

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Facultad de Ingeniería – Ingeniería Electrónica

Análisis y diseño de microprocesadores



UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

### **Integrantes:**

- Leidy Katherine Castelblanco Romero 20151005571
- Nicolás David Pastran Zamora 20151005087

### **Planteamiento del problema**

Como ya se sabe la tarjeta FPGA nos permite por medio de la descripción de hardware implementar variedad de circuitos lógicos ya sean combinacionales o secuenciales; nuestra tarea consiste en implementar un flip flop tipo D con flanco de bajada a partir de dos latch tipo D y con este flip flop implementar un registro universal en alto y bajo nivel de 8 bits. Que haga rotación, carga paralela, desplazamiento de bits de izquierda a derecha y desplazamiento de derecha a izquierda.

### **Solución**

El registro lo implementaremos como es de 8 bit tenemos que definir el número de Flip Flops el cual será de 8 dependiendo de las salidas ya que la salida de los Flips serán las salidas del registro universal. Para bajo nivel con el flip flop tipo D con flanco de bajada se diseñara a base de dos latch tipo D donde la salida del primero será la entrada del segundo, haciéndolo un flip flop maestro esclavo, donde un latch depende del otro. Para el diseño de alto nivel, usaremos un process que dependa de un ciclo de reloj, y de las entradas usadas para seleccionar operación. En cada uno de los casos la selección de operación la realizaremos con un multiplexor a la entrada de cada uno de los Flip Flop, y los selectores de todos los multiplexores se unen de tal manera que serán los seleccionadores de operación del circuito global.

Diagrama de Bloques

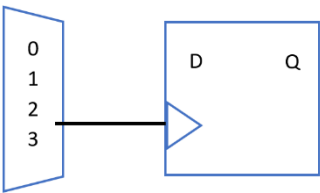


Figura 1 – Diagrama de Bloques Preliminar

RTL (VHDL):

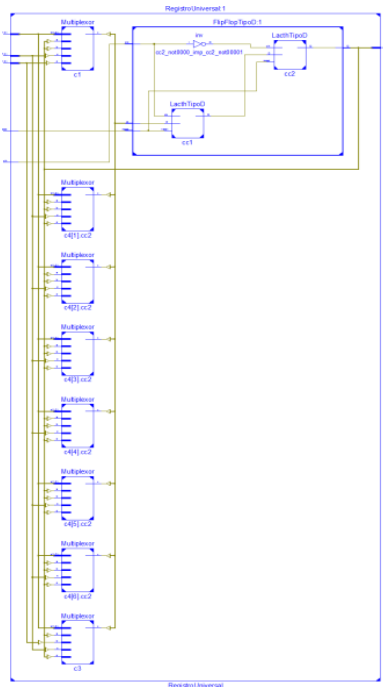


Figura 2 – RTL registró bajo nivel

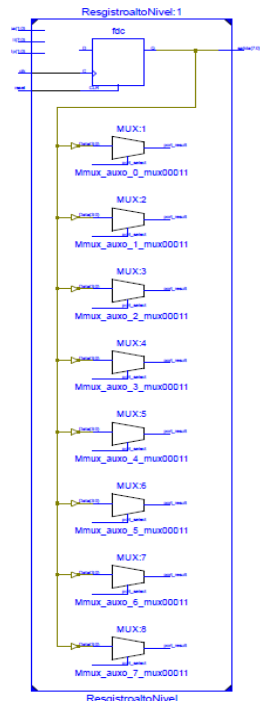


Figura 2 – RTL registró alto nivel

Diseño

Para comenzar el diseño del circuito definiremos las funciones que debe realizar el registro universal y el orden, tales funciones las podemos ver en la tabla 1.

C <sub>1</sub>	C <sub>0</sub>	FUNCION
0	0	Rotación
0	1	Desplazamiento hacia la izquierda
1	0	Carga Paralela
1	1	Desplazamiento hacia la derecha

Tabla 1 –Modo de funcionamiento del registro

Definimos las entradas y las salidas del circuito, De modo que:

I (0-7): Son las entradas

CLK: la señal de reloj del circuito

Reset: Pone 0 a las salidas de los FF

C(0-1): Carga y descarga (en este caso los seleccionadores de operación )

IS(0-1): Los valores para que remplazaran cada bit cuando hay desplazamiento.

