LAB 6- Configurable fabric description

Rishabh Jain

2019CSB1286

AIM- To design an FPGA for the purpose of a 4 bit synchronous counter, 4 bit adder and an 8:3 encoder.

A. Description of basic elements

1. Logic tile description-

Declaration of logic tile

```
module logic_tile(out, clock, in1, in2, in3, in4, in5);
```

1. in1, in2, in3, in4, in5 are single bit inputs. They are labelled as A, B, C, D, E respectively for a convenient naming convention of switchboxes. in1 is the LSB and in5 is the MSB.
2. out is a single bit output
3. clock is single bit input clock for latched logic tile

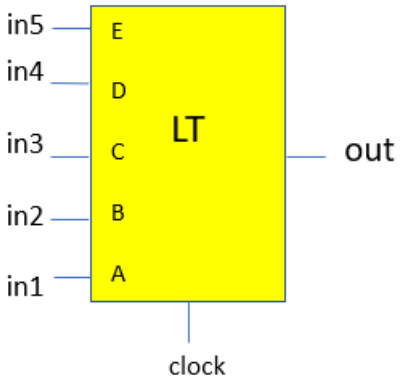

Figure 1- logic tile

4. Each logic tile contains an LUT (Look up table) with a 32 bit (2^5 bit) memory which can be configured according to use.
5. Each logic tile contains a D-flipflop for latched output.
6. Each logic tile can be configured to either give a synchronous or non-synchronous output.

2. Switch box description

Declaration of the 4x4 switch box -

```
module switch_box_4x4(out, in);
```

1. In[3:0] is a 4 bit input
2. Out[3:0] is a 4 bit output

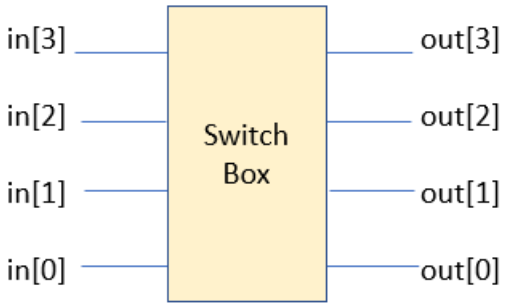

Figure 2- Switch box

Each output of the switch is one-hot encoded.

Each switch box contains a 16 bit memory for one-hot encoding of the outputs.

The following table describes the one-hot encoding of each of the 4 1-bit outputs of the switch box. 1st column contains the encoding of any one output, and the 2nd the output corresponding to it.

Configuration of out	Output
0001	in[0]
0010	in[1]
0100	in[2]
1000	in[3]
0000	0

B. Description of the FPGA

Given below is the FPGA designed for implementation of mod 16 counter, 4 bit adder and 8:3 encoder.

In total, 14 logic tiles and 22 switch boxes were used.

