Ritik Garg

1) 4 bit adder
2) 8 to 3 encoder
3) 8 bit universal shift register with serial as well as parallel input, parallel output with —

Functionality

Shl = 1	Left shift, LSB FF will take '0' as input
Sh = 1	Right shift, MSB FF take SI (Serial input)
L = 1	Parallel inputs
All zero	FF retain their values

Only one of the "L", "Sh", "Shl" could be '1' at a time

- Truth table and equations

1) Sum = A XOR B XOR C Cout = AB + BC + CA

A	B	C	Sum		Cout	
0	0	0	0	6	0	8
0	0	1	1		0	
0	1	0	1		0	
0	1	1	0		1	
1	0	0	1	9	0	E
1	0	1	0		1	
1	1	0	0		1	
1	1	1	1		1	

configuration for Sum – 96969696
for Cout – E8E8E8E8

2) Inputs: $I_0, I_1, I_2 \ldots\ldots I_7$
Outputs: Y_2, Y_1, Y_0

$Y_2 = I_4 + I_5 + I_6 + I_7$

$Y_1 = I2 + I3 + I6 + I7$

$Y_0 = I1 + I3 + I5 + I7$

Only one of the inputs can be high at a time. But these don't care conditions are ignored while constructing truth table for determining configuration (As it won't affect)

To find the configuration of logic tile with I_4, I_5, I_6 and I_7 as inputs and Y_2 as output :-

I_4	I_5	I_6	I_7	Y_2	
0	0	0	0	0	
0	0	0	1	1	E
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	1	
0	1	0	1	1	F
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	1	
1	0	0	1	1	F
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	1	F
1	1	1	0	1	
1	1	1	1	1	

Configuration for Y_2 — FFFEFFFE

Similarly for logic tile having I_2, I_3, I_6 and I_7 as inputs and Y_1 as output, configuration is FFFEFFFE.

Same for inputs I_1, I_3, I_5 and I_7, and output Y_0 configuration is FFFEFFFE.

3) General eq^n for flip flops:

$$X = shl * Q_{i-1} + sh * Q_{i+1} + L * D_i + shl'\, sh'\, L'\, Q_i$$

Put $shl = 0$ and $shl = 1$, we get:-

$$Z_1 = sh * Q_{i+1} + L * D_i + sh'\, L'\, Q_i$$ (five variable function)

$$Z_2 = Q_{i-1} + sh * Q_{i+1} + L * D_i$$ (five variable function)

Final eq^n —

$$X = shl' * Z_1 + shl * Z_2$$

However, the fact that only one of shl, sh and L will be one at a time, can be used to simplify the calculations and reduce the number of logic tiles.

Equivalently the expression can be written as:-

$$X = shl * Q_{i-1} + \underbrace{shl' * sh * Q_{i+1}}_{\text{since here } shl = 0 \Rightarrow shl' = 1} + shl' * L * D_i + shl'\, sh'\, L'\, Q_i$$

$= Sh_1 * Q_{i-1} + Sh_1' (Sh * Q_{i+1} + L * D_i + Sh' L' Q_i)$

Let $Z = Sh * Q_{i+1} + L * D_i + Sh' L' Q_i$

Hence the expression can be written as -

$Y = Sh_1 * Q_{i-1} + Sh_1' Z$

Sh	L	Q_{i+1}	D_i	Q_i	Z	
0	0	0	0	0	0	A
0	0	0	0	1	1	
0	0	0	1	0	0	
0	0	0	1	1	1	
0	0	1	0	0	0	A
0	0	1	0	1	1	
0	0	1	1	0	0	
0	0	1	1	1	1	
0	1	0	0	0	0	C
0	1	0	0	1	0	
0	1	0	1	0	1	
0	1	0	1	1	1	
0	1	1	0	0	0	C
0	1	1	0	1	0	
0	1	1	1	0	1	
0	1	1	1	1	1	
1	0	0	0	0	0	O
1	0	0	0	1	0	
1	0	0	1	0	0	
1	0	0	1	1	0	
1	0	1	0	0	1	F
1	0	1	0	1	1	
1	0	1	1	0	1	
1	0	1	1	1	1	
1	1	0	0	0	0	C
1	1	0	0	1	0	
1	1	0	1	0	1	
1	1	0	1	1	1	
1	1	1	0	0	1	F
1	1	1	0	1	1	
1	1	1	1	0	1	
1	1	1	1	1	1	

