COMPUTER AIDED DIGITAL DESIGN

AIM: Decodes a 3-Bit signal and asserts one bit of 8-Bit output. (using a case statement.)

Hardware used:

INTEL FPGA DE10-LITE, MAX1010M50DAF4484C7G

PROCEDURE:

Step1: Create a Quartus Project using system verilog HDL model.

Step2: Simulate the design using Quartus lite edition simulator.

Step3: Construct a testbench for the problem statement.

Step4: Synthesize the design and observe the Schematic.

Step5: Implement the design.

Step6: Generate the bitstream.

Step7: Configure the FPGA using the generated bitstream and verify the functionality in hardware

Introduction:

Decoder is a digital circuit which takes some inputs and decodes it and provides a decoded output. We know that every bit in digital can take 2 values, either 0 or 1. Hence, if I have N inputs to decode, I will get a maximum of 2^N outputs. Hence, Decoders are characterized by their sizes which are written in the form (N x 2^N) for an N- bit Decoder. 3 x 8 decoder has 3 input and 8 decoded output pins.

There are several types of binary decoders, but in all cases, a decoder is an electronic circuit with multiple inputs and multiple output signals, which converts every unique combination of input states to a specific combination of output states. In addition to integer data inputs, some decoders also have one or more “enable” inputs. When the enable input is negated (disabled), all decoder outputs are forced to their inactive states.

Why Do We Need a Decoder?

Decoding refers to the process in which the decoder **decodes or interprets a message that has been encoded by a source using his experiences and intellect**. The message has been kept simple and basic. As a result, the decoder will be able to quickly and easily decode the received message and send it back to the source.

Truth Table of 3:8 decoder:

A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

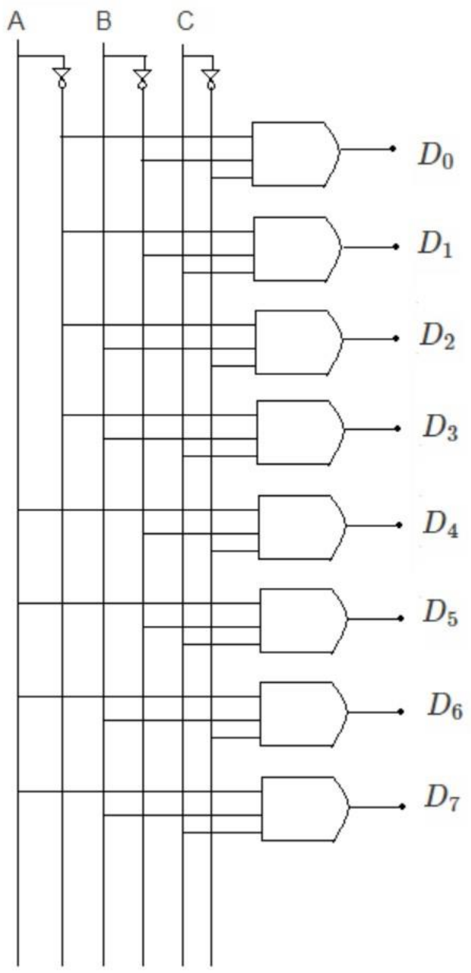
Boolean expression:

$$D_0 = \bar{A}\overline{B}\overline{C}, \quad D_1 = \bar{A}\overline{B}C, \quad D_2 = \bar{A}B\overline{C},$$

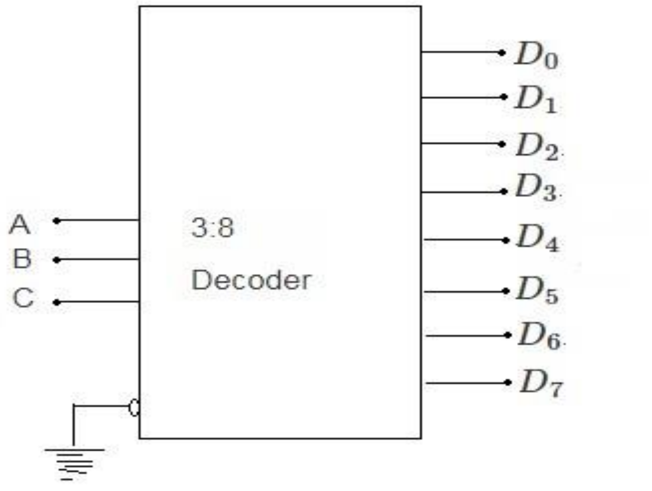
$$D_3 = \overline{A}BC, \quad D_4 = A\overline{B}\overline{C}, \quad D_5 = A\overline{B}C,$$

$$D_6 = AB\overline{C}, \quad D_7 = ABC$$

Logic Diagram of 3:8 decoder:



External View:



SYSTEM VERILOG CODE FOR 3:8 DECODER:

```
module decoder3_to_8( in,out, en);

input [2:0]  in; input en; output [7:0]

out;   reg [7:0] out;
```

```
 always @( in or en)
           begin
      if (en)
begin
out=8'd0;
case (in)
               3'b000: out[0]=1'b1;
               3'b001: out[1]=1'b1;
               3'b010: out[2]=1'b1;
               3'b011: out[3]=1'b1;
               3'b100: out[4]=1'b1;
               3'b101: out[5]=1'b1;
               3'b110: out[6]=1'b1;
3'b111: out[7]=1'b1;
default: out=8'd0;
endcase       end else
out=8'd0; end endmodule
```

TESTBENCH:

```
module decoder_tb();
wire [7:0] out; reg en;
reg [2:0] in; integer i;

decoder3_to_8 dut(in,out,en);
initial  begin
```

```
 $monitor( "en=%b, in=%d, out=%b ", en, in, out);
for ( i=0; i<16; i=i+1)         begin
          {en,in}  = i;
#1;        end end
endmodule
```

($monitor is a “system task” provided by Verilog itself for generating input and output to help verification.

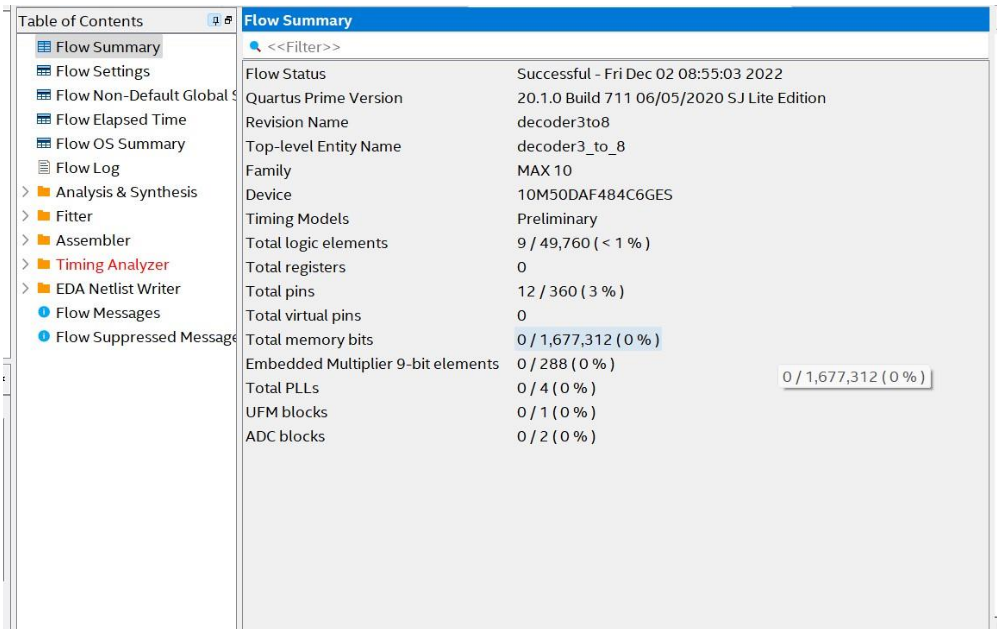
System tasks are generally not used(or ignored)by the synthesis tools.

A monitor statement has the syntax of

$monitor(‘format string’,parameter1,parameter2…);

$monitor displays the values of its parameters EVERY time ANY of its parameter changes value.)