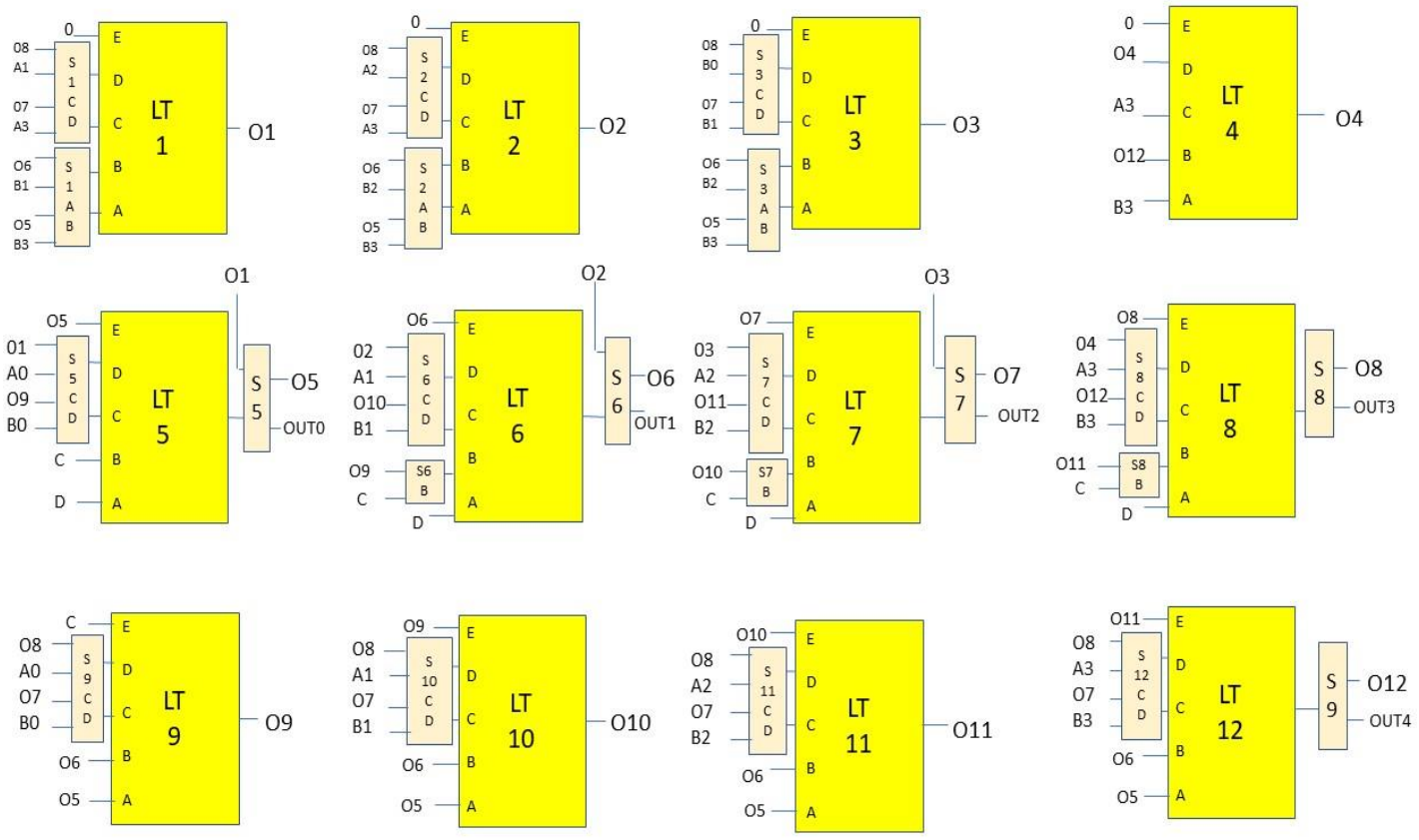

Figure 3- FPGA

Note- There are two extra logic tiles not shown here for latching c and d in the case of counter.

- Set of inputs to the FPGA-
  1. A[3:0] - 4 bit input. Labelled as A1, A2, A3, A4
  2. B[3:0] – 4 bit input. Labelled as B1, B2, B3, B4
  3. c – 1 bit input.
  4. d – 1 bit input.

c,d are available as C and D after passing through 2 separate logic tiles which latch them in the case of the counter.

5. clock – 1 bit input (clock for synchronous circuit)

- Set of outputs of FPGA
  1. out [4:0] – 5 bit output. These are labelled as OUT1, OUT2, OUT3, OUT4 in the circuit above.

Naming convention-

1. **Logic tile-**

a. The logic tiles are numbered 1 through 14. Logic tiles 1 to 12 are shown in figure 3, whereas 13 and 14 are omitted in the figure (reason mentioned above).
'i th' logic tile is named as 'LT i'.

b. The output of i th logic tile is labelled as 'O i'.

c. Input of each logic tile are labelled as A, B, C, D, E. A is the LSB and E the MSB.

```
logic_tile L1(L_out[0],clock,L1_in[0],L1_in[1],L1_in[2],L1_in[3],1'b0);
logic_tile L2(L_out[1],clock,L2_in[0],L2_in[1],L2_in[2],L2_in[3],1'b0);
logic_tile L3(L_out[2],clock,L3_in[0],L3_in[1],L3_in[2],L3_in[3],1'b0);
logic_tile L4(L_out[3],clock,L5_in[4],L6_in[4],L7_in[4],L8_in[4],1'b0);

logic_tile L5(L_out[4],clock,D,C,L5_in[2],L5_in[3],L5_in[4]);
logic_tile L6(L_out[5],clock,D,L6_in[1],L6_in[2],L6_in[3],L6_in[4]);
logic_tile L7(L_out[6],clock,D,L7_in[1],L7_in[2],L7_in[3],L7_in[4]);
logic_tile L8(L_out[7],clock,D,L8_in[1],L8_in[2],L8_in[3],L8_in[4]);

logic_tile L9(L_out[8],clock,L5_in[4],L6_in[4],L9_in[2],L9_in[3],C);
logic_tile L10(L_out[9],clock,L5_in[4],L6_in[4],L10_in[2],L10_in[3],L_out[8]);
logic_tile L11(L_out[10],clock,L5_in[4],L6_in[4],L11_in[2],L11_in[3],L_out[9]);
logic_tile L12(L12_in[0],clock,L5_in[4],L6_in[4],L12_in[2],L12_in[3],L_out[10]);

logic_tile L13_C(C,clock,c,1'b0,1'b0,1'b0,1'b0);
logic_tile L14_D(D,clock,d,1'b0,1'b0,1'b0,1'b0);
```

Figure 4- logic tile instantiations in lab6.v . Some naming conventions may differ, however they follow the same pattern as in this report.

2. **Switch box-**

a. The switch box attached to the output of a logic tile 'Oi', is named as 'Si'.

b. The switch box attached to

1. inputs ‘A’ and ‘B’ of the logic tiles ‘Oi’ are named as ‘Si ab’.
2. inputs ‘C’ and ‘D’ of the logic tiles ‘Oi’ are named as ‘Si cd’.
3. input ‘B’ of the of the logic tiles ‘Oi’ are named as ‘Si b’.

