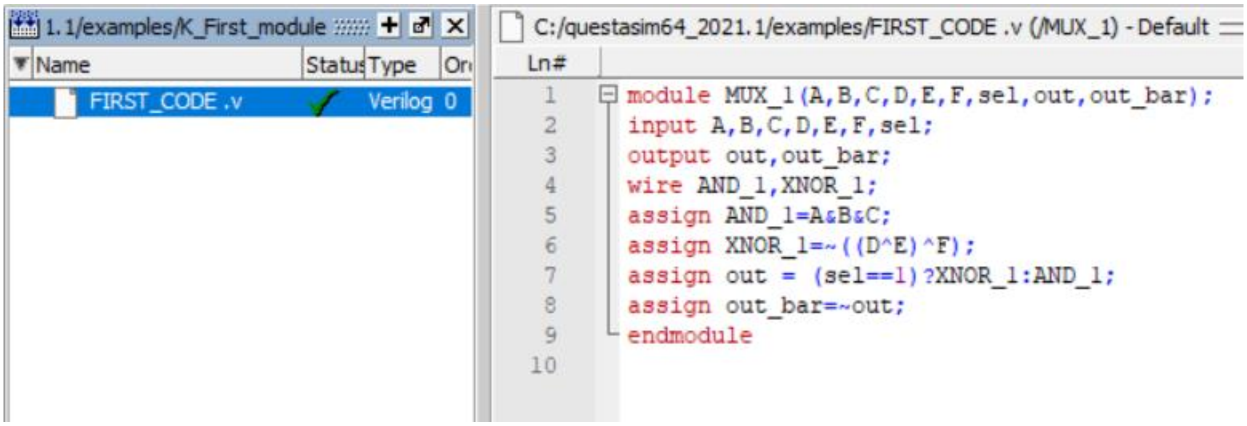
1)Logic circuit with Mux

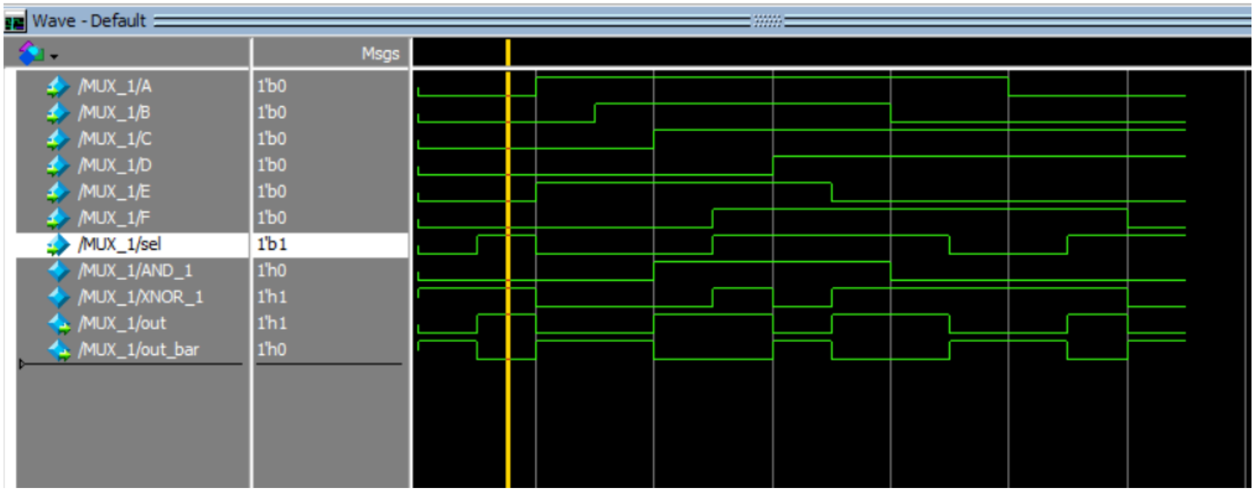


1.1/examples/K_First_module

Name	Status	Type	Or
FIRST_CODE .v	✓	Verilog	0

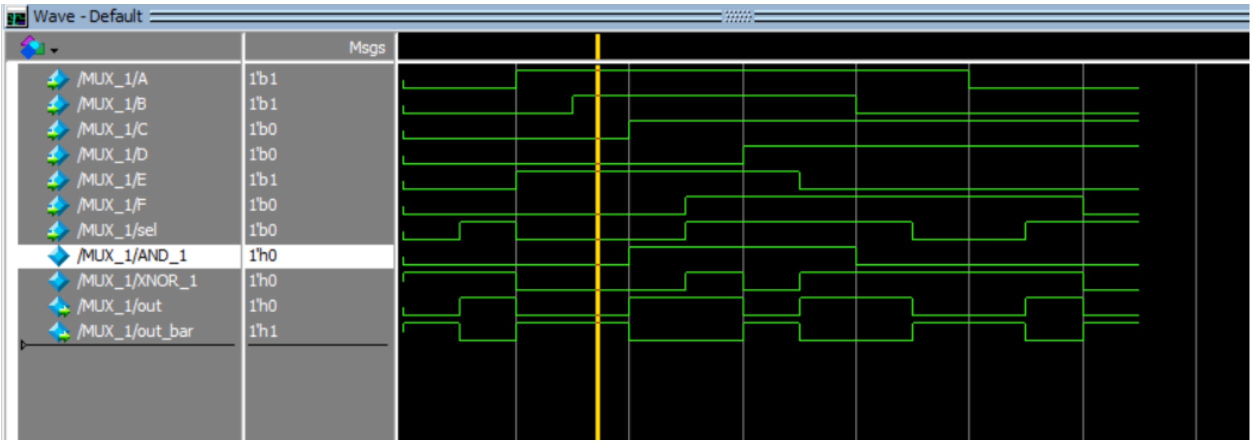
C:/questasim64_2021.1/examples/FIRST_CODE .v (/MUX_1) - Default

```
module MUX_1(A,B,C,D,E,F,sel,out,out_bar);
input A,B,C,D,E,F,sel;
output out,out_bar;
wire AND_1,XNOR_1;
assign AND_1=A&B&C;
assign XNOR_1=~((D^E)^F);
assign out = (sel==1)?XNOR_1:AND_1;
assign out_bar=~out;
endmodule
```