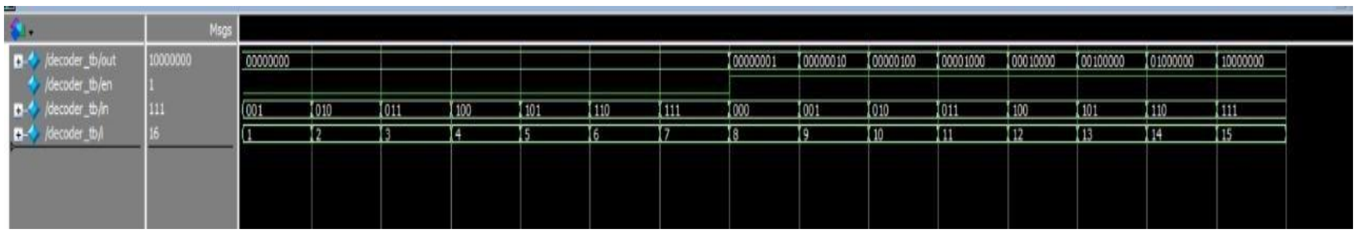
Compilation Report:



Flow Summary

Flow Status	Successful - Fri Dec 02 08:55:03 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	decoder3to8
Top-level Entity Name	decoder3_to_8
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	9 / 49,760 (< 1 %)
Total registers	0
Total pins	12 / 360 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

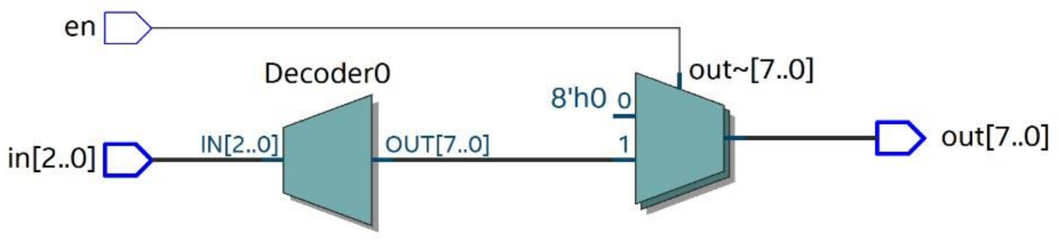
Simulation :



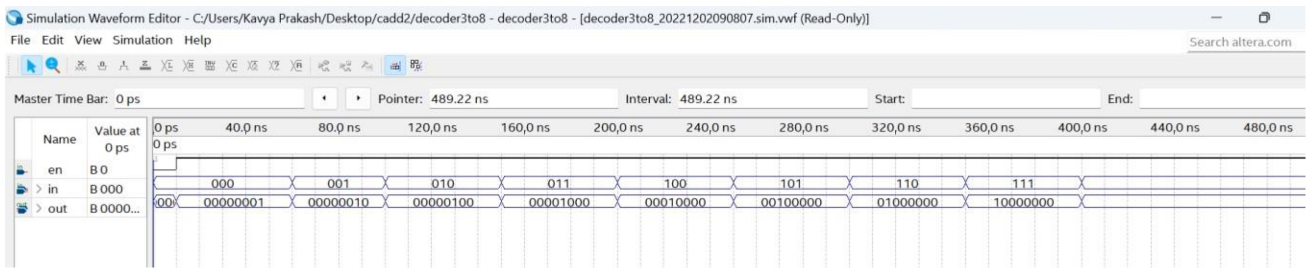
The simulation result hence shows the output when enable is 1 and how for an input 000 the y[0] is 1, for 001 the y[1] is 1, for 010 the y[1] is 1 and so on until vale of 111.

