

Ejercicio 1

1. Componentes de 3 estados

a) Completar la siguiente tabla:

U: undefined, indefinido. E: error.

A	A _{en}	B	B _{en}	C	C _{en}	Estimado	Obtenido
0	0	0	0	0	0	U	U
0	1	1	1	0	0	E	E
1	0	1	0	1	0	U	U
1	1	0	0	0	1	E	E
0	1	0	1	0	1	E	0
0	1	1	1	1	1	E	E
1	0	1	1	1	0	1	1

b) Completar la siguiente tabla:

Color	Interpretación
Gris	Sin señal. El cable no está conectado a una fuente
Verde claro	1 lógico
Verde oscuro	0 lógico
Azul	Indefinido
Rojo	Error

c) Enunciar la regla:

Para que el cable de salida nunca tome el color rojo las entradas de control deben ser todas 0, pero puede haber una única que sea 1.

d) Explicar cuáles son y por qué:

Cuando hay dos entradas de control prendidas, aunque pase el mismo valor, consideramos que sería error. En cambio, en Logisim es válido, permite pasar el valor repetido.

2. Transferencia entre registros

a) Detallar entradas y salidas:

- clk: es el clock, es una entrada de control.
- Force_Input: es una entrada que nos permite introducir un valor arbitrario en el circuito.
- en_Force_Input: es un enable que deja o no pasar el Force_Input, es una entrada de control.
- w0, w1, w2: tres writes que permiten cargar el valor de la entrada al registro, o ignorarlo si son 0. Son entradas de control.
- en_out0, en_out1, en_out2: tres enable out que permiten leer el valor cargado en los registros. Son entradas de control.
- R0, R1, R2: tres salidas que son de debug y permiten leer siempre el valor almacenado independientemente del enable out.
- E_S: entrada salida que se conecta a los 3 registros.

b) Secuencia de señales:

Para que el registro R1 pase a tener el valor 1	
clk	0
en_out0	0
en_out1	0
en_out2	0
en_Force_Input	0

Para que el registro R1 pase a tener el valor 1	
Force_Input	1
en_Force_Input	1
w1	1
clk	1

c) Secuencia de señales:

Para poner un valor arbitrario (VA) en R0	
w0	0
w1	0
w2	0
clk	0
en_out0	0
en_out1	0
en_out2	0
en_Force_Input	0
Force_Input	VA
en_Force_Input	1
w0	1
clk	1
w0	0

Transferirlo a R1	
en_Force_Input	0
en_out0	1
w1	1

Transferirlo a R1	
clk	0
clk	1
w1	0
en_out0	0

Pasar el valor de R2 a R0	
en_out2	1
w0	1
clk	0
clk	1
w0	0
en_out2	0

Pasar el valor de R1 a R2	
en_out1	1
w2	1
clk	0
clk	1
w2	0
en_out1	0

3. Máquina de 4 registros con suma y resta.

a) Detallar entradas y salidas:

- clk: reloj, es una entrada de control que discretiza el tiempo.
- ALU_A_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en la ALU.

- ALU_B_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en la ALU.
- OP: es una entrada de control de dos bits que determina que operación se hace entre dos tiras de 4 bits. (ADD: 00, SUB: 01, AND: 10, OR: 11)
- ALU_enableOut: es una entrada de control que determina si se puede leer el resultado de la ALU.
- Reg0_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en el registro 0.
- Reg0_enableOut: es una entrada de control que determina si se elige el valor almacenado en el registro 0 para la ALU.
- Reg1_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en el registro 1.
- Reg1_enableOut: es una entrada de control que determina si se elige el valor almacenado en el registro 1 para la ALU.
- Reg2_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en el registro 2.
- Reg2_enableOut: es una entrada de control que determina si se elige el valor almacenado en el registro 2 para la ALU.
- Reg3_Write: es una entrada de control que determina si se almacena internamente el valor de la entrada en el registro 3.
- Reg3_enableOut: es una entrada de control que determina si se elige el valor almacenado en el registro 3 para la ALU.
- Reg4_Debug: es una salida que aparece en todos los registros y expresa el valor almacenado.
- Force_Input: es una entrada de 4 bits que se podría almacenar en algunos de los registros, se conecta con la salida Reg4_output.
- en_Force_Input: es una entrada de control que determina si se utiliza el valor de Force_Input.
- A_Debug: es una salida que expresa el valor almacenado de A.
- B_Debug: es una salida que expresa el valor almacenado de B.
- S_Debug: es una salida que expresa el valor almacenado del resultado de la operación.
- Reg4_output: es una salida que expresa el valor del output si el enable out está activado. Cuatro pertenecen a cada registro, y una es salida de la ALU, que se conecta con la entrada Force_Input.
- N: es una salida que se enciende si el resultado en complemento a 2 es negativo.
- Z: es una salida que se enciende si el resultado es 0.

