Digital Design Principles

Magnitude Comparator

Name: Rajkaran Singh Grewal
Student No: 8882386
Email: rgrewal2386@conestogac.on.ca

Objective:

- Use VHDL and Verilog to implement a simple magnitude comparator.
- Set up a project in Quartus II targeting your FPGA board.

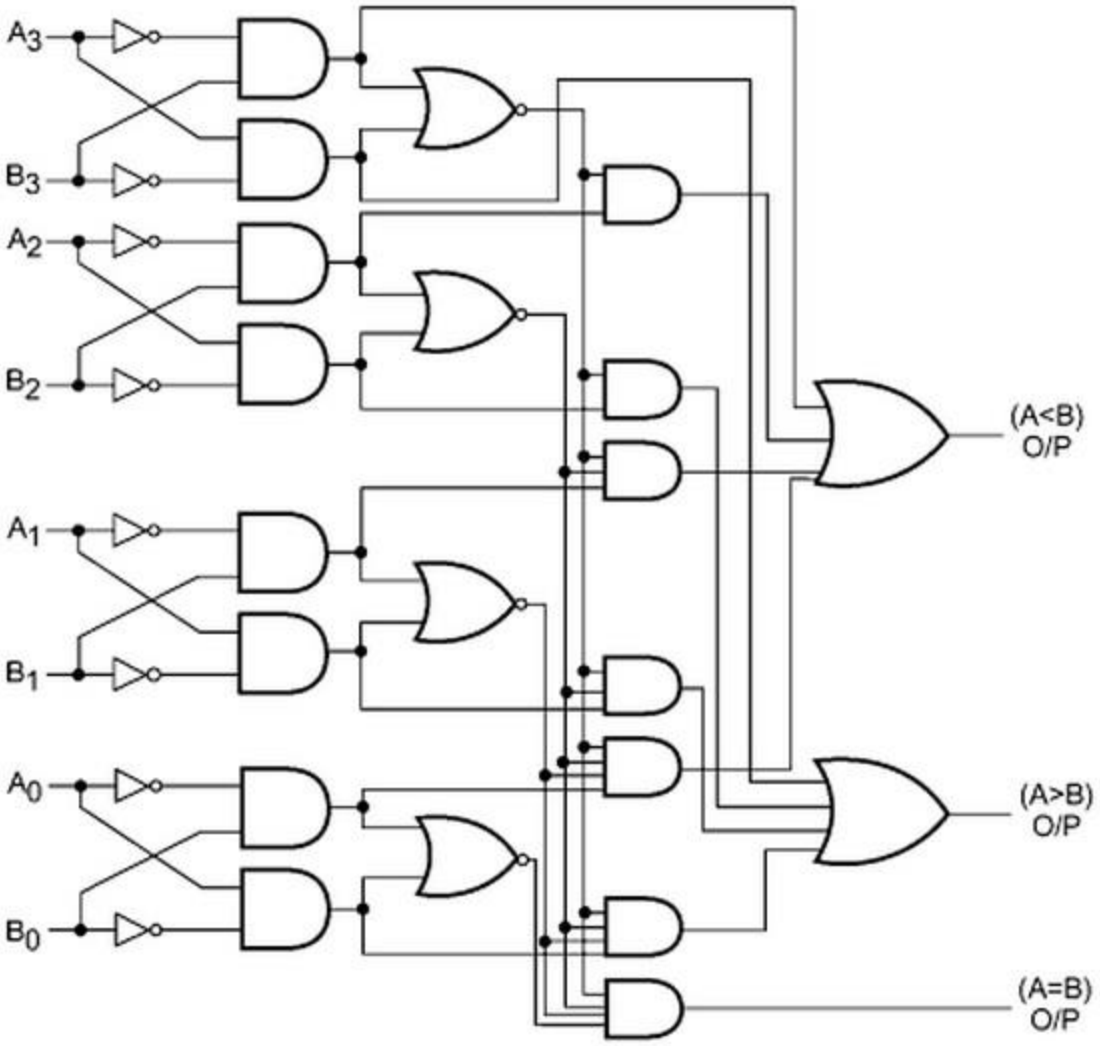
Truth Table:

