



1. Dibujar un posible diagrama de hardware que implementa la siguiente operación de transferencia entre registros:

$$C_3: R2 \leftarrow R1, R1 \leftarrow R2$$

2. Las salidas de los registros R0, R1, R2 y R3 están conectadas a través de un MUX 4 a 1 a las entradas de un registro R4. Cada registro consta de 8 bits. Las operaciones de transferencia que se quieren implementar según 4 variables de control son:

$$C_0: R4 \leftarrow R0$$

$$C_1: R4 \leftarrow R1$$

$$C_2: R4 \leftarrow R2$$

$$C_3: R4 \leftarrow R3$$

Las variables de control son mutuamente excluyentes, es decir, en cada instante como mucho 1 variable puede tomar el valor “1” mientras que las 3 restantes están a “0”. Por otro lado, cuando las 4 variables de control se encuentran a “0” no se produce ninguna transferencia a R4 y los registros deben mantener su contenido.

Diseñar el sistema de transferencia:

- (a) Utilizando registros y 1 multiplexor, dibujar un esquema lógico detallado del hardware que implementa el bit i de estas transferencias entre registros.
 - (b) Dibujar el esquema de puertas lógicas que convierten las variables de control C_0, C_1, C_2 y C_3 en las entradas a los elementos del apartado (a) que controlan la transferencia entre registros.
3. Utilizando 2 registros de 4 bits: R1 y R2 y las puertas AND, OR e inversores que sean necesarias, dibujar el diagrama lógico del bit i de los registros sobre los que se pueden realizar las siguientes operaciones:

$$C_0: R2 \leftarrow 0$$

$$C_1: R2 \leftarrow R1$$

$$C_2: R2 \leftarrow \overline{R2}$$

Todas las operaciones deben ser síncronas. Las variables de control son mutuamente excluyentes. Además no ocurre ninguna transferencia cuando todas las variables de control valen 0

4. Dibujar el diagrama de bloques del hardware que implementa la operación:

$$C_1+C_2: R1 \leftarrow R1+R2$$

Donde R1 y R2 son registros de n bits con entradas de carga (LOAD) y C_1 y C_2 las variables de control. Incluir las puertas lógicas para la función de control. Recordar que el símbolo + designa a la operación OR en una operación lógica o de control pero representa la suma aritmética en una microoperación.

5. Indicar el hardware necesario para la transferencia entre 3 registros R0, R1 y R2 entre los cuales se quieren realizar las siguientes transferencias:

$$C_A: R1 \leftarrow R2, R0 \leftarrow R2$$

$$C_B: R1 \leftarrow R0$$

$$C_C: R0 \leftarrow R1, R2 \leftarrow R0$$

Suponer que las variables de control son mutuamente excluyentes y que si todas están a “0” no se produce ninguna transferencia.

- (a) Utilizar registros y multiplexores dedicados. Dibujar un diagrama lógico detallado del hardware que implementa el bit i de los registros.
- (b) Dibujar el diagrama lógico del elemento que convierte las variables de control en las entradas de selección de los multiplexores y las de carga de los registros.

6. Ejecutar las funciones lógicas AND, OR y XOR a nivel de bit con los operandos de 8 bits 10100101 y 00111001.

7. ¿Qué operación debe realizarse para y qué operando debe ser utilizado para...?
 - Borrar todas las posiciones impares (suponiendo que las posiciones van de 15 a 0 de izquierda a derecha)
 - Poner a set los 4 bits más significativos.
 - Complementar los 8 bits menos significativos

8. Si el contenido de un registro es 01001101, mostrar qué palabra se obtiene tras aplicar las operaciones de desplazamiento indicadas en la ruta de datos de la teoría.

9. Una ruta de datos similar a la de teoría consta de 56 registros. ¿Cuántas líneas de selección se necesitan para cada conjunto de multiplexores y para el decodificador?

10. Diseñar un circuito aritmético con una variable de selección S y 2 entradas de n bits A y B. El circuito debe generar las siguientes 4 operaciones aritméticas teniendo en cuenta además el valor del acarreo de entrada C_{in} .

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

Dibujar el diagrama lógico de las dos cifras menos significativas del circuito aritmético.

11. Diseñar un circuito aritmético de 4 bits con 2 variables de selección S_1 y S_0 que genere las siguientes operaciones aritméticas:

$S_1 S_0$	$C_{in} = 0$	$C_{in} = 1$
0 0	$F = A+B$ (suma)	$F=A+B+1$
0 1	$F=A$ (transf)	$F=A+1$ (incremento)
1 0	$F = \bar{B}$ (complemento)	$F = \bar{B} + 1$ (compl. a dos)
1 1	$F = A + \bar{B}$	$F = A + \bar{B} + 1$ (resta)

Dibujar el diagrama lógico para una etapa i del circuito.

12. Las entradas X_i e Y_i de cada sumador completo en un circuito aritmético tienen la lógica digital especificada por las funciones lógicas:

$$X_i = A_i \quad Y_i = \bar{B}_i S + B_i \bar{C}_{in}$$

donde S es la variable de selección, C_{in} el acarreo de entrada y A_i y B_i las entradas de datos para la etapa i.

- Dibujar el diagrama lógico del circuito de 4 bits utilizando sumadores completos y multiplexores.
 - Determinar la operación aritmética realizada por cada una de las 4 combinaciones de S y C_{in} : 00, 01, 10 y 11.
13. Diseñar la etapa i de un circuito digital que realiza las 4 operaciones lógicas: XOR, XNOR, NOR y NAND sobre los operandos almacenados en los registros A y B y cuyo resultado se almacena en el registro A. Utilizar 2 variables de selección para escoger entre las cuatro operaciones lógicas citadas.
- Utilizando diagramas de Karnaugh, diseñar con el mínimo número de puertas de un único tipo y mostrar el diagrama lógico resultante.
 - Repetir el apartado a) utilizando diferentes asignaciones para los códigos de selección de las cuatro operaciones para comprobar si el número de puertas se puede simplificar aún más.

14. Diseñar una ALU que realice las siguientes operaciones:

$A+B$	sl A
$A-B$	$A \vee B$ (OR)
\bar{A}	$A \oplus B$
$\bar{A} + 1$	$A \wedge B$ (AND)

El diseño debe constar de una única línea de acarreo de entrada. Dibujar el esquema del bit i de la ALU.

15. Indicar cuál sería la salida Y del desplazador circular de 4 bits de la figura 2.22 de la teoría para las siguientes combinaciones de entradas $S_1 S_0 D_3 D_2 D_1 D_0$.

- 110111
- 011000
- 001010
- 100011

16. Especificar cuál debe ser la palabra de control de 17 bits que debe ser aplicada a la ruta de datos de la teoría para implementar cada una de las microoperaciones:

- $R1 \leftarrow R2 + R6$
- $R3 \leftarrow \text{Data in}$
- $R4 \leftarrow \text{sl } R3$
- $R6 \leftarrow R4 - \text{Constant in}$
- $R3 \leftarrow R2 \oplus R7$
- $R7 \leftarrow 0$
- $R1 \leftarrow R0 + 1$
- $R2 \leftarrow \text{sr } R3$

17. Dadas las siguientes palabras de control de 17 bits para la ruta de datos de la teoría, determinar:

- La microoperación que se está ejecutando
- El cambio en el contenido de los registros para cada palabra de control suponiendo que se trata de registros de 8 bits y que previamente a la ejecución de la microoperación contienen el valor de su número (i.e. registro R5 contiene 05 en hexadecimal). Suponer que Constant vale 11 y que Data in vale 12 ambos en hexadecimal.

