

Taller de Lógica Digital - Parte 2

Organización del Computador 1

Primer Cuatrimestre 2023

Ejercicios

1. Componentes de 3 estados

a) Completar la siguiente tabla:

A	A _{en}	B	B _{en}	C	C _{en}	Estimado	Obtenido
0	0	0	0	0	0	No Conectado	No Conectado
0	1	1	1	0	0	Error	Error
1	0	1	0	1	0	No conectado	No conectado
1	1	0	0	0	1	Error	Error
0	1	0	1	0	1	0	0
0	1	1	1	1	1	Error	Error
1	0	1	1	1	0	1	1

b) Completar la siguiente tabla:

Color	Interpretación
Gris	Cables sin ningún conexión.
Verde claro	La señal es 1.
Verde oscuro	La señal es 0.
Azul	La señal es indefinida o desconocida.
Rojo	Están pasando dos valores distintos por el mismo cable.

c) Enunciar la regla:

La condición o regla sería la siguiente:

Si dos o más componentes de 3 estados están dejando pasar la señal entonces esas señales deben tener el mismo valor.

d) Explicar cuáles son y por qué:

Las combinaciones basura son aquellas que tienen más de un componente de 3 estados activo al mismo tiempo. Esta se debe a que son lógicamente inútiles (y la misma lógica podría ser representado con menos compuertas de forma más simple).

2. Transferencia entre registros

a) Detallar entradas y salidas:

Entradas:
ckl: control que determina cuando se guarda el dato de entrada Force_input
Force_input: data a almacenar en los registros
en_Force_input: control para almacenar un valor arbitrario en el circuito
Salidas:
R0, R1, R2: de monitoreo de dato almacenado en los registros

b) Secuencia de señales:

- I. Habilitar en_Force_input en 1,
- II. Habilitar la escritura(W) de R1 en 1,
- III. Habilitar Force_input en 1,
- IV. Un clk (flanco ascendente).

c) Secuencia de señales:

- I. en_out R0 en 1,
- II. activar w de R1 en 1,
- III. clk(flanco ascendente),
- IV. desactivar w de R1
- V. desactivar en_out R0,
- VI. ---- en_out de R2 en 1,
- VII. Activar w de R0 en 1,
- VIII. clk(flanco ascendente),
- IX. desactivar w de R0,
- X. desactivar en_out de R2
- XI. --- activar w de R2,
- XII. activar en_out de R1 ,
- XIII. clk(flanco ascendente).

3. Máquina de 4 registros con suma y resta.

a) Detallar entradas y salidas:

Entradas Control:	
OP: señal de control de operaciones de la ALU	
clk : señal de control para guardar en los registros	
De Reg0_Write ...Reg3_Write : señal de control que habilita la escritura	
De Reg0_EnableOut...Reg3_EnableOut: señal de control que habilita la salida de los registros 00 a 03	
ALU_A_Write y ALU_B_Write : señal de control que habilita a la escritura de los registros de la ALU A y B	
ALU_EnableOut: señal de control que habilita al cálculo almacenado en la ALU_regs	
en_Force_input: señal de control que habilita la entrada de un valor arbitrario a los registros de 00 a 03	
Entrada:	
Force_input: ingresa un valor arbitrario en los registros 00 al 03	
Salidas:	
FLAGS de la ALU:	
N : Indica si el valor almacenado es negativo	
Z : Indica si el valor almacenado es 0	
V: Indicar si hubo overflow	
C: en el caso de la suma indica si hubo carry -out y en el caso de la resta indica si hubo borrow.	

b) Detallar el contenido de cada display:

Displays	Funcionalidad
Displays de registros 00 hasta 03	Todos representan el valor almacenado en su registro de 4 bits correspondiente.
Displays conectados A y B debug	representan el valor almacenado en los registros A y B de la ALU_regs.
Display S-debug	representa el valor almacenado resultante de la operación realizada por la ALU_regs.

Como observación final todos los displays interpretan la información almacenada como sin signo y en base hexadecimal.

c) Secuencia de señales:

- I. Activar Reg2_Write de R2,
- II. Activar en_Force_input,
- III. En el Force_input ingreso 0100 (4),
- IV. Realizar un clk(flanco ascendente),
- V. Volver a clk (0),
- VI. ---- Desactivar Reg2_Write de R2,
- VII. Activar Reg3_Write R3,
- VIII. Ingresar en el Force_input 1101 (-3),
- IX. clk (flanco ascendente).

Los números están interpretados en complemento a 2.

d) Completar la siguiente tabla:

Los resultados interpretados en sin signo y en complemento a 2.

Valor inicial	Resultado operación 1	Flags	Resultado operación 2	Flags
(4,0)	C2: 4 Sin-signo: 4	N:0,Z:0,V:0,C:0	C2: 4 Sin-signo: 4	N:0,Z:0,V:0,C:0
(7, -1)	C2: -8 Sin-signo: 8	N:1,Z:0,V:1,C:1	C2: 7 Sin-signo: 7	N:0,Z:0,V:0,C:0
(-8, -2)	C2: 6 Sin-signo: 6	N:0,Z:0,V:1,C:1	C2: -6 Sin-signo: 10	N:1,Z:0,V:1,C:1
(8, -9)	C2: Sin-signo:	No representables de C2 y en Sin-Signo	C2: Sin-signo:	No representables de C2 y en Sin-Signo

Secuencia de señales de primer caso:

Almacenamos los datos en el R1 y R0:

- I. Activar en_Force_input,
- II. Cargar en Force_input el numero 0100(4),
- III. Habilitar Reg0_Write de R0 ,
- IV. clk(flanco ascendente),
- V. Desactivar Reg0_Write de R0,en_Force_input ,
- VI. Activar Reg1_Write de R1,
- VII. Cargar en Force_input el numero 1101(-3)
- VIII. Desactivar en_Force_input,
- IX. clk(flanco ascendente).
- X. -- Habilitar la salida de R0,
- XI. Activar w de Alu_A_WRITE,
- XII. clk(flanco ascendente),
- XIII. Desactivar w de Alu_A_Write,
- XIV. Desactivar la salida de R0,
- XV. Activar la salida de R1,
- XVI. Activar W ALU_B_WRITE,
- XVII. clk(flanco ascendente).
- XVIII. Desactivar la salida de R1
- XIX. Desactivar el W de ALU_B_WRITE

--- Ahora realizar las operaciones:

- XX. Operación: "OR" en op(11),
- XXI. clk(flanco ascendente),
- XXII. Activar la salida de alu_regs, activar Reg2_Write de R2,
- XXIII. clk(flanco ascendente),
- XXIV. desactivar la Reg2_Write R2,
- XXV. Operación: "sub" en op(01),
- XXVI. clk(flanco ascendente),
- XXVII. Habilitar Reg3_Write de R3,
- XXVIII. clk(flanco ascendente).

e) Explicar

En la ALU se niega el clk para sincronizar el almacenamiento del dato en el flanco opuesto del reloj, como la ALU procesa datos en el flanco ascendente de clk, el registro de salida almacena cuando el flanco es descendente para estabilizar la información.

Corrección

Integrantes:

Nombre y Apellido: Martin Nievas Wilberger

LU: 453/24

Nombre y Apellido: Felipe Durini

LU: 469/24

Nombre y Apellido: Ludmila Wang

LU: 589/24

Para uso de los docentes:

1	2	3