

Taller de Lógica Digital - Parte 2

Organización del Computador 1

Grupo 24: NullPointerException

Primer cuatrimestre 2025

1 Componentes de 3 estados

1.1 Completar la siguiente tabla

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

1.2 Completar la siguiente tabla

Color	Interpretación
Gris	Res no tiene ningún cable conectado.
Verde claro	A Res le llegarán cables verdes claros o "encendidos" solo con que una de las compuertas esté "abierta" y reciba un valor 1 de entrada.
Verde oscuro	Para que lleguen cables de color verde oscuro o apagado, es necesario que todas las entradas cuyas compuertas estén abiertas reciban de valor de entrada un 0.
Azul	Cuando las entradas A_{en} , B_{en} y C_{en} están en 0, Res recibe cables de color azul, puesto que al estar las "compuertas" cerradas, no hay ningún resultado que llegue a Res, por lo que sería lo mismo a que estén desconectadas.
Rojo	En caso de que más de una compuerta estuviera "abierta" y los valores de entrada fueran diferentes, se produciría un error, por lo que a Res le llegarían cables de color rojo.

1.3 Enunciar la regla

Solo se permitirá que haya más de una compuerta abierta si los valores de entrada de estas son iguales entre sí

1.4 Explicar cuáles son y por qué

Se interpreta como basura las combinaciones donde distintos valores de entrada, o sea 1 y 0, pasan por compuertas abiertas diferentes, causando un error en el circuito

2 Transferencia entre registros

2.1 Detallar entradas y salidas

- clk
 - Señal de reloj: controla el momento en que se escriben los valores en los registros (flanco de subida).
 - Entrada de control.
- Force_Input
 - Se usa para cargar valores manualmente a los registros.
- en_Force_Input
 - Controla un buffer: si es 1, se selecciona el valor de Force_Input; si es 0, se selecciona la salida de uno de los registros activados.
 - Entrada de control.
- w
 - Señal de escritura: si es 1 al momento del flanco de subida del clk, el valor presente en la entrada se guarda en el registro.
 - Entrada de control.
- en_out
 - Habilita la salida del valor del registro al bus compartido.
 - Entrada de control.

2.2 Secuencia de señales

Paso	Señal	Valor	Acción
1	Force_Input	1	Se coloca el valor
2	en_Force_Input	1	Se selecciona Force_Input como fuente
3	w (R1)	1	Se habilita la escritura en R1
4	clk	↑ (ó 1)	Se carga el valor 1 en el registro R1
5	w (R1)	0	Finalizo la escritura

2.3 Secuencia de señales

Paso	Señal	Valor	Acción
1	Force_Input	1	Se coloca el valor
2	en_Force_Input	1	Se selecciona Force_Input como fuente
3	w (R0)	1	Se habilita la escritura en R0
4	clk	↑ (ó 1)	Se carga el valor 1 en el registro R0
5	w (R0)	0	Finalizo la escritura
6	en_Force_Input	0	Desahibilita la entrada Force_Input
7	en_out (R0)	1	Habilito la salida de R0
8	w (R1)	1	Habilito la escritura en R1
9	clk	↑ (ó 1)	Transfiero el valor de R0 a R1
10	en_out (R0)	0	Desactivo la salida de R0
11	w (R1)	0	Finalizo la escritura en R1
12	en_out (R2)	1	Habilito la salida de R2
13	w (R0)	1	Habilito la escritura en R0
14	clk	↑ (ó 1)	Transfiero el valor de R2 a R0
15	en_out (R2)	0	Desactivo la salida de R2
16	w(R0)	0	Finalizo la escritura en R0
17	en_out (R1)	1	Habilito la salida de R1
18	w (R2)	1	Habilito la escritura en R2
19	clk	↑ (ó 1)	Transfiero el valor de R1 a R2
20	en_out (R1)	0	Desactivo la salida de R1
21	w (R2)	0	Finalizo la escritura en R2