S(0-7) : son las salidas

Las salidas de los multiplexores deben de ir conectadas a las entradas de los Flip Flops de modo que pase una de las 4 entradas del multiplexor a la salida. Los seleccionadores deben ser c1 y c2. De modo que para que el primer multiplexor dependiendo de la operación, a las posiciones del selector será:

$X_0$	$X_1$	$X_2$	$X_3$
S(7)	S(7)	I(0)	IS(1)

Desde el segundo multiplexor al séptimo será

$X_0$	$X_1$	$X_2$	$X_3$
S(n-1)	S(n-1)	I(n)	S(n-1)

Y para el último multiplexor:

$X_0$	$X_1$	$X_2$	$X_3$
S(6)	IS(0)	I(7)	S(6)

## Resultados y Análisis

### Simulación del circuito

- Carga Paralela

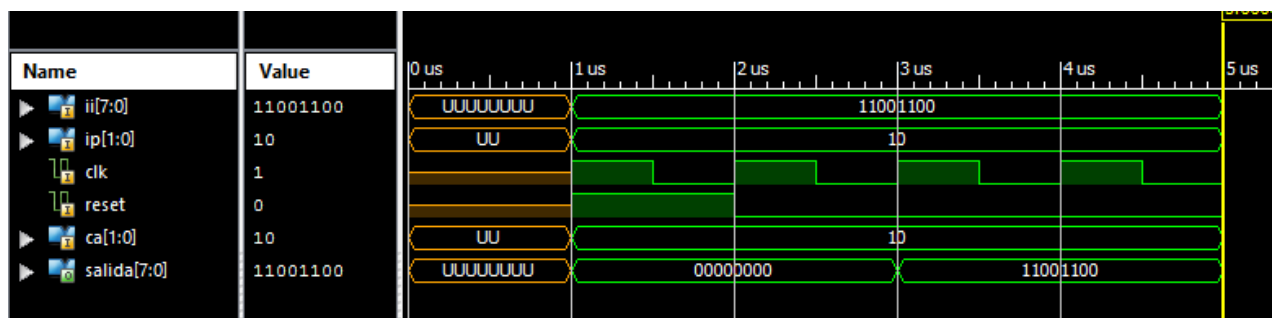


Figura 5 – Simulación Carga Paralela

Como ya se dijo el registro se implementó con flips flops tipo D y además tiene reset asíncrono, así que el reset inicializa la salida en cero, y en el siguiente flanco la salida se carga con la entrada (ii), ya que nuestro control ca esta en “10” y este determina que la función a realizar es carga paralela, y esto se cumple como se visualiza en figura 5.

- Rotacion

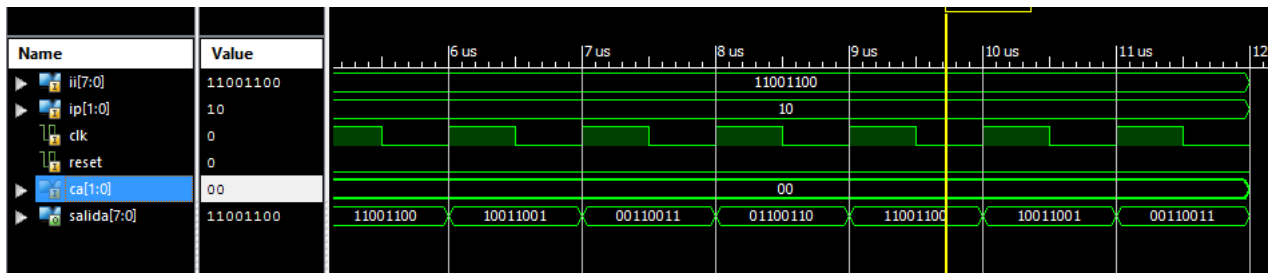


Figura 6 – Simulación Rotación

Ya que nuestras salidas estan inicializadas, el reset se mantiene en cero y el que varia como siempre es el CLK. Como se puede ver en la figura 6 nuestro control ca esta en “00” lo cual nos determina que la funcion a realizar es la rotacion, y vemos como en cada flanco nuestra salida va a cambiando, cargando el bit 0 con el bit 7, el bit 7 con el bit 6, el bit 6 con el bit 5 y asi sucesivamente, cargando a un bit x (x variando de 1 a n-1) de la salida lo que tiene el bit x-1, y para el bit 0 se cargara el bit n-1.