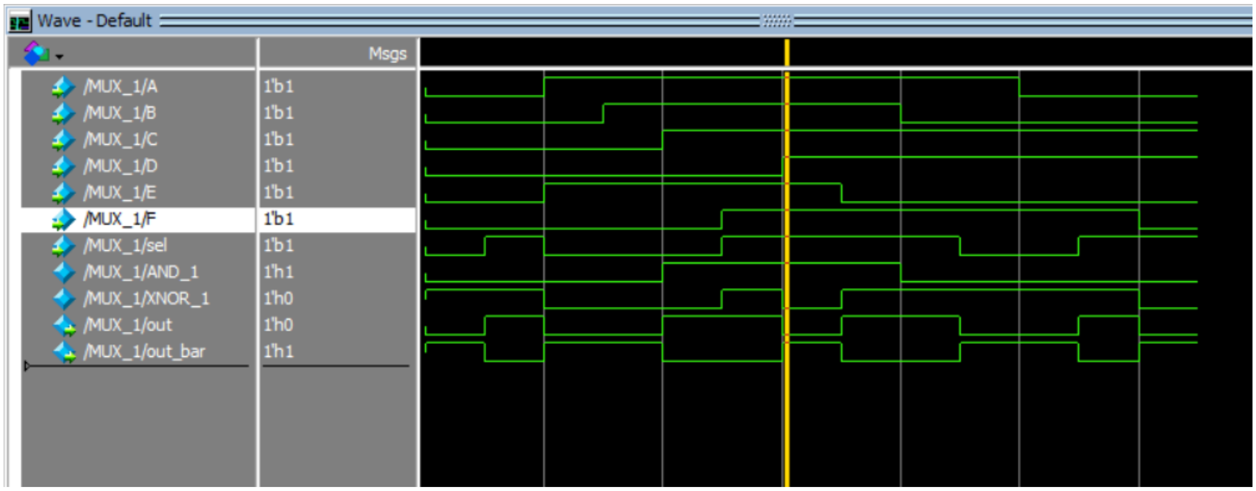
Wave - Default

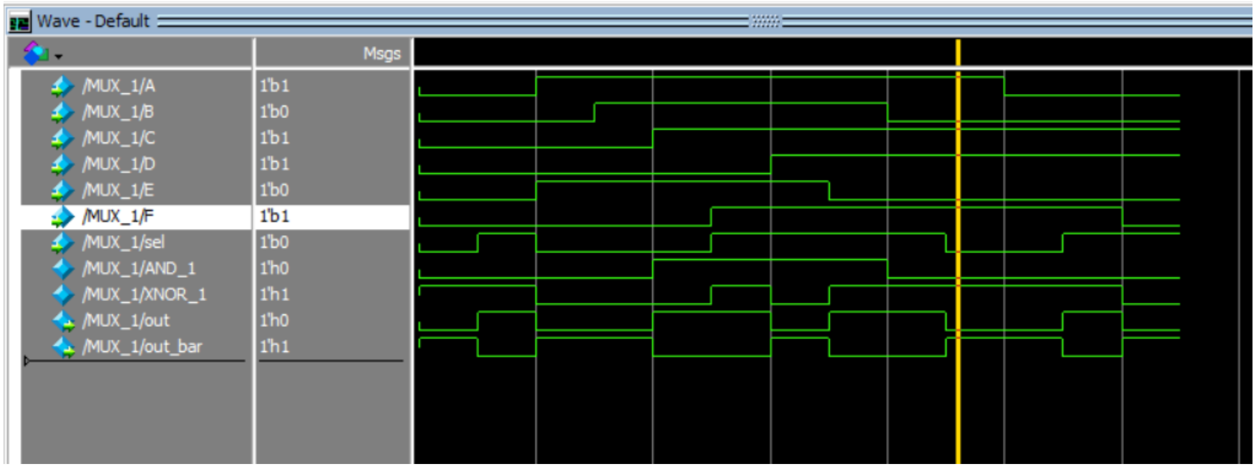
	Msgs
/MUX_1/A	1'b0
/MUX_1/B	1'b0
/MUX_1/C	1'b0
/MUX_1/D	1'b0
/MUX_1/E	1'b0
/MUX_1/F	1'b0
/MUX_1/sel	1'b1
/MUX_1/AND_1	1'h0
/MUX_1/XNOR_1	1'h1
/MUX_1/out	1'h1
/MUX_1/out_bar	1'h0



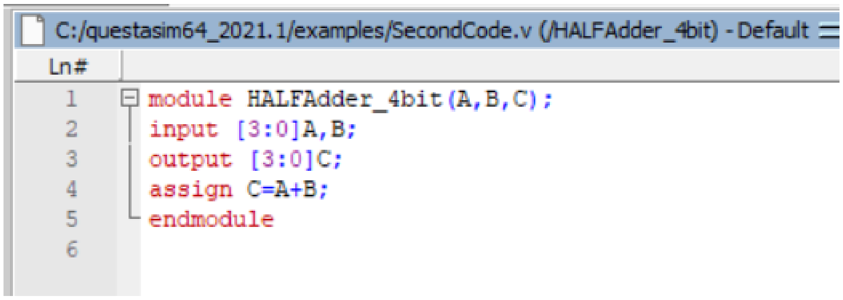
Wave - Default

	Msgs
/MUX_1/A	1'b1
/MUX_1/B	1'b1
/MUX_1/C	1'b0
/MUX_1/D	1'b0
/MUX_1/E	1'b1
/MUX_1/F	1'b0
/MUX_1/sel	1'b0
/MUX_1/AND_1	1'h0
/MUX_1/XNOR_1	1'h0
/MUX_1/out	1'h0
/MUX_1/out_bar	1'h1