3 Máquina de 4 registros con suma y resta

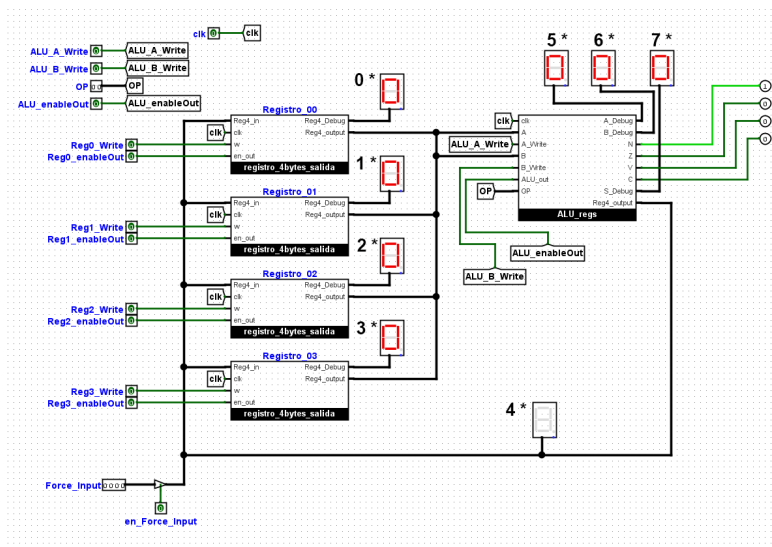
3.1 Detallar entradas y salidas

- Entradas de datos
 - Force.Input: valor forzado manualmente (de forma externa)
 - Reg0.output \wedge Reg1.output: salidas de los registros conectadas al bus de datos
- Entradas de control
 - RegX.write: habilita la escritura en el registro X
 - RegX.enableOut: habilita que el contenido del registro X sea puesto en el bus
 - en.Force.Input: habilita que Force.Input envíe su valor al bus
 - ALU_A.Write \wedge ALU_B.Write: seleccionan los operandos A, B para la ALU
 - OP: código de operación de la ALU
 - ALU.enableOut: activa la salida de la ALU al bus
 - clk: "pulso" de reloj para registrar el valor
- Salidas
 - ALU_regs: módulo que devuelve el resultado de la ALU
 - Flags N, Z, V, C
 - Salidas individuales de cada registro

3.2 Detallar el contenido de cada display

Cada display conectado a la salida de un registro o componente devuelve el contenido del bus de datos:

Display	Contenido	Descripción
0	R0	Valor del registro en hexadecimal
1	R1	Valor del registro en hexadecimal
2	R2	Valor del registro en hexadecimal
3	R3	Valor del registro en hexadecimal
4	Force_Input	Valor del Force.Input en hexadecimal
5	Entrada A de la ALU	Operando A de la ALU
6	Entrada B de la ALU	Operando B de la ALU
7	Salida de la ALU	Resultado actual de la operación en hexadecimal



Referencia utilizada en la tabla de displays

3.3 Secuencia de señales

Paso	Señal	Valor	Acción
1	Force_Input	0100	Se coloca el valor 4
2	en_Force_Input	1	Se selecciona Force_Input como fuente
3	Reg2_Write	1	Se habilita la escritura en R2
4	clk	↑ (ó 1)	Se carga el valor 4 en el registro R2
5	en_Force_Input	0	Deshabilito la salida de Force.Input
6	Reg2_Write	0	Deshabilito la escritura en R2
7	clk	0	Se reinicia el reloj

Carga de 4 en el registro R2

Paso	Señal	Valor	Acción
1	Force_Input	1101	Se coloca el valor -3
2	en_Force_Input	1	Se selecciona Force_Input como fuente
3	Reg3_Write	1	Se habilita la escritura en R3
4	clk	\uparrow (ó 1)	Se carga el valor -3 en el registro R3
5	en_Force_Input	0	Deshabilito la salida de Force_Input
6	Reg3_Write	0	Deshabilito la escritura en R3
7	clk	0	Se reinicia el reloj

Carga de -3 en el registro R3

3.4 Completar la siguiente tabla

Aclaración: Res OP_n = sin signo | complemento a 2 (tipo de operación)

Valor inicial	Res OP_1	Flags	Res OP_2	Flags
(4, 0)	4 4 (\vee)	N=0, Z=0, V=0, C=0	4 4 (-)	N=0, Z=0, V=0, C=0
(7, -1)	8 -8 (-)	N=0, Z=1, V=1, C=1	7 7 (\wedge)	N=0, Z=0, V=0, C=0
(-8, -2)	6 6 (+)	N=0, Z=0, V=0, C=1	10 -6 (-)	N=1, Z=0, V=0, C=1
(8, -9)	-	-	-	-

El caso (8, -9) se excede el rango representable en 4 bits con complemento a 2, el cual es de -8 a +7, por lo tanto no puede ser representado correctamente.

3.5 Explicar (\neg clk)

El **clk negado** se utiliza como entrada "de reloj" para el registro que guarda el resultado de la ALU (registro_4bytes_salida). El resto de los componentes (registros A y B e incluso la ALU) utilizan el clk estandar, que no está negado. Esto esta implementado de forma que haya una separación temporal entre 2 circunstancias:

- El momento en que los registros A y B cargan los operandos; en el flanco de subida de clk (\uparrow clk) ó el flanco de bajada del \neg clk ($\downarrow \neg$ clk)
- El momento en que el registro de salida (registro_4bytes_salida) captura el resultado; en flanco de subida del \neg clk ($\uparrow \neg$ clk) ó el flanco de bajada del clk original (\downarrow clk)

Para uso del docente:

1	2	3

Alumnos:

- Nombre y Apellido:
- Nombre y Apellido:
- Nombre y Apellido: