$ y $T3$. En este ejemplo, para $t = t_0$ se activa la señal de sincronización $T2$, en ese instante el codificador genera la salida $AB = 10$ y todos los multiplexores seleccionan la entrada I_2 , como consecuencia la entrada del registro R_5 contiene el dato del registro R_2 . Este dato se carga en R_5 en el próximo flanco ascendente del reloj en $t = t_1$.

Queda para el alumno el diseño del diagrama equivalente utilizando buffers de tres estados.

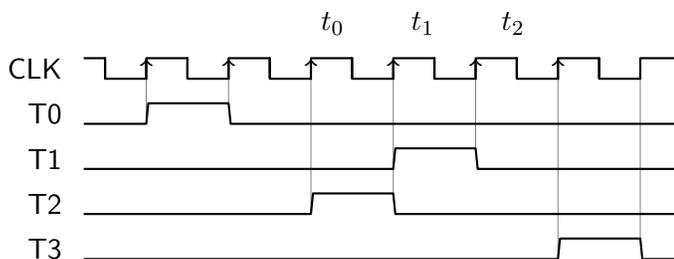


Figura 3: Diagrama de sincronización.

Tabla 1: Lógica de control

T0	T1	T2	T3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

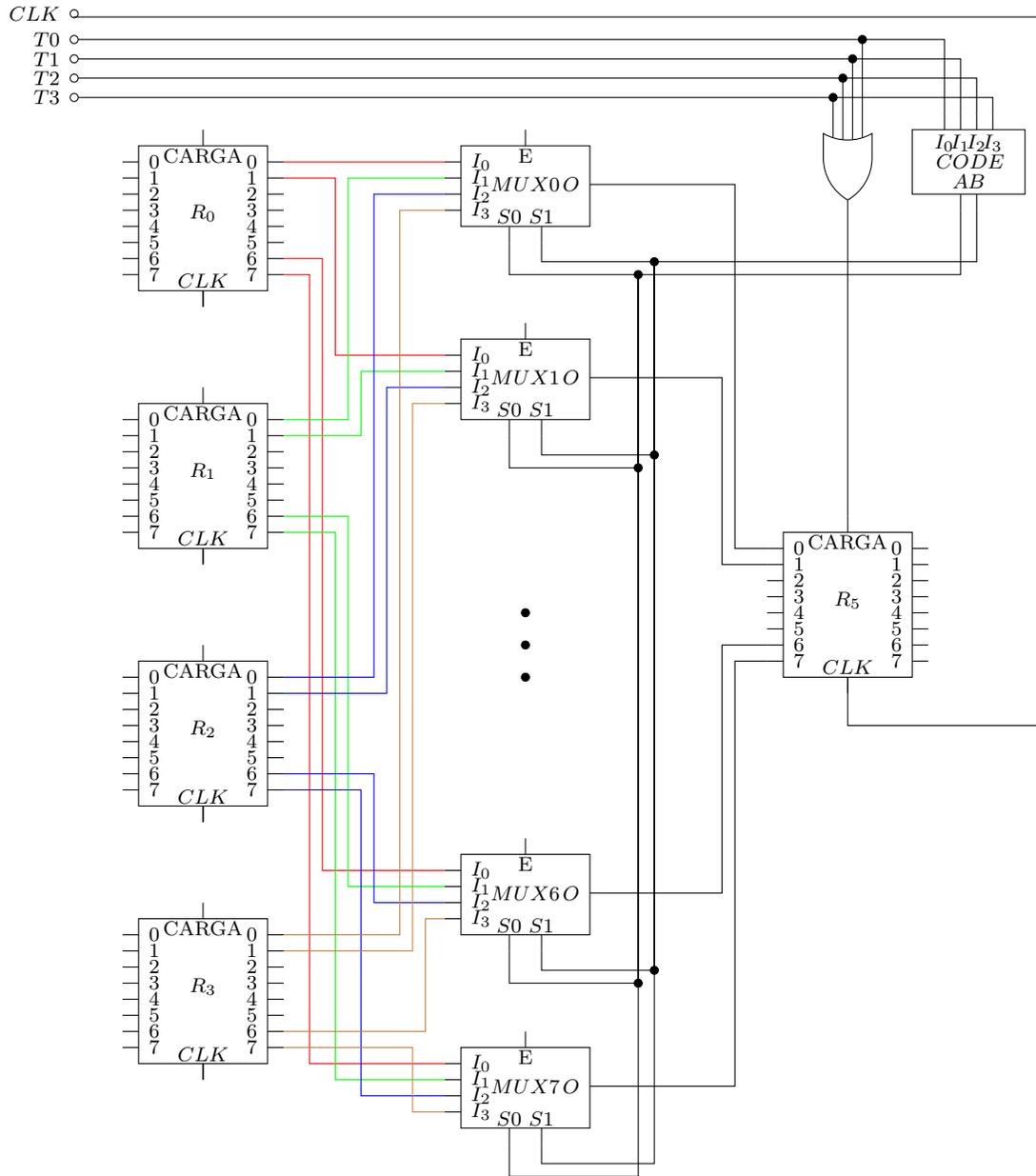


Figura 4: Diagrama del hardware.