RTL View:

By using enable,as follows:



Waveform :



Simulation Waveform Editor - C:/Users/Kavya Prakash/Desktop/cadd2/decoder3to8 - decoder3to8 - [decoder3to8_20221202090807.sim.vwf (Read-Only)]

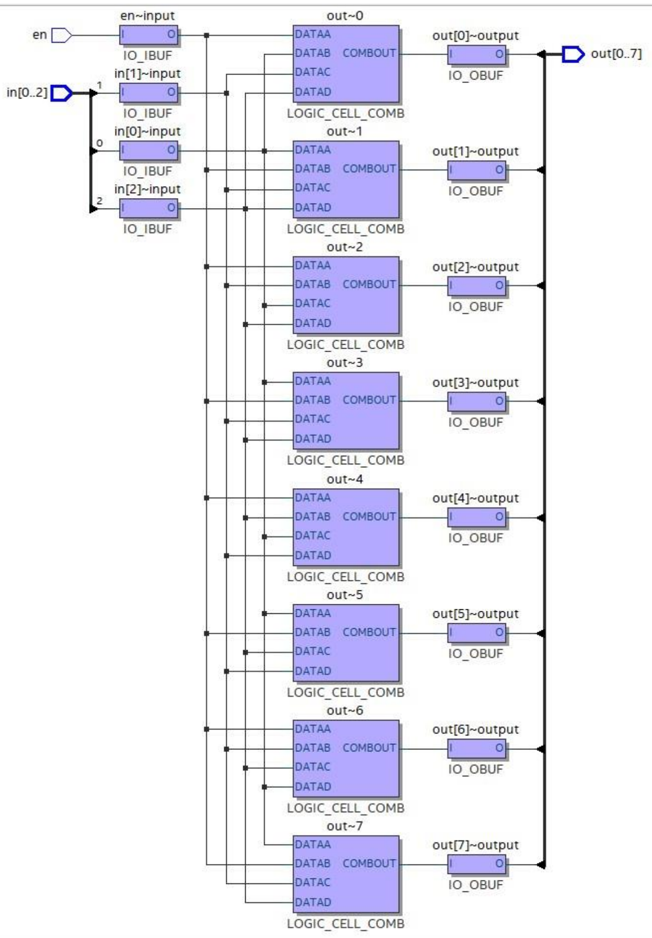
Master Time Bar: 0 ps   Pointer: 489.22 ns   Interval: 489.22 ns   Start: End:

Name	Value at 0 ps									
en	B 0									
in	B 000		000	001	010	011	100	101	110	111
out	B 0000...	00	00000001	00000010	00000100	00001000	00010000	00100000	01000000	10000000

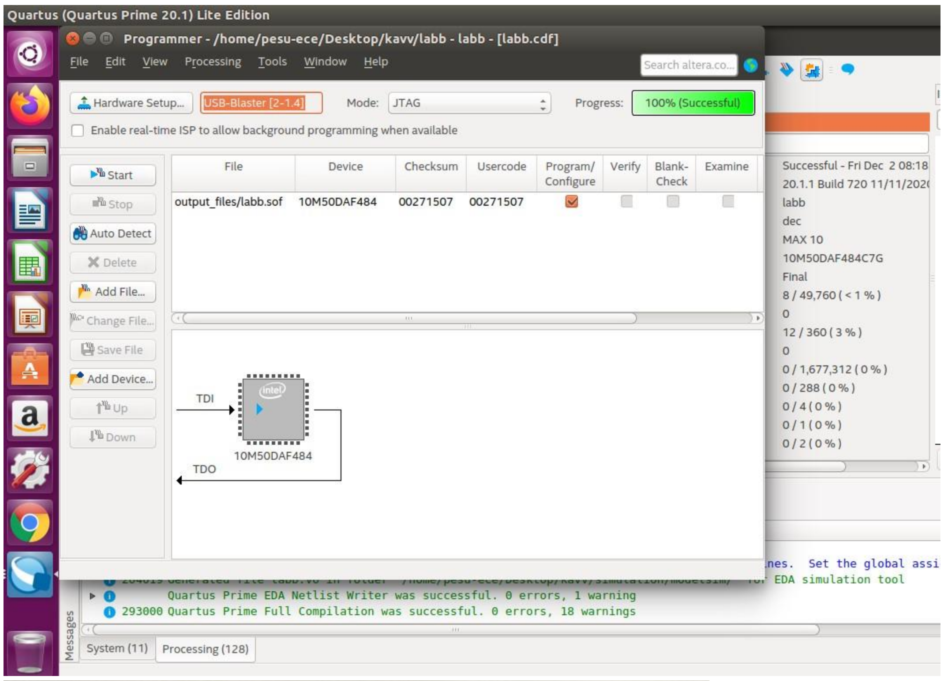
```
0 ps  40.0 ns  80.0 ns  120,0 ns  160,0 ns  200,0 ns  240,0 ns  280,0 ns  320,0 ns  360,0 ns  400,0 ns  440,0 ns  480,0 ns
```

The waveform found matches with the truth table of 3:8 dcecoder when enable is set to 1 as expected.