TRUTH TABLE

INPUTS							OUTPUTS		
COMPARING				CASCADING					
A3, B3	A2, B2	A1, B1	A0, B0	A < B	A = B	A > B	A < B	A = B	A > B
A3 > B3	X	X	X	X	X	X	0	0	1
A3 = B3	A2 > B2	X	X	X	X	X	0	0	1
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	1	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	1	0	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	0	0	1	0	0
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	1	0	0
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	1	0	0
A3 = B3	A2 < B2	X	X	X	X	X	1	0	0
A3 < B3	X	X	X	X	X	X	1	0	0

X = Don't Care Logic 1 = High Level Logic 0 = Low Level

Logic Diagram:



VHDL Introduction:

To make the program in VHDL we will need to initialize the library and entity. Within the entity we will state the 8 input switches and the 3 output LEDs. Within the architecture of the program, we will use 3 different array variables. Where the first array variable will have a total storage of 8 bits and will contain the either A' AND B or A AND B'. The second array variable will store 4 bits where it will contain nor or the ascending order of the previous array. The final array will be 7 bits long and will be AND Gate which merge variable of the second array and some case the first array so that the logic structure will follow the truth table.

VHDL program screenshot:

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.STD_LOGIC_UNSIGNED.ALL;
4   use IEEE.numeric_std.all;
5
6   entity rgrewallab5vhdl is
7       port(
8           A: in std_logic_vector(3 downto 0);
9           B: in std_logic_vector(3 downto 0);
10          LED: out std_logic_vector(2 downto 0)
11      );
12  end rgrewallab5vhdl;
13
14  architecture comparator of rgrewallab5vhdl is
15      signal C: std_logic_vector(7 downto 0);
16      signal D: std_logic_vector(3 downto 0);
17      signal E: std_logic_vector(6 downto 0);
18      begin
19          process(A,B,C,D,E) is
20          begin
21              C(0) <= NOT(A(3)) AND B(3);
22              C(1) <= A(3) AND NOT(B(3));
23              C(2) <= NOT(A(2)) AND B(2);
24              C(3) <= A(2) AND NOT(B(2));
25              C(4) <= NOT(A(1)) AND B(1);
26              C(5) <= A(1) AND NOT(B(1));
27              C(6) <= NOT(A(0)) AND B(0);
28              C(7) <= A(0) AND NOT(B(0));
29
30              D(0) <= NOT(C(0) OR C(1));
```