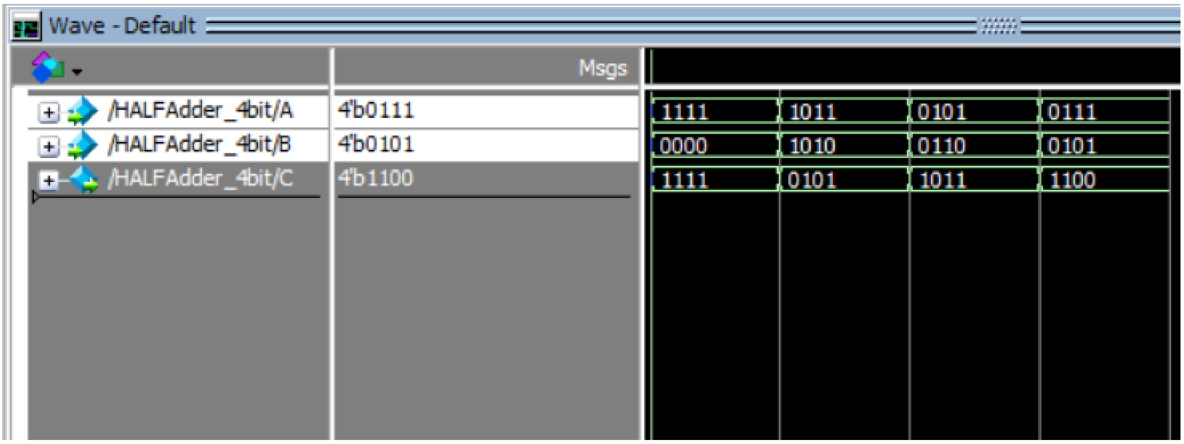




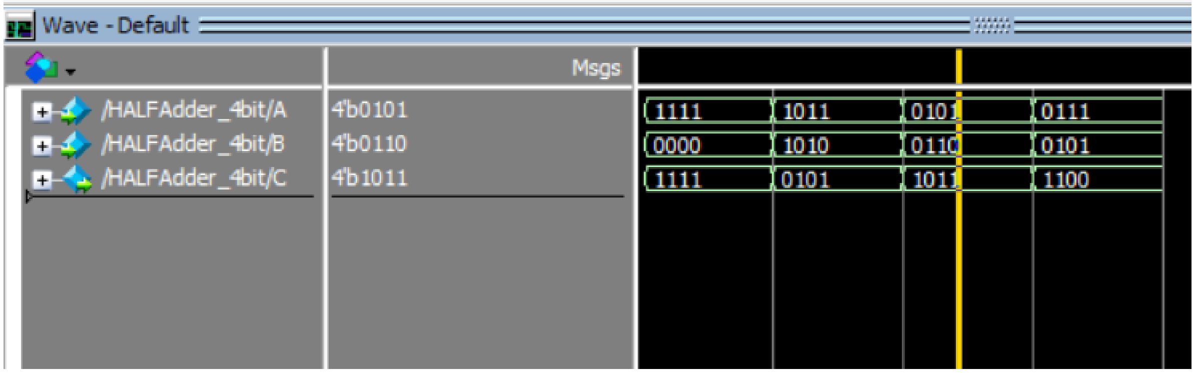
2) Implement 4-bit adder using addition operator


C:/questasim64_2021.1/examples/SecondCode.v (/HALFAdder_4bit) - Default
Ln#

```
module HALFAdder_4bit(A,B,C);
input [3:0]A,B;
output [3:0]C;
assign C=A+B;
endmodule
```