Naming of output pins of the switch boxes can be seen in figure 3. They do not follow any pattern as such.

```
switch_box_4x4 S5({SwDp[1:0],L5_in[4],out[0]}, {1'b0,1'b0,L_out[0],L_out[4]});
switch_box_4x4 S6({SwDp[3:2],L6_in[4],out[1]}, {1'b0,1'b0,L_out[1],L_out[5]});
switch_box_4x4 S7({SwDp[5:4],L7_in[4],out[2]}, {1'b0,1'b0,L_out[2],L_out[6]});
switch_box_4x4 S8({SwDp[7:6],L8_in[4],out[3]}, {1'b0,1'b0,1'b0,L_out[7]});
switch_box_4x4 S1cd({S1_t[3:2],L1_in[3],L1_in[2]},{L8_in[4],A[1],L7_in[4],A[3]});
switch_box_4x4 S1ab({S1_t[1:0],L1_in[1],L1_in[0]},{L6_in[4],B[1],L5_in[4],B[3]});
switch_box_4x4 S2cd({S2_t[3:2],L2_in[3],L2_in[2]},{L8_in[4],A[2],L7_in[4],A[3]});
switch_box_4x4 S2ab({S2_t[1:0],L2_in[1],L2_in[0]},{L6_in[4],B[2],L5_in[4],B[3]});
switch_box_4x4 S3cd({S3_t[3:2],L3_in[3],L3_in[2]},{L8_in[4],B[0],L7_in[4],B[1]});
switch_box_4x4 S3ab({S3_t[1:0],L3_in[1],L3_in[0]},{L6_in[4],B[2],L5_in[4],B[3]});
switch_box_4x4 S5dc({S5_t,L5_in[3],L5_in[2]},{L_out[0],A[0],L_out[8],B[0]});
switch_box_4x4 S6dc({S6_t,L6_in[3],L6_in[2]},{L_out[1],A[1],L_out[9],B[1]});
switch_box_4x4 S6b({S6b_t,L6_in[1]},{1'b0,1'b0,L_out[8],C});
switch_box_4x4 S7dc({S7_t,L7_in[3],L7_in[2]},{L_out[2],A[2],L_out[10],B[2]});
switch_box_4x4 S7b({S7b_t,L7_in[1]},{1'b0,1'b0,L_out[9],C});
switch_box_4x4 S8dc({S8_t,L8_in[3],L8_in[2]},{L_out[3],A[3],L_out[11],B[3]});
switch_box_4x4 S8b({S8b_t,L8_in[1]},{1'b0,1'b0,L_out[10],C});
switch_box_4x4 S9dc({S9_t,L9_in[3],L9_in[2]},{L_out[7],A[0],L_out[6],B[0]});
switch_box_4x4 S10dc({S10_t,L10_in[3],L10_in[2]},{L_out[7],A[1],L_out[6],B[1]});
switch_box_4x4 S11dc({S11_t,L11_in[3],L11_in[2]},{L_out[7],A[2],L_out[6],B[2]});
switch_box_4x4 S12dc({S12_t,L12_in[3],L12_in[2]},{L_out[7],A[3],L_out[6],B[3]});
switch_box_4x4 carry({ct,out[4],L_out[11]},{1'b0,1'b0,1'b0,L12_in[0]});
```

Figure 5- Switch box instantiations in lab6.v

Note- For the clarity in the code, S9 was labelled as ‘carry’. However, it will be referred to as S9 in this report.

Implementation description

Now, implementation of the three different applications is discussed here.

The active inputs and outputs in the circuits have been highlighted in cyan. Others are inactive/don't care inputs are outputs.