	DA	AA	BA	MB	FS	MD	RW
a)	101	100	101	0	01000	0	1
b)	110	010	100	0	00101	0	1
c)	101	000	110	0	10100	0	1
d)	101	000	000	0	00000	1	1
e)	100	100	000	1	01100	0	1
f)	011	000	000	0	01100	0	1

18. Dada la siguiente secuencia de palabras de control de 17 bits para la ruta de datos de la teoría y suponiendo que inicialmente en los registros R0.....R7 se guarda el correspondiente código ASCII de 8 bits que se indica a continuación, ejecutar las microoperaciones que aparecen en secuencia. El resultado es una palabra cifrada que se guarda en los registros R0-R7 (una letra en cada uno de ellos) ¿De qué palabra se trata?

Palabra de Control	Registro	Cont. Ini. Reg.
011 011 001 0 00010 0 1	R0	00000000
100 100 001 0 01010 0 1	R1	00100000
101 101 001 0 01100 0 1	R2	01000100
001 001 000 0 01110 0 1	R3	01000111
001 001 000 0 00001 0 1	R4	01010100
110 110 001 0 00101 0 1	R5	01001100
111 111 001 0 00101 0 1	R6	01000001
001 111 000 0 00000 0 1	R7	01001001

BITS				CONTROL				NUMBERS & SYMBOLS				UPPERCASE				LOWERCASE			
B7	B6	B5	B4	B3	B2	B1		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1				
0	0	0	0	0	0	0	NUL	0	16	32	48	64	80	96	112				
0	0	0	1	0	0	1	SOH	17	33	49	65	81	97	113					
0	0	1	0	0	1	0	STX	18	34	50	66	82	98	114					
0	0	1	1	0	1	1	ETX	19	35	51	67	83	99	115					
0	1	0	0	0	1	0	EOT	20	36	52	68	84	100	116					
0	1	0	1	0	1	1	ENQ	21	37	53	69	85	101	117					
0	1	1	0	0	1	0	ACK	22	38	54	70	86	102	118					
0	1	1	1	0	1	1	BEL	23	39	55	71	87	103	119					
1	0	0	0	0	1	0	BS	24	40	56	72	88	104	120					
1	0	0	1	0	1	0	HT	25	41	57	73	89	105	121					
1	0	1	0	0	1	0	LF	26	42	58	74	90	106	122					
1	0	1	1	0	1	1	VT	27	43	59	75	91	107	123					
1	1	0	0	0	1	0	FF	28	44	60	76	92	108	124					
1	1	0	1	0	1	0	CR	29	45	61	77	93	109	125					
1	1	1	0	0	1	0	SO	30	46	62	78	94	110	126					
1	1	1	1	0	1	1	SI	31	47	63	79	95	111	127					

19. Una ruta de datos tiene 5 componentes principales: A, B, C, D y E en un ciclo similar al de la ruta de datos sin canalización que hemos estudiado. El retardo máximo de cada uno de ellos es: A-2ns, B-1ns, C-3ns, D-4ns y E-4ns.

- ¿Cuál es la frecuencia de reloj máxima a la que puede funcionar?
- Si la convertimos en una ruta canalizada de 3 etapas. ¿Cómo se deben combinar las componentes y cuál es la máxima frecuencia de reloj que se podría alcanzar?
- Repetir b) para el caso de 4 etapas.

20. Realizar una simulación temporal similar a la estudiada en la teoría para la ruta de datos sin canalización, fig. 2.21. (transp 2-28) con la secuencia de microoperaciones de la Tabla 2.13. (trans 2-27) sobre la ruta de datos canalizada vista en teoría (fig. 2.24 (transp 2-30)). Suponer que los registros plataforma de la canalización se llaman OF/EX:A, OF/EX:B, EX/WB:F y EX/WB:DI. Suponer además que inicialmente los registros del archivo de registros contienen un valor igual a su número, que el valor en Constant In es 4F y que el valor en Data In es 0E