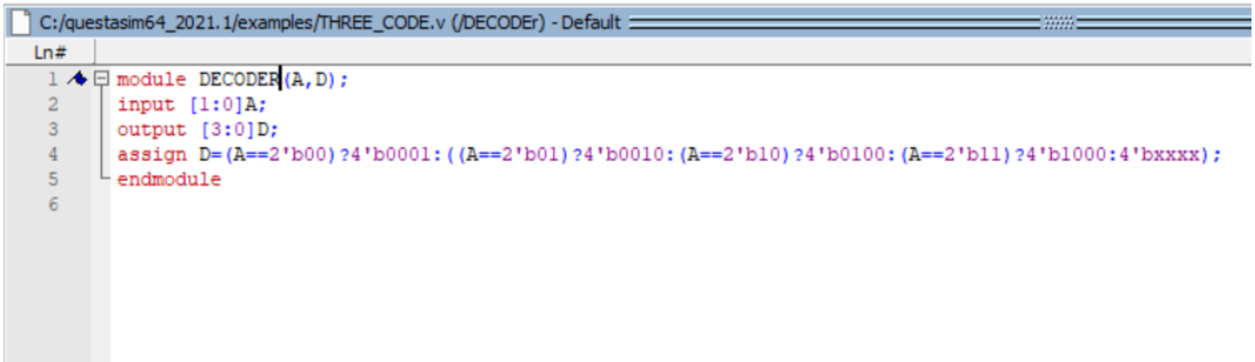


/HALFAdder_4bit/A 4'b0111
/HALFAdder_4bit/B 4'b0101
/HALFAdder_4bit/C 4'b1100

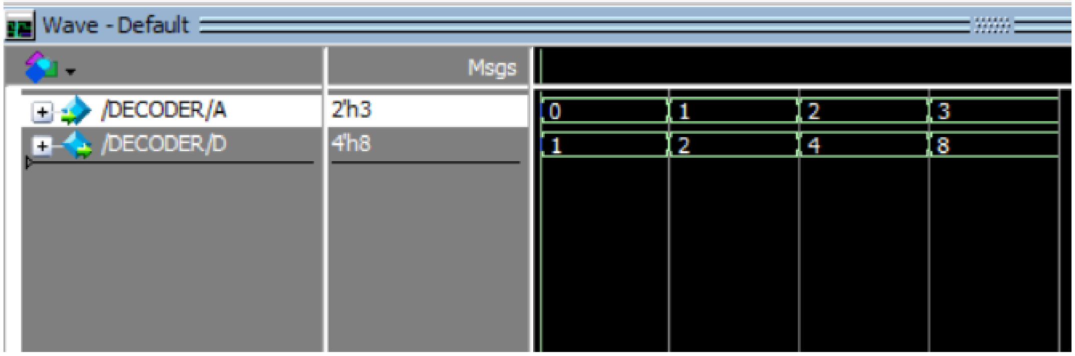


/HALFAdder_4bit/A 4'b0101
/HALFAdder_4bit/B 4'b0110
/HALFAdder_4bit/C 4'b1011

3) Implement 2-to-4 Decoder using conditional operator

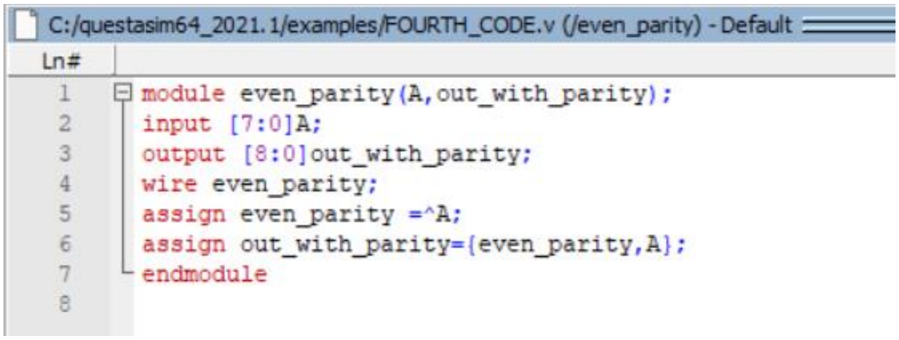

C:/questasim64_2021.1/examples/THREE_CODE.v (/DECODEr) - Default
Ln#

```
module DECODER(A,D);
input [1:0]A;
output [3:0]D;
assign D=(A==2'b00)?4'b0001:((A==2'b01)?4'b0010:(A==2'b10)?4'b0100:(A==2'b11)?4'b1000:4'bxxxx);
endmodule
```