4 bit synchronous counter

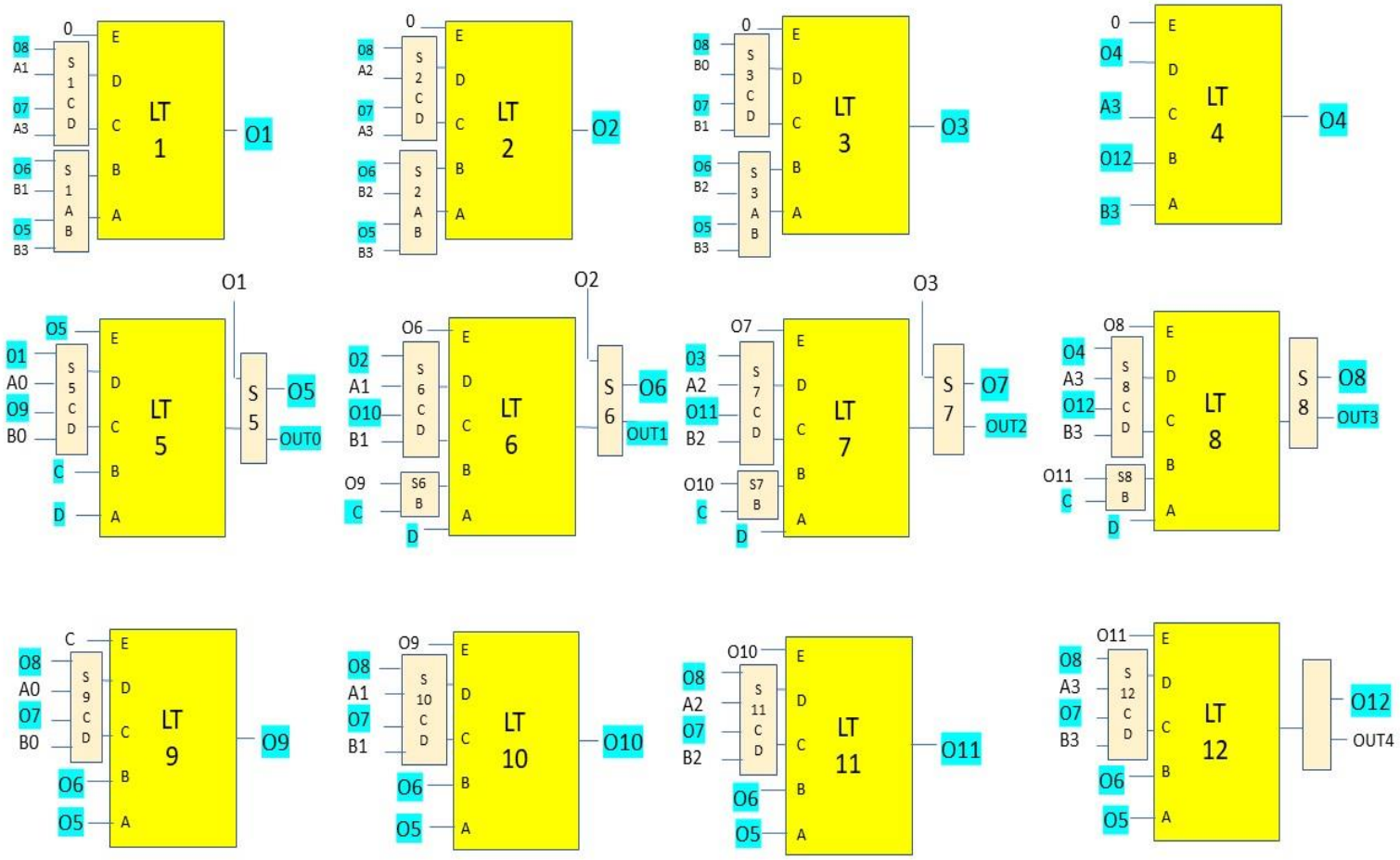

Figure 6 – Portion of the FPGA for the 4 bit synchronous counter

Working-

1. The counter is state machine, updating its state at every positive edge of the clock.
2. The 4 bit synchronous counter has 16 states, each corresponding to the numbers from 0 through 15.
3. Thus, 4 bits are required to store the current state of the machine. This can be done using 4 D-flipflops.
4. Two control inputs C and D are available. When CD==’10’, the counter increments. When CD==’01’, the counter decrements. When CD=’00’, the counter’s state does not change. CD cannot be ‘11’.

For a 4 bit synchronous counter, 4 flip flops store the state of the counter and a few logic gates are required for the implementing the logic of the predictors of the next state. This logic is implemented using LUTs.

Implementation-

Here the whole circuit is broken down into 3 sections to better explain the implementation.