Shl	Q_{i-1}	Z	X	
0	0	0	0	A
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	C
1	0	1	0	
1	1	0	1	
1	1	1	1	

Configuration for Z FCF0CCAA

Configuration for X CACACACA

Deciding Input & Output:

The universal shift register has the maximum number of inputs and outputs amongst the three circuits. Hence, it decides the number of input and output ports in FPGA.

Number of input ports = 14

Input ports: in0, in1, in2, in3, in4, in5, in6, in7, in8, in9, in10, in11, clk, rst

Output ports: out0, out1, out2, out3, out4, out5, out6, out7

# of output ports = 8

Semantic meaning of the ports for the three circuits:

1) 4 bit adder

Number to be added – A ($a_3 a_2 a_1 a_0$) and B ($b_3 b_2 b_1 b_0$)

Input carry – cin

Output – Sum ($S_3 S_2 S_1 S_0$) and cout

Semantics

in0 – a_0
in1 – a_1
in2 – a_2
in3 – a_3
in4 – b_0
in5 – b_1
in6 – b_2
in7 – b_3
in8 – cin

out0 – S_0
out1 – S_1
out2 – S_2
out3 – S_3
out4 – cout

2) 8 to 3 encoder

Inputs: $I_0, I_1, I_2 \ldots I_7$

Outputs: Y_2, Y_1, Y_0

Semantics

in0 – I_0
in1 – I_1
in2 – I_2
in3 – I_3
in4 – I_4
in5 – I_5
in6 – I_6
in7 – I_7
out0 – Y_0
out1 – Y_1
out2 – Y_2

3) 8-bit universal shift register-

Inputs: Parallel inputs to all flip flops - $D_7, D_6, D_5 \ldots D_0$
Serial input to most significant flip flop - SI
Shl, Sh, L, reset, clock

Outputs: Parallel outputs of all flip flops
$Q_7, Q_6, Q_5 \ldots Q_0$

Semantics:

in0 - D_0 in4 - D_4 in8 - SI rst-reset
in1 - D_1 in5 - D_5 in9 - L clk - clock
in2 - D_2 in6 - D_6 in10 - Sh
in3 - D_3 in7 - D_7 in11 - Shl

out0 - Q_0 out4 - Q_4
out1 - Q_1 out5 - Q_5
out2 - Q_2 out6 - Q_6
out3 - Q_3 out7 - Q_7

Design:

We can use switch box as one-hot encoded MUX, by using only one of the output ports of the switch box. In general, at every input of a logic tile, there can maximally be three choices (since there are three circuits) to select from. Similarly for deciding the value passed to

an output port, one can maximally encounter three different choices. Let us have a uniform configuration of all switch boxes for a specific circuit, but different from the others.

Adder- 0004
Encoder- 0002
Register- 0001

} configuration of switch boxes.