Solución del ejercicio 3

En este caso el registro R_2 recibe información de tres fuentes diferentes en tiempos distintos. Cuando $T_1 = 1$ se carga con el contenido de R_1 , con $T_2 = 1$ se carga con su propio contenido negado (complemento a 1) y con $T_3 = 1$ se carga con ceros. Los registros son de 4 bits, por lo tanto se utilizará un multiplexor cuádruple de 2×1 para seleccionar entre las configuraciones 1 y 2. La configuración 3 se genera deshabilitando el multiplexor. Supondremos que las salidas del multiplexor son iguales a cero cuando se deshabilita (si fueran salidas con alta impedancia se debe agrega una resistencia pull down a cada salida con el fin de generar esa configuración).

Bajo estas condiciones, la carga de datos al registro R_2 se debe realizar con la ocurrencia de cualquiera de las tres señales de sincronización. El multiplexor se debe habilitar con la ocurrencia de T_1 o T_2 . Ambas situaciones se resuelve mediante el uso de compuertas *OR* como se muestra en el diagrama del hardware de la Fig.5. Luego, la selección entre las configuraciones 1 y 2 se realiza conectando la señal T_2 a la entrada de selección del multiplexor.

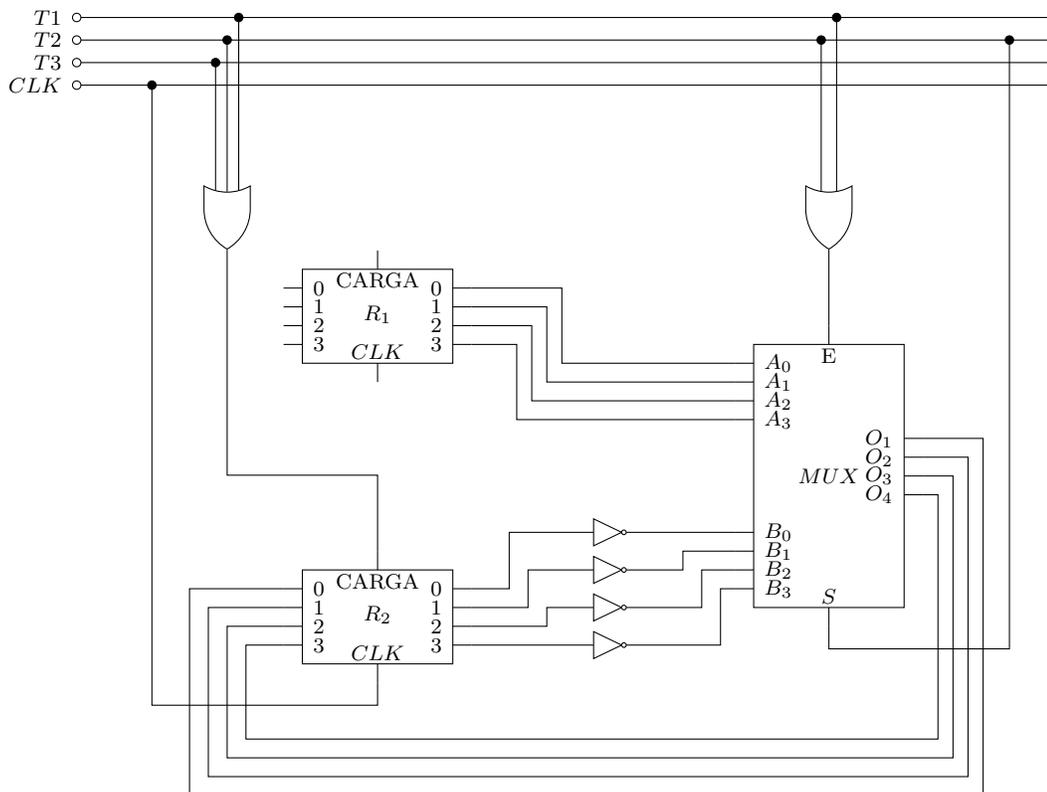


Figura 5: Diagrama del hardware.

Solución del ejercicio 5

De acuerdo con la Tabla 2, la lógica de control propuesta deberá seleccionar una de las cuatro operaciones requeridas dependiendo del valor de las señales S y C_{in} (en este caso el acarreo de entrada además de formar parte de la operación aritmética se utiliza para seleccionar la operación a ejecutar).