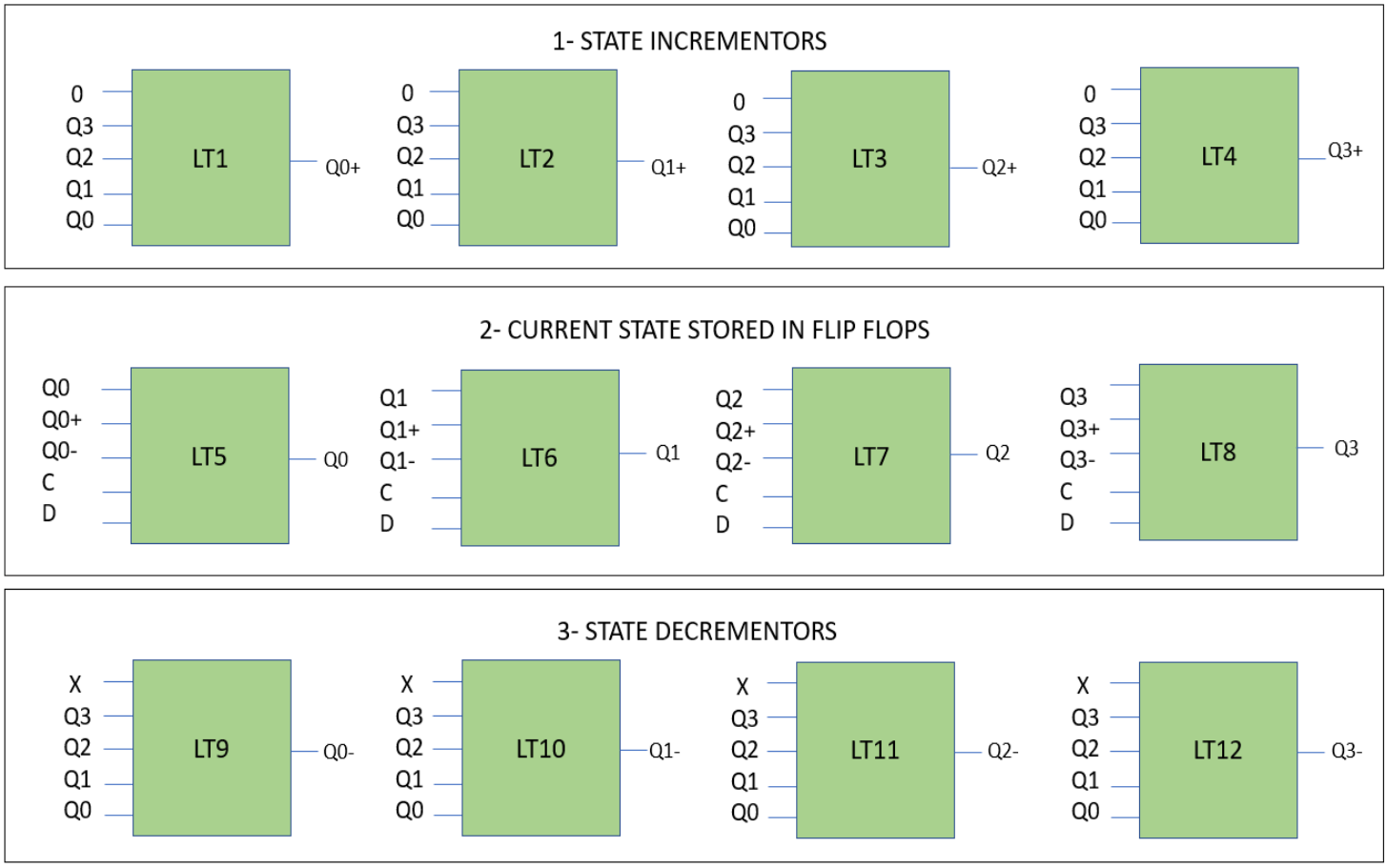

Figure 7

Being a 4 bit synchronous counter, the state of system is stored in 4 bits Q0, Q1, Q2, Q3, with Q0 being the LSB and Q3 the MSB.

- **Sections 2** consists of latched LT5, LT6, LT57, LT8. These logic tiles are latched in order to store the current state of the state machine, i.e Q0, Q1, Q2, Q3 respectively. The output of these logic tiles are the current state of the system Q0, Q1, Q2, Q3 respectively.

- **Section 1** consists of non-latched LT1, LT2, LT3, LT4 which find the next state of the system, that is Q0+, Q1+, Q2+, Q3+ respectively based on the values of Q0, Q1, Q2, Q3.

- **Section 3** consists of non-latched LT9, LT10, LT11, LT12 which find the previous state of the system, that is Q0-, Q1-, Q2-, Q3- respectively based on the values of Q0, Q1, Q2, Q3.

- Each of the logic tile in sections 2 corresponding to bit 'i' receive 5 inputs, that are the same state Qi, the next state Qi+, the previous state Qi-, and the two control bits C and D

Based on the values of C and D, one of those values are stored in the flipflops, which then becomes the new state of the system.

If

a. If CD=='00', Qi is stored (no change in state)
b. If CD=='10', Qi+ is stored (incremented)
c. If CD=='01', Qi- is stored (decremented)

Memory configuration-

The memory configuration of the Logic tiles and switch boxes are given in hexadecimal format.

- Configuration of LUTs-

Logic tile	Configuration(mem[31:0])	mem[32] (latched or not)
LT1	00005555	0
LT2	00006666	0
LT3	00007878	0
LT4	00007F80	0
LT5	75316420	1
LT6	75316420	1
LT7	75316420	1
LT8	75316420	1
LT9	55555555	0
LT10	99999999	0
LT11	E1E1E1E1	0
LT12	FE01FE01	0
LT13	00000002	1
LT14	00000002	1

- Configuration of switch boxes-

Switch box	Configuration(configure[15:0])
S5	0011
S6	0011
S7	0011
S8	0011
S1CD	0082
S1AB	0082
S2CD	0082
S2AB	0082
S3CD	0082
S3AB	0082
S5CD	0082
S6CD	0082
S6B	0011
S7CD	0082
S7B	0001
S8CD	0082
S8B	0001
S9CD	0082