```
31              D(1) <= NOT(C(2) OR C(3));
32              D(2) <= NOT(C(4) OR C(5));
33              D(3) <= NOT(C(6) OR C(7));
34
35              E(0) <= D(0) AND C(2);
36              E(1) <= D(0) AND C(3);
37              E(2) <= D(0) AND D(1) AND C(4);
38              E(3) <= D(0) AND D(1) AND C(5);
39              E(4) <= D(0) AND D(1) AND D(2) AND C(6);
40              E(5) <= D(0) AND D(1) AND D(2) AND C(7);
41              E(6) <= D(0) AND D(1) AND D(2) AND D(3);
42
43              LED(0) <= C(0) OR E(0) OR E(2) OR E(4);
44              LED(1) <= E(6);
45              LED(2) <= C(1) OR E(1) OR E(3) OR E(5);
46          end process;
47  end comparator;
```

VHDL compilation report:

Compilation Report - rgrewallab5vhdl

Flow Summary

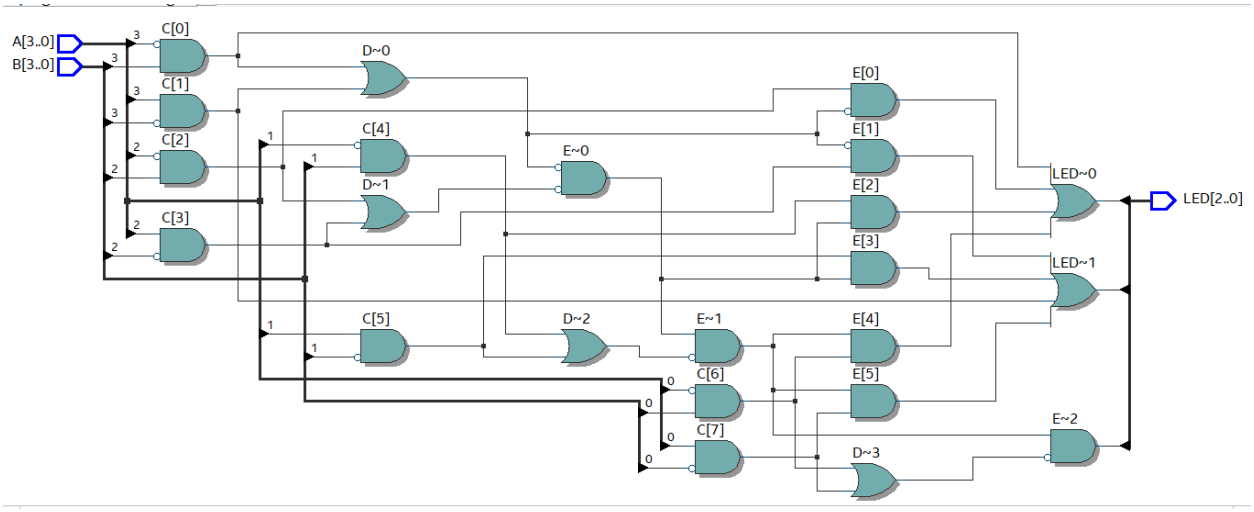
<<Filter>>

Flow Status	Successful - Thu Feb 16 14:17:35 2023
Quartus Prime Version	22.1std.0 Build 915 10/25/2022 SC Lite Edition
Revision Name	rgrewallab5vhdl
Top-level Entity Name	rgrewallab5vhdl
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	5 / 32,070 (< 1 %)
Total registers	0
Total pins	11 / 457 (2 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Pin Planner screenshot:

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
A[3]	Input	PIN_AF10	3A	B3A_N0	PIN_AB27	2.5 V (default)		12mA (default)	
A[2]	Input	PIN_AF9	3A	B3A_N0	PIN_AA26	2.5 V (default)		12mA (default)	
A[1]	Input	PIN_AC12	3A	B3A_N0	PIN_AB28	2.5 V (default)		12mA (default)	
A[0]	Input	PIN_AB12	3A	B3A_N0	PIN_V25	2.5 V (default)		12mA (default)	
B[3]	Input	PIN_AC9	3A	B3A_N0	PIN_AD29	2.5 V (default)		12mA (default)	