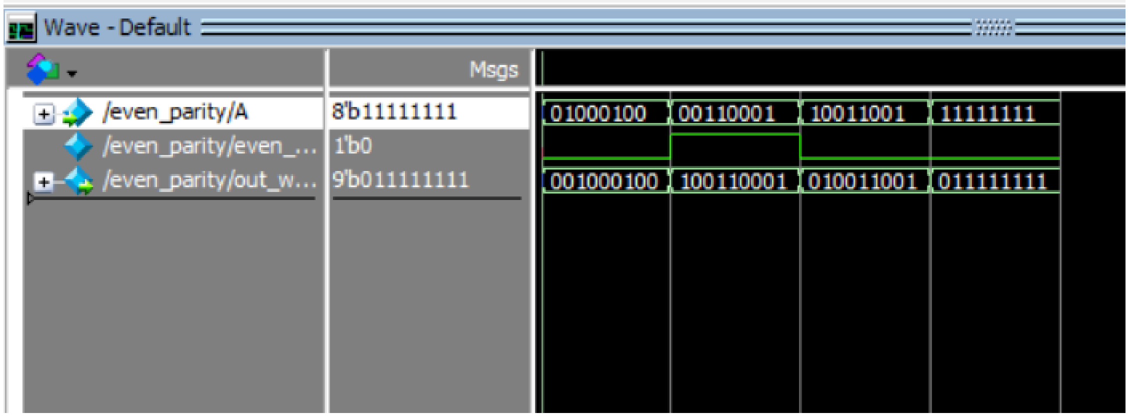


/DECODER/A 2'h3
/DECODER/D 4'h8

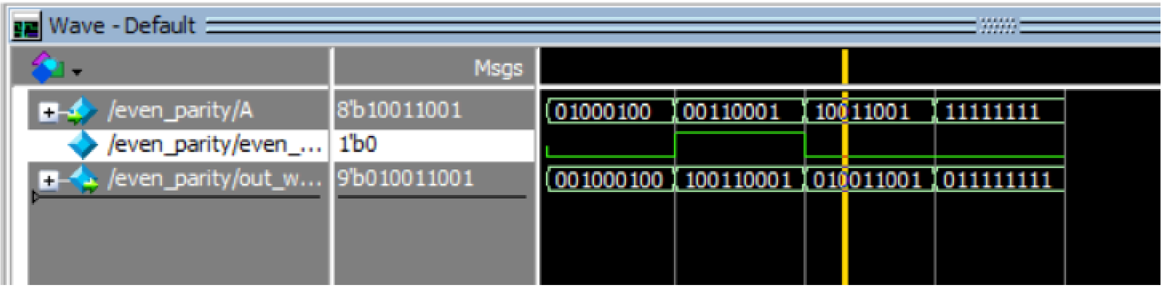
4) Implement an even parity generator module.


C:/questasim64_2021.1/examples/FOURTH_CODE.v (/even_parity) - Default
Ln#

```
module even_parity(A,out_with_parity);
input [7:0]A;
output [8:0]out_with_parity;
wire even_parity;
assign even_parity =^A;
assign out_with_parity={even_parity,A};
endmodule
```

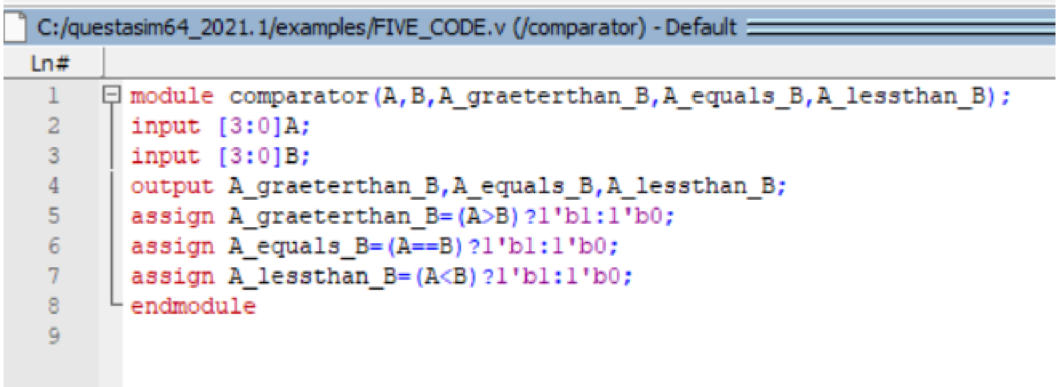


```
/even_parity/A          8'b11111111    01000100   00110001   10011001   11111111
/even_parity/even_...   1'b0
/even_parity/out_w...   9'b011111111   001000100  100110001  010011001  011111111
```