- V: es una salida que se enciende si la operación dio overflow.
- C: es una salida que se enciende si la suma binaria produjo acarreo o la resta borrow.
- A y B: es el valor seleccionado de uno de los registros para operar en la ALU.

b) Detallar el contenido de cada display:

Los 4 displays de los registros muestran el valor almacenado de cada uno (Reg4_Debug). A_Debug, B_Debug y S_Debug expresan el valor almacenado de A, B y S respectivamente, de la ALU. El display conectado al valor de salida-entrada Reg4_output muestra el que se obtiene de la ALU. Los ocho displays son expresados en base hexagesimal.

c) Secuencia de señales:

Cargar el valor 4 en el registro R2	
clk	0
en_Force_Input	0
ALU_enableOut	0
Force_Input	0100
Reg2_Write	1
en_Force_Input	1
clk	1
Reg2_Write	0

Cargar el valor -3 en el registro R3	
en_Force_Input	0
Force_Input	1101
Reg3_Write	1

Cargar el valor -3 en el registro R3	
en_Force_Input	1
clk	0
clk	1

d) Completar la siguiente tabla:

Valor inicial	Resultado operación 1	Interpretación sin signo	Interpretación complemento a 2	Flags
(4, 0)	OR 0100	4	4	0000
(7, -1)	SUB 1000	8	-8	1011
(-8, -2)	ADD 0110	6	6	0011
(8, -9)	El -9 y el 8 no se pueden representar			

Valor inicial	Resultado operación 2	Interpretación sin signo	Interpretación complemento a 2	Flags
(4, 0)	SUB 0100	4	4	0000
(7, -1)	AND 0111	7	7	0000
(-8, -2)	SUB 1010	10	no se puede representar	1001
(8, -9)	El -9 y el 8 no se pueden representar			

Para el primer caso, escribir la secuencia completa de activación de señales:

Cargar el valor 4 en el registro R0	
clk	0
Reg0_enableOut	0
Reg1_enableOut	0
Reg2_enableOut	0

Cargar el valor 4 en el registro R0	
Reg3_enableOut	0
en_Force_Input	0
ALU_enableOut	0
Force_Input	0100
Reg0_Write	1
en_Force_Input	1
Reg0_enableOut	1
clk	1
ALU_A_Write	1
clk	0
clk	1
Reg0_Write	0
ALU_A_Write	0
Reg0_enableOut	0
clk	0

Cargar el valor 0 en el registro R1	
en_Force_Input	0
Force_Input	0000
Reg1_Write	1
en_Force_Input	1
Reg1_enableOut	1
clk	1
ALU_B_Write	1
clk	0
clk	1
Reg1_Write	0

Cargar el valor 0 en el registro R1	
ALU_B_Write	0
Reg1_enableOut	0

Cargar resultado de OR en R2	
en_Force_Input	0
OP	11
clk	0
clk	1
ALU_enableOut	1
Reg2_Write	1
clk	0
clk	1
Reg2_Write	0
ALU_enableOut	0

Cargar resultado de SUB en R3	
OP	01
clk	0
clk	1
ALU_enableOut	1
Reg3_Write	1
clk	0
clk	1

e) Explicar:

Se niega para representar que la ALU tarda 1 periodo en calcular el resultado.