Una posible solución es diseñar la lógica de control utilizando un multiplexor de cuatro entradas y una salida, cuyas señales de control sean S y C_{in} . Esta solución se muestra en la Fig.6.

En el diagrama se muestran las dos primeras etapas del sumador de n bits. Para completar el sumador de n bits deben concatenarse las $(n - 2)$ etapas restantes, es decir, la salida C_{out} de la etapa 2 se conecta a la entrada C_{in} de la etapa 3 y así con las etapas restantes.

Otra solución puede plantearse diseñando la lógica de control utilizando circuito combinacionales.

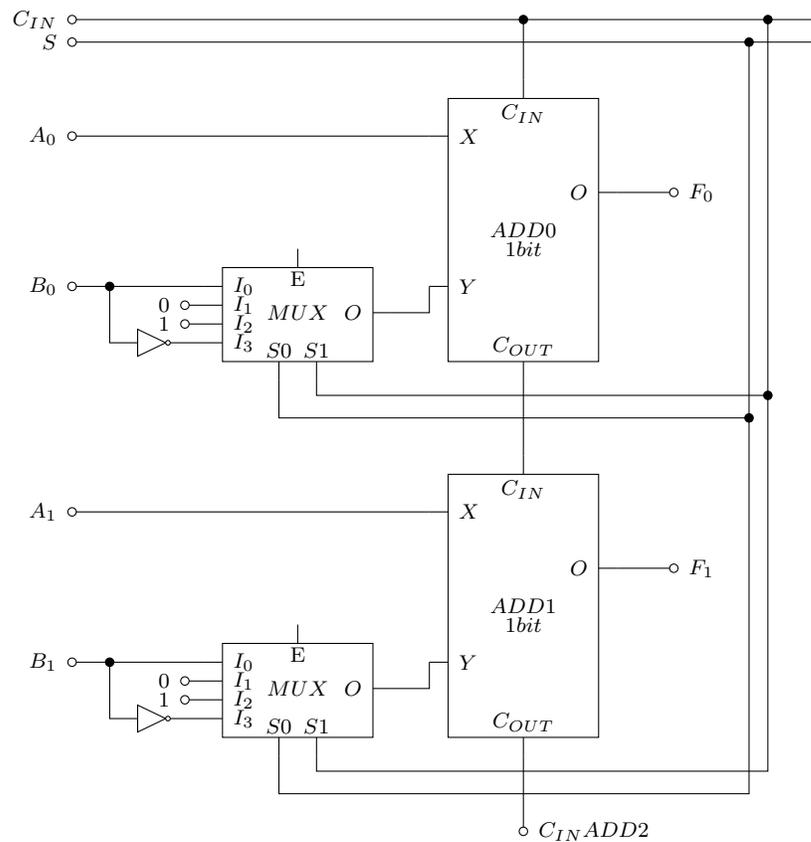


Figura 6: Primeras dos etapas del circuito aritmético correspondiente a la Tabla 2.

Solución del ejercicio 9

La Fig.7 muestra los circuitos que realizan las operaciones lógicas descritas en la Tabla 4 y en la Tabla 5. El circuito de la izquierda corresponde a una etapa del circuito lógico de n bits que realiza las operaciones lógicas de la Tabla 4 y el circuito de la derecha corresponde a una etapa del circuito lógico que realiza las operaciones lógicas de la Tabla 5. En ambos casos se utiliza un multiplexor de 4 entradas y 1 salida que selecciona la operación que se realiza entre los registros a nivel de bits y una serie de compuertas que ejecutan la operación seleccionada.

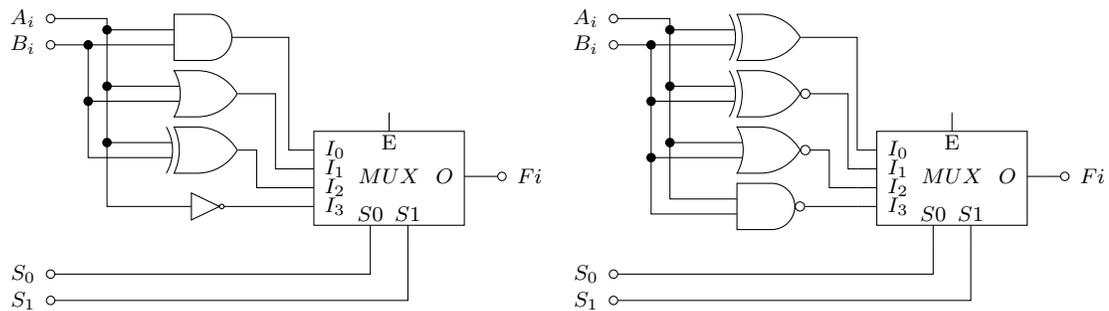


Figura 7: Circuitos lógicos correspondientes a los apartados a) y b) del ejercicio 7.