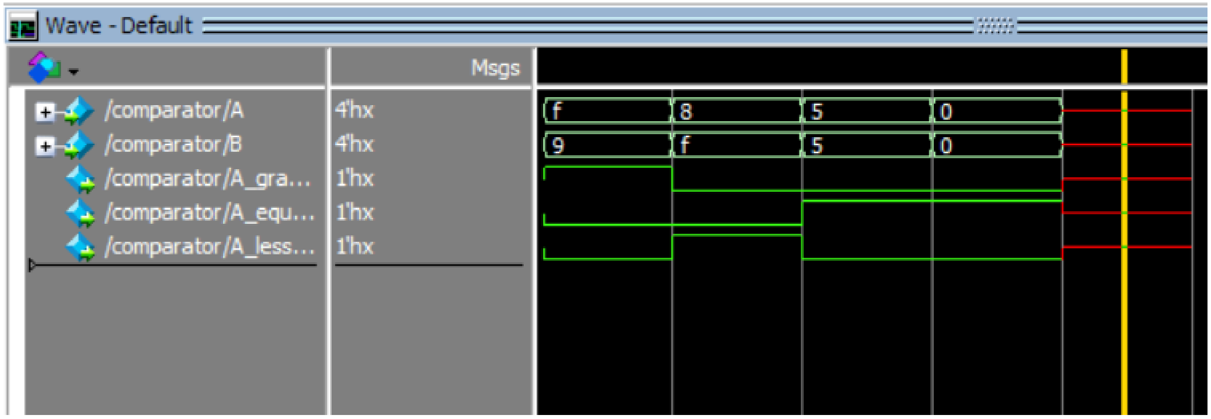


```
/even_parity/A          8'b10011001    01000100   00110001   10011001   11111111
/even_parity/even_...   1'b0
/even_parity/out_w...   9'b010011001   001000100  100110001  010011001  011111111
```