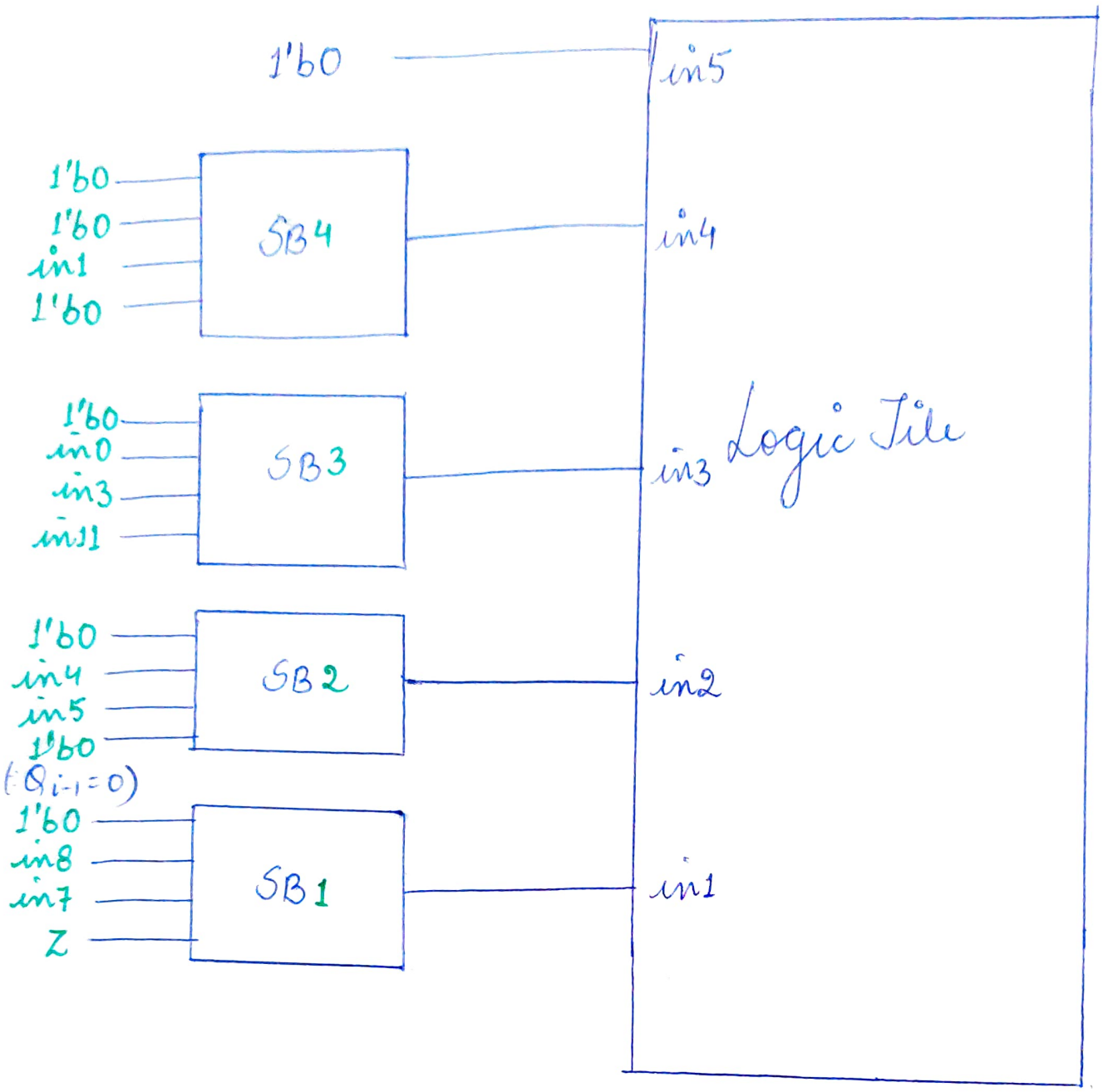
Consider the example of the first logic tile.

The inputs to this logic tile will be-

- for adder - $a_0 \equiv in0$
  $b_0 \equiv in4$
  $cin \equiv in8$
- for encoder- $I_1 \equiv in1$
  $I_3 \equiv in3$
  $I_5 \equiv in5$
  $I_7 \equiv in7$
- for register - $shl \equiv in11$
  Q_{i-1}
  Z

Choosing semantics wisely can help to reduce the number of switch boxes required.

Here, for first logic tile, we require four switch boxes.



The same idea is used for the complete design