Post-Synthesis(Mapping) Schematic:



Hardware implementation:



Quartus (Quartus Prime 20.1) Lite Edition

Programmer - /home/pesu-ece/Desktop/kavv/labb - labb - [labb.cdf]

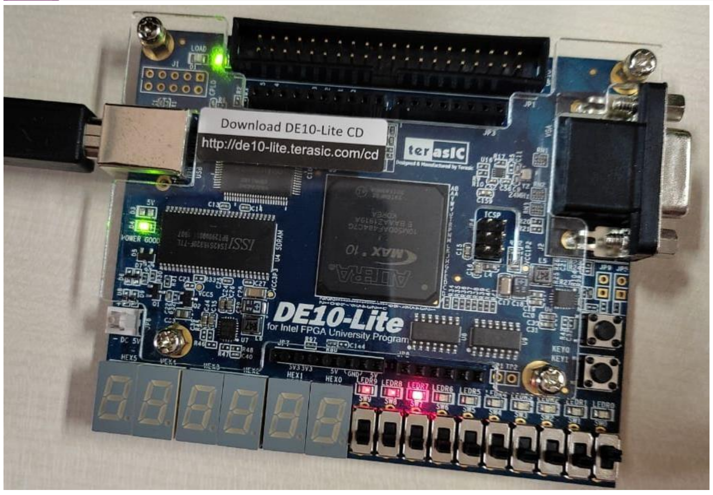
USB-Blaster [2-1.4]   Mode: JTAG   Progress: 100% (Successful)

Enable real-time ISP to allow background programming when available

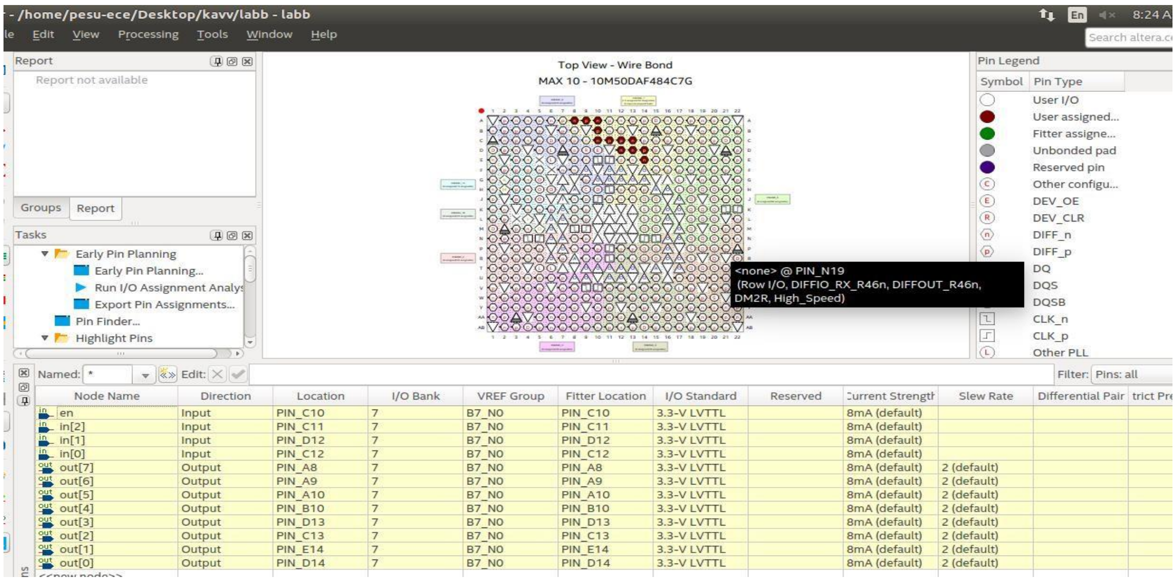
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/labb.sof	10M50DAF484	00271507	00271507	✓			

```
Successful - Fri Dec  2 08:18
20.1.1 Build 720 11/11/2020
labb
dec
MAX 10
10M50DAF484C7G
Final
8 / 49,760 ( < 1 % )
0
12 / 360 ( 3 % )
0
0 / 1,677,312 ( 0 % )
0 / 288 ( 0 % )
0 / 4 ( 0 % )
0 / 1 ( 0 % )
0 / 2 ( 0 % )
```

```
nes.  Set the global assi
for EDA simulation tool
   Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
293000 Quartus Prime Full Compilation was successful. 0 errors, 18 warnings
```