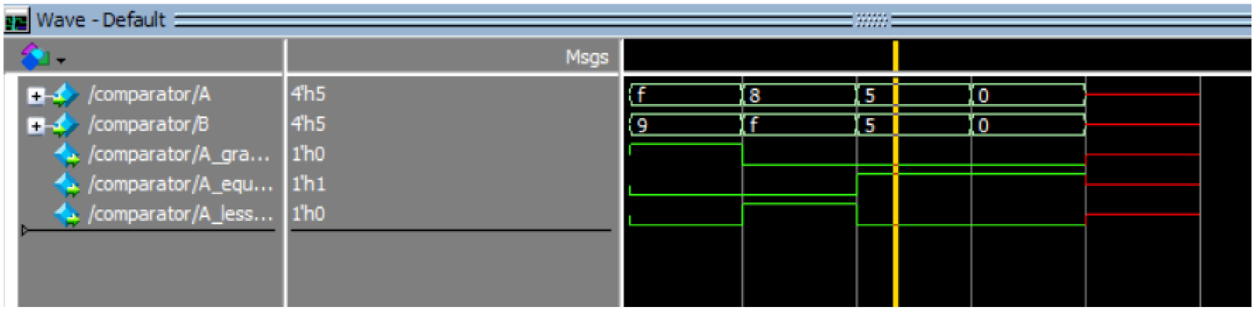
5) Implement a comparator


C:/questasim64_2021.1/examples/FIVE_CODE.v (/comparator) - Default
Ln#

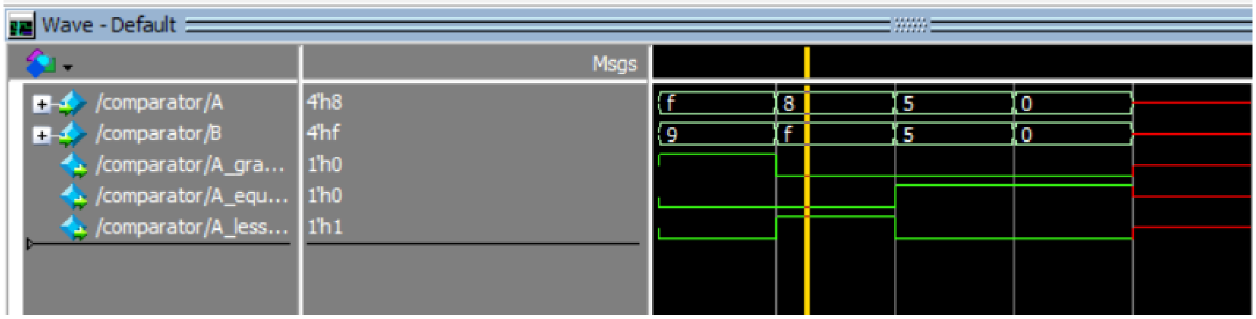
```
module comparator(A,B,A_graeterthan_B,A_equals_B,A_lessthan_B);
input [3:0]A;
input [3:0]B;
output A_graeterthan_B,A_equals_B,A_lessthan_B;
assign A_graeterthan_B=(A>B)?1'b1:1'b0;
assign A_equals_B=(A==B)?1'b1:1'b0;
assign A_lessthan_B=(A<B)?1'b1:1'b0;
endmodule
```



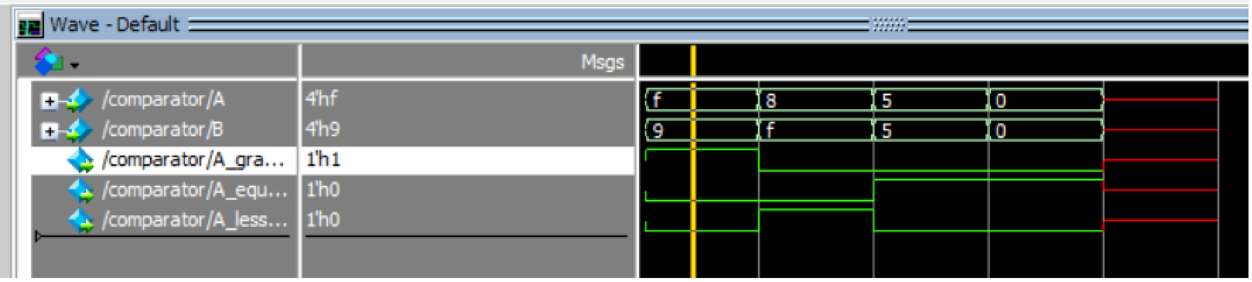
```
Wave - Default
                               Msgs
+ /comparator/A            4'hx
+ /comparator/B            4'hx
  /comparator/A_gra...     1'hx
  /comparator/A_equ...     1'hx
  /comparator/A_less...    1'hx
```



```
Wave - Default
                               Msgs
+ /comparator/A            4'h5
+ /comparator/B            4'h5
  /comparator/A_gra...     1'h0
  /comparator/A_equ...     1'h1
  /comparator/A_less...    1'h0
```



```
Wave - Default
                               Msgs
+ /comparator/A            4'h8
+ /comparator/B            4'hf
  /comparator/A_gra...     1'h0
  /comparator/A_equ...     1'h0
  /comparator/A_less...    1'h1
```



```
Wave - Default
                               Msgs
+ /comparator/A            4'hf
+ /comparator/B            4'h9
  /comparator/A_gra...     1'h1
  /comparator/A_equ...     1'h0
  /comparator/A_less...    1'h0
```