S10CD	0082
S11CD	0082
S12CD	0082
S9	0001

4-bit Adder

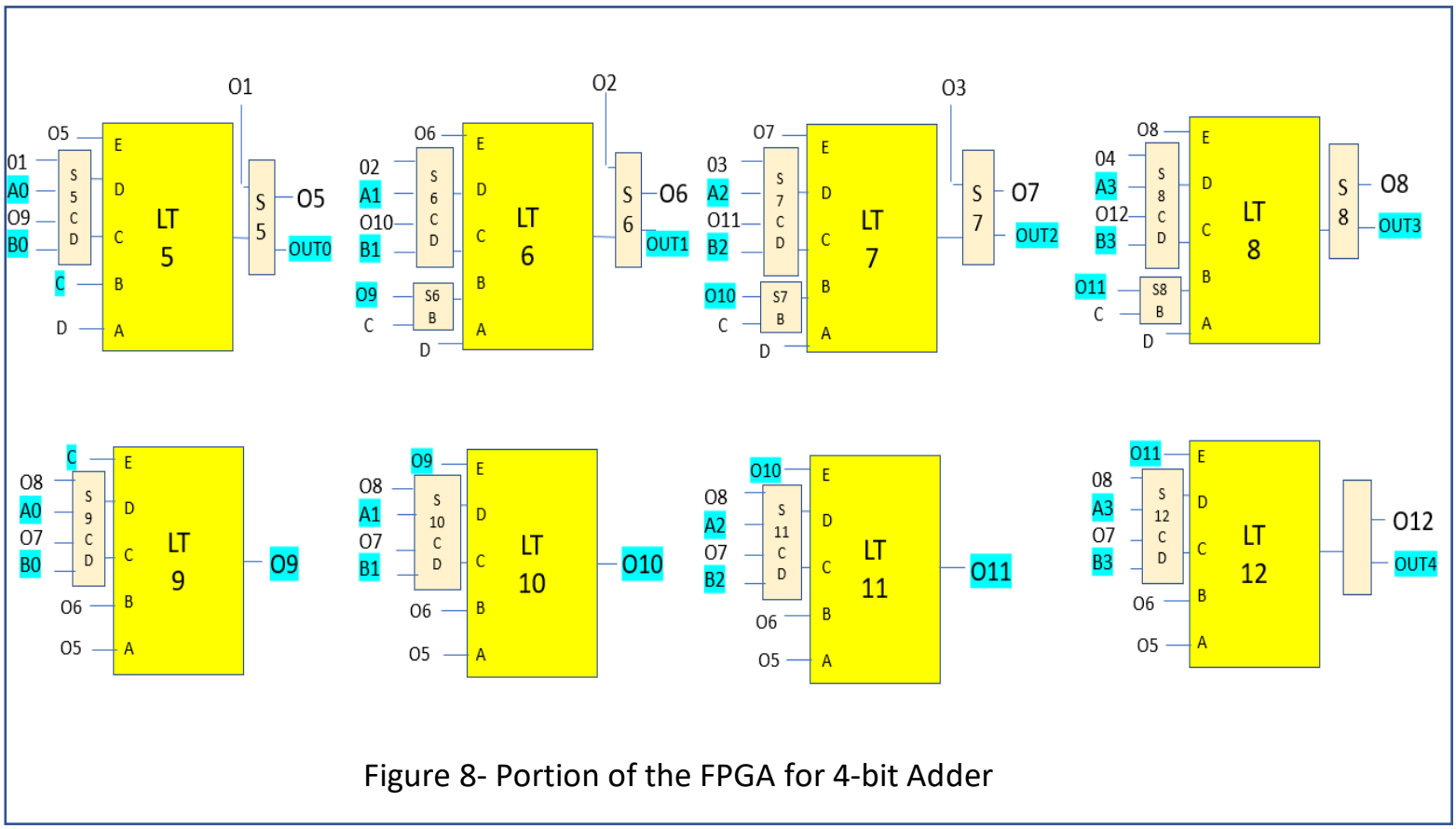

Figure 8- Portion of the FPGA for 4-bit Adder

Note- LT1, LT2, LT3 and LT4 have been omitted for clarity

Objective- Design a 4-bit adder that adds two 4-bit numbers.

Inputs-

1. 2 4-bit numbers {A3,A2,A1,A0} and {B3,B2,B1,B0}.
2. Carry in (CIN) – ‘c’

Outputs-

1. 4 bit sum- {OUT3,OUT2,OUT1,OUT0}
2. Carry out (COUT) - OUT4.

Implementation description-

This 4 bit adder is a combinational circuit, based on the principle of ripple adder.

Here the circuit in figure 8 is broken down into 2 sections for explanation.