- Desplazamiento a la derecha

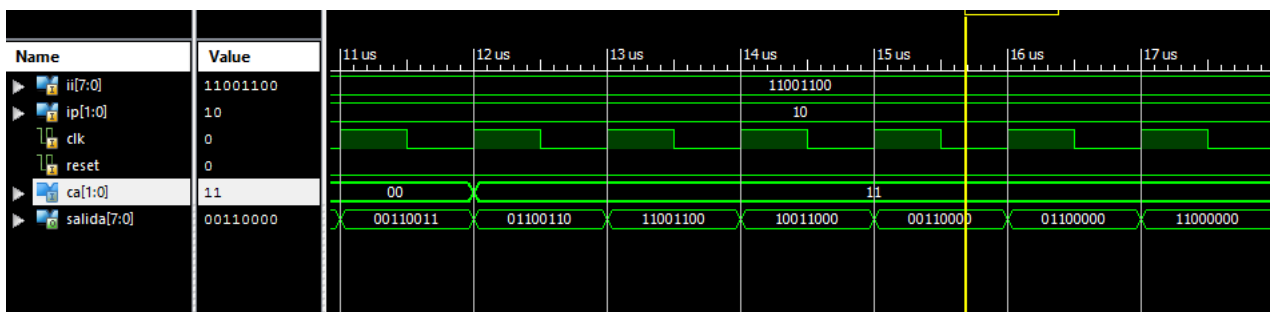


Figura 7 – Simulación Desplazamiento a la Derecha

Ya que nuestras salidas estan inicializadas, el reset se mantiene en cero y el que varia como siempre es el CLK. Como se puede ver en la figura 7 nuestro control ca esta en “11”

lo cual nos determina que la funcion a realizar es el desplazamiento a la derecha, y vemos como en cada flanco nuestra salida va a cambiando, cargando con ceros del bit menos significativo al mas significativo en cada flanco del CLK, se carga con ceros ya que la entrada lp(0) esta en cero, si lp(0) fuera uno, sucederia lo mismo pero con unos, del bit menos significativo al mas significativo uno a uno se volverian unos.

- Desplazamiento a la izquierda

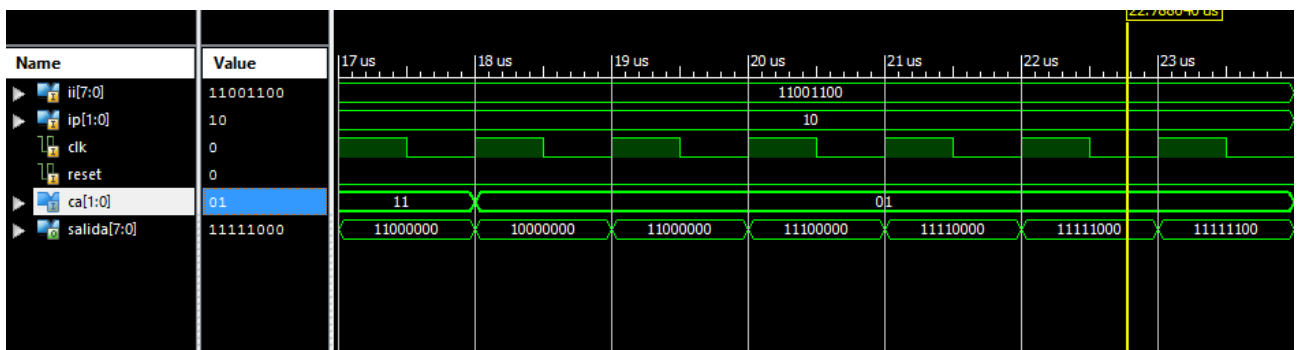


Figura 8 – Simulación Desplazamiento a la Izquierda

Ya que nuestras salidas estan inicializadas, el reset se mantiene en cero y el que varia como siempre es el CLK. Como se puede ver en la figura 8 nuestro control ca esta en “01” lo cual nos determina que la funcion a realizar es el desplazamiento a la izquierda, y vemos como en cada flanco nuestra salida va a cambiando, cargando con unos del bit mas significativo al menos significativo en cada flanco del CLK. Se carga con unos ya que la entrada lp(1) esta en uno, si lp(1) fuera cero, sucederia lo mismo pero con ceros, del bit mas significativo al menos significativo uno a uno se cargaran con ceros.