B[2]	Input	PIN_AE11	3A	B3A_N0	PIN_AA30	2.5 V (default)		12mA (default)	
B[1]	Input	PIN_AD12	3A	B3A_N0	PIN_AB30	2.5 V (default)		12mA (default)	
B[0]	Input	PIN_AD11	3A	B3A_N0	PIN_AA28	2.5 V (default)		12mA (default)	
LED[2]	Output	PIN_V17	4A	B4A_N0	PIN_AC28	2.5 V (default)		12mA (default)	1 (default)
LED[1]	Output	PIN_W16	4A	B4A_N0	PIN_AC29	2.5 V (default)		12mA (default)	1 (default)
LED[0]	Output	PIN_V16	4A	B4A_N0	PIN_W25	2.5 V (default)		12mA (default)	1 (default)

VHDL RTL View:



Verilog introduction:

For the program in Verilog we will need to initialize the module with the parameters having the three inputs and eight outputs. Within the module block we will use 3 different array variables. Where the first array variable will have a total storage of 8 bits and will contain the either A' AND B or A AND B'. The second array variable will store 4 bits where it will contain nor or the ascending order of the previous array. The final array will be 7 bits long and will be AND Gate which merge variable of the second array and some case the first array so that the logic structure will follow the truth table.

Verilog program screenshot:

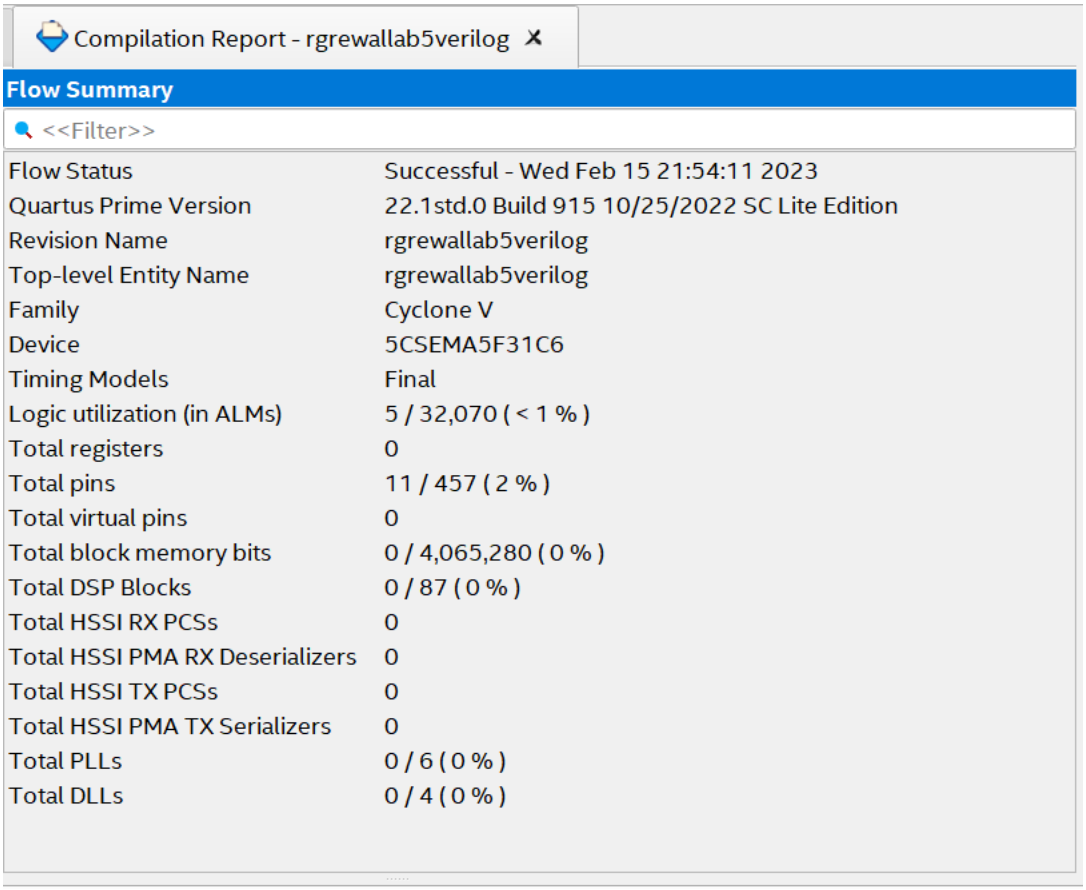
```
module rgrewallab5verilog(input [3:0]A,input [3:0]B,output reg [2:0]LED);
    reg [7:0] C;
    reg [3:0] D;
    reg [6:0] E;
    always @(A,B,C,D,E) begin
        C[0] = !A[3] && B[3];
        C[1] = A[3] && !B[3];
        C[2] = !A[2] && B[2];
        C[3] = A[2] && !B[2];
        C[4] = !A[1] && B[1];
        C[5] = A[1] && !B[1];
        C[6] = !A[0] && B[0];
        C[7] = A[0] && !B[0];