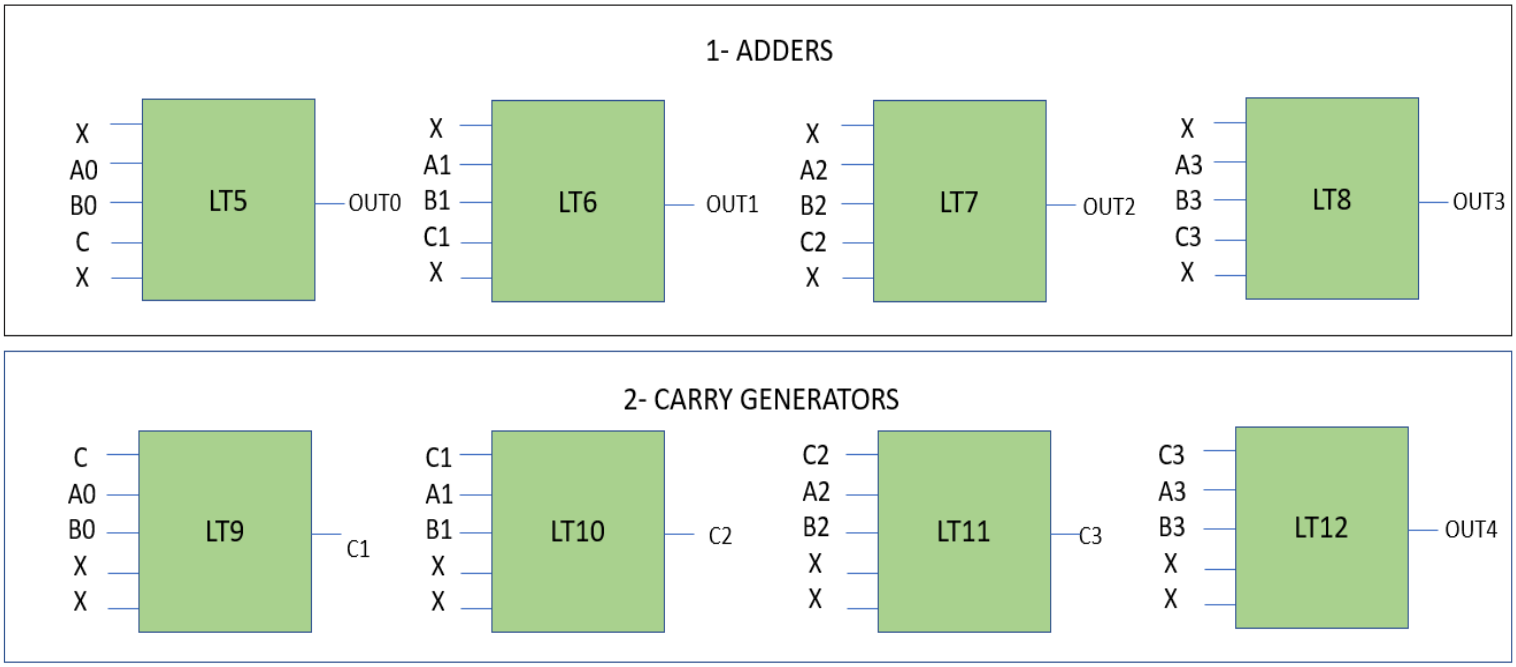

Figure 9

Logic tiles in section 1 are full adders. Each one of them receive inputs Ai, Bi and Ci and give the sum, or the XOR as 'OUTi'. C0 is the input CIN denoted as C in the above figure.

Logic tile in section 2 are carry generators. They take inputs Ai, Bi and Ci and generate the carry 'C_{i+1}'. C4 or carry-out COUT, is denoted as OUT4 in the above figure.

Don't care inputs to LUTs are marked as 'X'.

Memory configuration-

The memory configuration of the Logic tiles and switch boxes are given in hexadecimal format.

- Configuration of LUTs-

Logic tile	Configuration(mem[31:0])	mem[32] (latched or not)
LT1(not in use)	00000000 (Don't care)	0
LT2(not in use)	00000000 (Don't care)	0
LT3(not in use)	00000000 (Don't care)	0
LT4(not in use)	00000000 (Don't care)	0
LT5	C33CC33C	0
LT6	C33CC33C	0
LT7	C33CC33C	0
LT8	C33CC33C	0
LT9	FFF0F000	0
LT10	FFF0F000	0
LT11	FFF0F000	0
LT12	FFF0F000	0
LT13	00000002	0
LT14(not in use)	00000000 (Don't care)	0

- Configuration of switch boxes-

Switch box	Configuration(configure[15:0])
S5	0001
S6	0001
S7	0001
S8	0001
S1CD(not in use)	0000
S1AB(not in use)	0000
S2CD(not in use)	0000
S2AB(not in use)	0000
S3CD(not in use)	0000
S3AB(not in use)	0000
S5CD	0041
S6CD	0041
S6B	0002
S7CD	0041
S7B	0002
S8CD	0041
S8B	0002
S9CD	0041