Solución del ejercicio 12

La Fig.8 muestra el diagrama en bloques de una unidad de corrimiento de 4 bits diseñada utilizando multiplexores de cuatro entradas y una salida, conectados de forma tal que generen las operaciones que se detallan en la Tabla 7, controlados por las señales H_0 y H_1 .

En rojo se indican las conexiones correspondientes a un desplazamiento a la izquierda. Donde I_L indica la conexión de entrada serie y O_L la conexión de salida serie.

De la misma forma, en verde se indican las conexiones necesarias para generar un desplazamiento del contenido de los registros hacia la derecha. Donde I_R indica la conexión de entrada serie y O_R la conexión de salida serie.

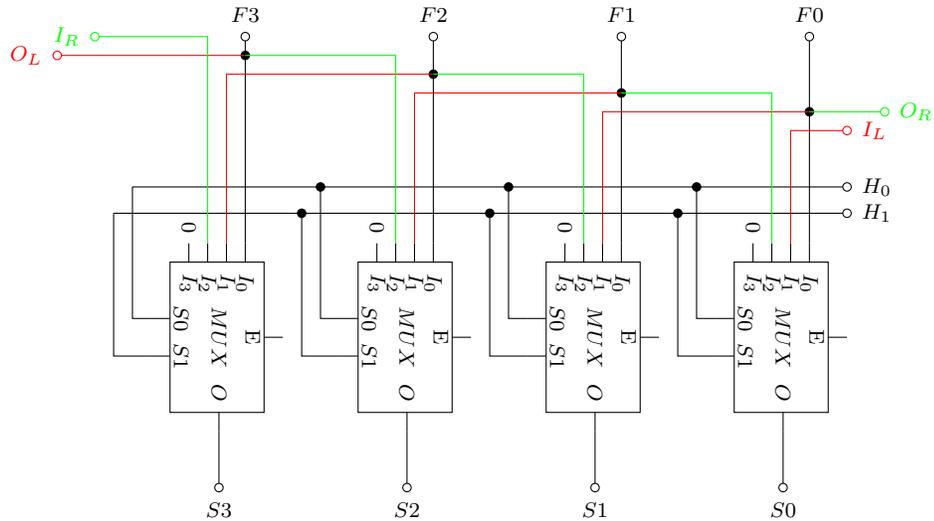


Figura 8: Unidad de corrimiento de 4 bits correspondiente a la Tabla 7.

Solución del ejercicio 16

La palabra de control es una cadena finita de bits que son manejados como un conjunto por el sistema. En el modelo de la unidad procesadora presentada en el teórico se vio que existen 16 entradas de selección binarias en la unidad. Estas 16 entradas definen la longitud de la palabra de control, cuyo valor determina qué operación debe realizar la unidad procesadora.

La palabra de control definida de esta manera consta de cinco partes llamadas campos, donde cada campo está designado por una letra. El campo A, es un campo de tres bits que codifica el registro fuente para una entrada de la ALU. El campo B, también es un campo de tres bits y codifica el registro fuente para la otra entrada de la ALU. El campo D, es de tres bits y selecciona el registro destino. El campo F, es el único campo de cuatro bits y codifica una de las 12 operaciones que puede realizar la ALU. El campo H, es de tres bits y selecciona el tipo de corrimiento que realizará la unidad de corrimiento.

De esta forma, toda microoperación puede codificarse en una única palabra de control que luego será ejecutada por el hardware correspondiente.

A	B	D	F	H
Selecc. del bus A	Selecc. del bus B	Selecc. del destino	Selecc. de op. de la ALU	Selecc. del tipo de corrimiento

Tabla 10: Microoperación - Palabras de control

Microoperación	Designación simbólica					Palabra de control				
	A	B	D	F	H	A	B	D	F	H
$R_2 \leftarrow R_1 + 1$	R_1	–	R_2	$A + 1$	s/corr	001	000	010	0001	000
$R_3 \leftarrow R_4 + R_5$	R_4	R_5	R_3	$A + B$	s/corr	100	101	011	0010	000
$R_6 \leftarrow \overline{R_6}$	R_6	–	R_6	\bar{A}	s/corr	110	000	110	1110	000
$R_7 \leftarrow R_7 - 1$	R_7	–	R_7	$A - 1$	s/corr	111	000	110	0110	000
$R_1 \leftarrow shl R_1$	R_1	–	R_1	A	<i>shl</i>	001	000	001	0000	001
$R_2 \leftarrow ror R_2$	R_2	–	R_2	A	<i>ror</i>	010	000	010	0000	110
$R_5 \leftarrow R_3 + R_1$	R_3	R_1	R_5	$A + B$	s/corr	011	001	101	0010	000
$R_6 \leftarrow R_7$	R_7	–	R_6	A	s/corr	111	000	110	0000	000