Download DE10-Lite CD
http://de10-lite.terasic.com/cd
terasic
DE10-Lite
for Intel FPGA University Program
MAX 10
ALTERA



- /home/pesu-ece/Desktop/kavv/labb - labb

<none> @ PIN_N19 (Row I/O, DIFFIO_RX_R46n, DIFFOUT_R46n, DM2R, High_Speed)

Named: * Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	trict Pre
en	Input	PIN_C10	7	B7_N0	PIN_C10	3.3-V LVTTL		8mA (default)			
in[2]	Input	PIN_C11	7	B7_N0	PIN_C11	3.3-V LVTTL		8mA (default)			
in[1]	Input	PIN_D12	7	B7_N0	PIN_D12	3.3-V LVTTL		8mA (default)			
in[0]	Input	PIN_C12	7	B7_N0	PIN_C12	3.3-V LVTTL		8mA (default)			
out[7]	Output	PIN_A8	7	B7_N0	PIN_A8	3.3-V LVTTL		8mA (default)	2 (default)		
out[6]	Output	PIN_A9	7	B7_N0	PIN_A9	3.3-V LVTTL		8mA (default)	2 (default)		
out[5]	Output	PIN_A10	7	B7_N0	PIN_A10	3.3-V LVTTL		8mA (default)	2 (default)		
out[4]	Output	PIN_B10	7	B7_N0	PIN_B10	3.3-V LVTTL		8mA (default)	2 (default)		
out[3]	Output	PIN_D13	7	B7_N0	PIN_D13	3.3-V LVTTL		8mA (default)	2 (default)		
out[2]	Output	PIN_C13	7	B7_N0	PIN_C13	3.3-V LVTTL		8mA (default)	2 (default)		
out[1]	Output	PIN_E14	7	B7_N0	PIN_E14	3.3-V LVTTL		8mA (default)	2 (default)		
out[0]	Output	PIN_D14	7	B7_N0	PIN_D14	3.3-V LVTTL		8mA (default)	2 (default)		

Conclusion:

Decoders are combinational circuits implemented with the help of logic gates.

The main idea behind using them is to save space occupied by data and reduce the number of wires required to implement circuits.

Decoders can hence be widely used in speed synchronisation of multiple motors in industries, war field flying robot with a night vision flying camera, robotic vehicle with metal detector, home automation systems, automatic health monitoring systems etc.

Future Scope:

Although we can always model a decoder with simple components that is without an enable we still made use of enable because a Decoder with Enable input can function as a demultiplexer.

A demultiplexer is a circuit that receives information from a single line and directs it to one of possible output lines.

Demux obtained henceforth is used in a lot of digital circuits and electronic applications.

Decoders are usually used along with encoders in various applications as mentioned above.

Source code:

```
module
decoder3_to_8( in,out, en); input
[2:0] in; input en; output [7:0] out;
reg [7:0] out;

 always @( in or en)
```

```
          begin
       if (en)          begin
out=8'd0;            case (in)
3'b000: out[0]=1'b1;
              3'b001: out[1]=1'b1;
              3'b010: out[2]=1'b1;
              3'b011: out[3]=1'b1;
              3'b100: out[4]=1'b1;
              3'b101: out[5]=1'b1;
              3'b110: out[6]=1'b1;
3'b111: out[7]=1'b1;
default: out=8'd0;
endcase       end else
out=8'd0; end endmodule
```