S10CD	0041
S11CD	0041
S12CD	0041
S9	0010

8:3 Encoder

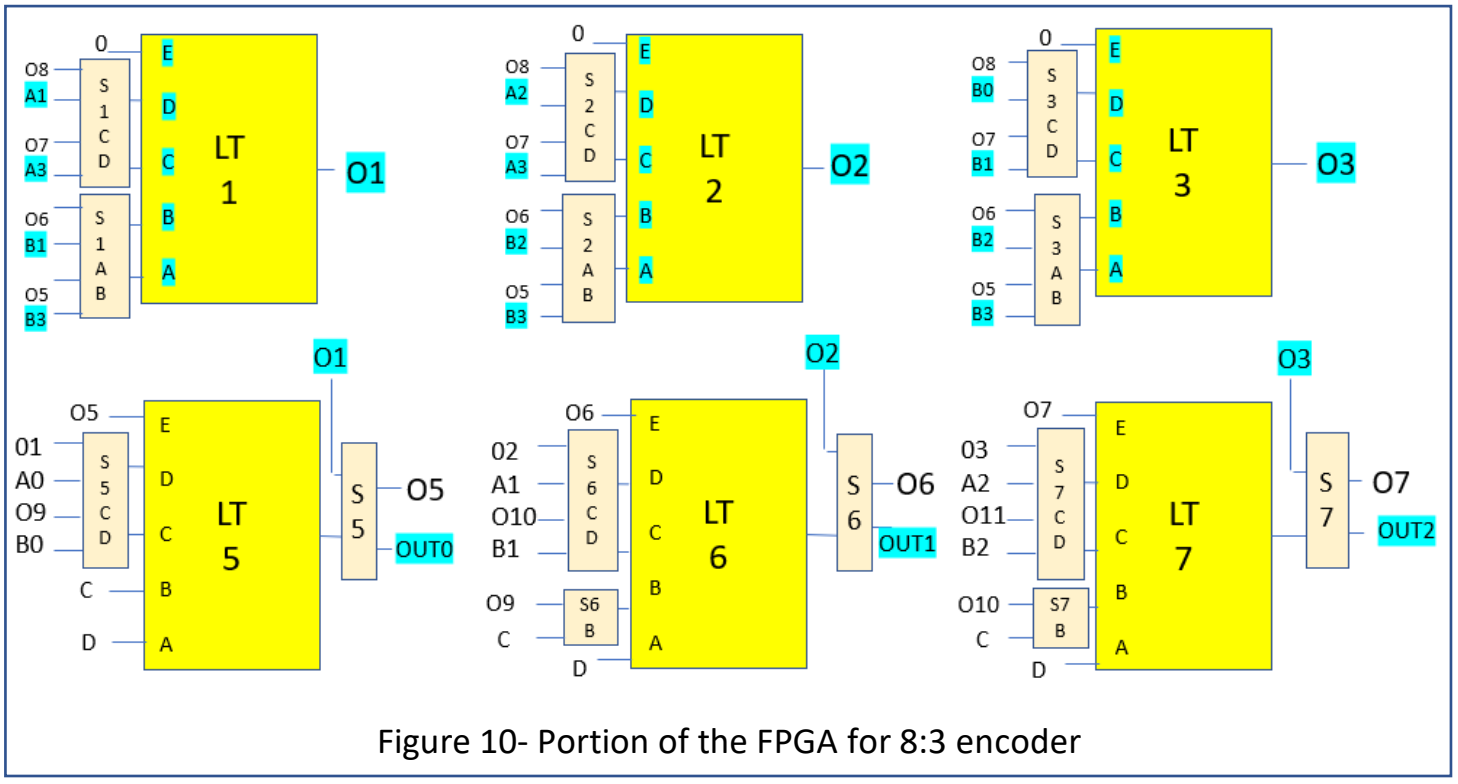

Figure 10- Portion of the FPGA for 8:3 encoder

Objective- Implement an 8:3 encoder, such that only one of the inputs are high at one time.

Input-

8-bit input {B3,B2,B1,B0,A3,A2,A1,A0}. A0 is the LSB and B3 is the MSB.

Output-

3-bit output {OUT2, OUT1, OUT0}. OUT0 is the LSB and OUT2 the MSB.

Input-output mapping-

Input {B3,B2,B1,B0,A3,A2,A1,A0}	Output {OUT2, OUT1, OUT0}
00000001	000
00000010	001
00000100	010
00001000	011
00010000	100
00100000	101
01000000	110
10000000	111

Implementation description-

Logic tiles LT1, LT2, LT3 work as the logic for the output(described below).

Their outputs are connected to switch boxes S5, S6, S7 respectively to assign their outputs to OUT0, OUT1 and OUT2 respectively.

1. LT1 generates OUT0. OUT0 = (A1+A3+B1+B3).
2. LT2 generates OUT1. OUT1 = (A2+A3+B1+B3).
3. LT3 generates OUT2. OUT2 = (B0+B1+B2+B3).

Note- Logic tiles LT5, LT6, and LT7 have no utilisation in this circuit and are shown in figure 10 just because switch boxes S5, S6 and S7 connected to them are in utilisation.

Memory configuration

- Configuration of LUTs-

Logic tile	Configuration(mem[31:0])	mem[32] (latched or not)
LT1	FFFEFFFE	0
LT2	FFFEFFFE	0
LT3	FFFEFFFE	0
LT4 (not in use)	00000000 (Don't care)	0
LT5 (not in use)	00000000 (Don't care)	0
LT6 (not in use)	00000000 (Don't care)	0
LT7 (not in use)	00000000 (Don't care)	0
LT8 (not in use)	00000000 (Don't care)	0
LT9 (not in use)	00000000 (Don't care)	0
LT10 (not in use)	00000000 (Don't care)	0
LT11 (not in use)	00000000 (Don't care)	0
LT12 (not in use)	00000000 (Don't care)	0
LT13 (not in use)	00000000 (Don't care)	0

LT14 (not in use)	00000000 (Don’t care)	0

- Configuration of switch boxes-

Switch box	Configuration(configure[15:0])
S5	0002
S6	0002
S7	0002
S8(not in use)	0000
S1CD	0041
S1AB	0041
S2CD	0041
S2AB	0041
S3CD	0041
S3AB	0041
S5CD	0041
S6CD (not in use)	0000
S6B (not in use)	0000
S7CD (not in use)	0000
S7B (not in use)	0000
S8CD (not in use)	0000
S8B (not in use)	0000
S9CD (not in use)	0000
S10CD (not in use)	0000
S11CD (not in use)	0000
S12CD (not in use)	0000
S9 (not in use)	0000