        D[0] = !(C[0] || C[1]);
        D[1] = !(C[2] || C[3]);
        D[2] = !(C[4] || C[5]);
        D[3] = !(C[6] || C[7]);

        E[0] = D[0] && C[2];
        E[1] = D[0] && C[3];
        E[2] = D[0] && D[1] && C[4];
        E[3] = D[0] && D[1] && C[5];
        E[4] = D[0] && D[1] && D[2] && C[6];
        E[5] = D[0] && D[1] && D[2] && C[7];
        E[6] = D[0] && D[1] && D[2] && D[3];

        LED[0] = C[0] || E[0] || E[2] || E[4];
        LED[1] = E[6];
        LED[2] = C[1] || E[1] || E[3] || E[5];
    end
endmodule
```

Verilog compilation report:



Compilation Report - rgrewallab5verilog

Flow Summary

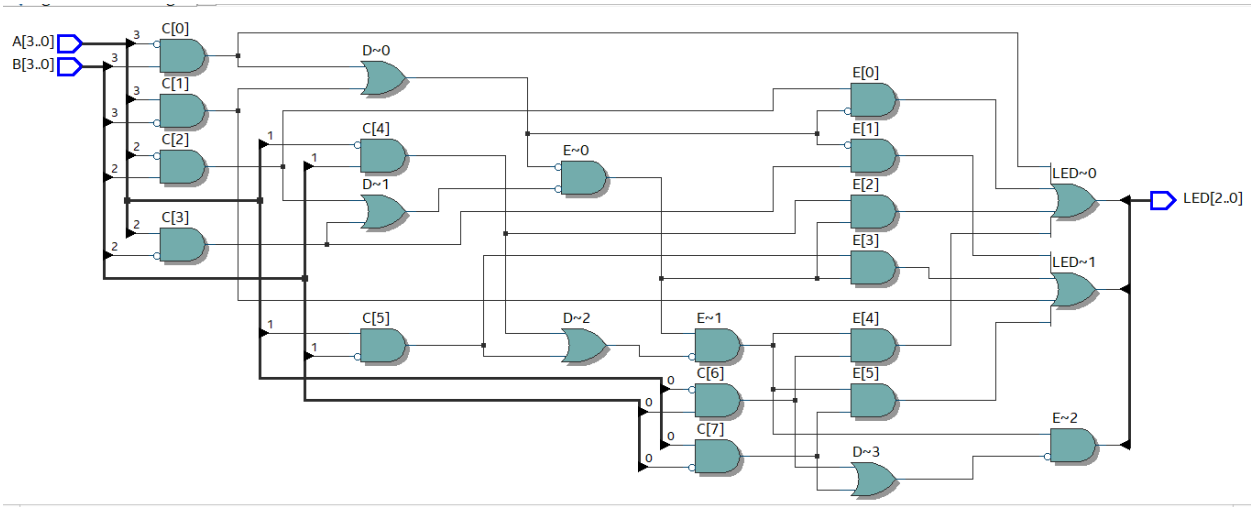
<<Filter>>

Flow Status	Successful - Wed Feb 15 21:54:11 2023
Quartus Prime Version	22.1std.0 Build 915 10/25/2022 SC Lite Edition
Revision Name	rgrewallab5verilog
Top-level Entity Name	rgrewallab5verilog
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	5 / 32,070 (< 1 %)
Total registers	0
Total pins	11 / 457 (2 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Verilog pin planner screenshot:

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
A[3]	Input	PIN_AF10	3A	B3A_N0	PIN_AB27	2.5 V (default)		12mA (default)	
A[2]	Input	PIN_AF9	3A	B3A_N0	PIN_AA26	2.5 V (default)		12mA (default)	
A[1]	Input	PIN_AC12	3A	B3A_N0	PIN_AB28	2.5 V (default)		12mA (default)	
A[0]	Input	PIN_AB12	3A	B3A_N0	PIN_V25	2.5 V (default)		12mA (default)	
B[3]	Input	PIN_AC9	3A	B3A_N0	PIN_AD29	2.5 V (default)		12mA (default)	
B[2]	Input	PIN_AE11	3A	B3A_N0	PIN_AA30	2.5 V (default)		12mA (default)	
B[1]	Input	PIN_AD12	3A	B3A_N0	PIN_AB30	2.5 V (default)		12mA (default)	
B[0]	Input	PIN_AD11	3A	B3A_N0	PIN_AA28	2.5 V (default)		12mA (default)	
LED[2]	Output	PIN_V17	4A	B4A_N0	PIN_AC28	2.5 V (default)		12mA (default)	1 (default)
LED[1]	Output	PIN_W16	4A	B4A_N0	PIN_AC29	2.5 V (default)		12mA (default)	1 (default)
LED[0]	Output	PIN_V16	4A	B4A_N0	PIN_W25	2.5 V (default)		12mA (default)	1 (default)

Verilog RTL View:



Conclusion:

Thus we can conclude that by both the programming languages we by following the above truth table we can create a 4-bit magnitude comparator.

Reference:

https://www.101computing.net/binary-comparators-using-logic-gates/

https://forums.ni.com/t5/Multisim-and-Ultiboard/4-bit-magnitude-comparator/td-p/3887290