## Paramentos de desempeño

### En alto nivel

- Numero de recursos:

#### Device utilization summary:

Selected Device : 3s200aft256-5

Number of Slices:	8	out of	1792	0%
Number of Slice Flip Flops:	8	out of	3584	0%
Number of 4 input LUTs:	16	out of	3584	0%
Number of IOs:	22			
Number of bonded IOBs:	22	out of	195	11%
Number of GCLKs:	1	out of	24	4%

#### Advanced HDL Synthesis Report

##### Macro Statistics

# Registers	: 8
Flip-Flops	: 8
# Multiplexers	: 8
1-bit 4-to-1 multiplexer	: 8

- Tiempo de retardo:

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 2.089ns (frequency: 478.709MHz)

Total number of paths / destination ports: 21 / 8

Delay: 2.089ns (Levels of Logic = 2)

Source: auxo\_5 (FF)

Destination: auxo\_6 (FF)

Source Clock: clk rising

Destination Clock: clk rising

Data Path: auxo\_5 to auxo\_6

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDC:C->Q	4	0.495	0.607	auxo_5 (auxo_5)
LUT4:I0->O	1	0.561	0.000	Mmux_auxo_6_mux00013_G (N27)
MUXF5:I1->O	1	0.229	0.000	Mmux_auxo_6_mux00013 (auxo_6_mux0001)
FDC:D		0.197		auxo_6

Total 2.089ns (1.482ns logic, 0.607ns route)  
(70.9% logic, 29.1% route)

#### En bajo nivel

- Numero de recursos:

#### Device utilization summary:

Selected Device : 3s200aft256-5

Number of Slices:	13	out of	1792	0%
Number of Slice Flip Flops:	16	out of	3584	0%
Number of 4 input LUTs:	17	out of	3584	0%
Number of IOs:	22			
Number of bonded IOBs:	21	out of	195	10%
Number of GCLKs:	1	out of	24	4%

#### Advanced HDL Synthesis Report

##### Macro Statistics

# Latches	: 16
1-bit latch	: 16
# Multiplexers	: 8
1-bit 4-to-1 multiplexer	: 8

- **Tiempo de retardo:**

Data Path: c<0> to cc1/cc1/Q

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	16	0.824	0.987	c_0_IBUF (c_0_IBUF)
LUT3:I0->O	1	0.561	0.000	c1/Mmux_x_3 (c1/Mmux_x_3)
MUXF5:I1->O	1	0.229	0.000	c1/Mmux_x_2_f5 (xD<0>)
LD:D		0.197		cc1/cc1/Q
Total		2.798ns (1.811ns logic, 0.987ns route) (64.7% logic, 35.3% route)		

## Análisis de Resultados

En los dos casos los números de recursos son muy similares ya que en ambos caso usamos la misma cantidad de multiplexores 4-1 para seleccionar operaciones y la cantidad de Flips Flops tipo D, sin embargo como en el modelo a bajo nivel usamos FF's a base de latches y alguna compuerta eso representa una cantidad más grande de recursos la cual se ve reflejada en el tiempo de respuesta del en cada de los casos ya que para bajo nivel el tiempo de respuesta es 2.798ns y para alto nivel 2.089 ns.

## Conclusiones

- Al describir un circuito secuencial en vhdl, el circuito secuencial deberá ser a base de Flip Flops Tipo D, ya que por su versatilidad es muy fácil de implementar por la FPGA, sin embargo no en todos los casos es el Flip Flop más adecuado para implementar un circuito digital.
- Cuando hacemos una asignación dentro de un condicional que depende de la señal de reloj esta generara un Flip Flop para cada asignación, sin importar que no sea una salida directamente dependiente de la señal reloj.