EE142---Project_2

TRAFFIC LIGHT CONTROLLER WITHOUT CAUSİNG ANY TRAFFİC JAM

Sefa Kayraklık
2015401066 |

Introduction:

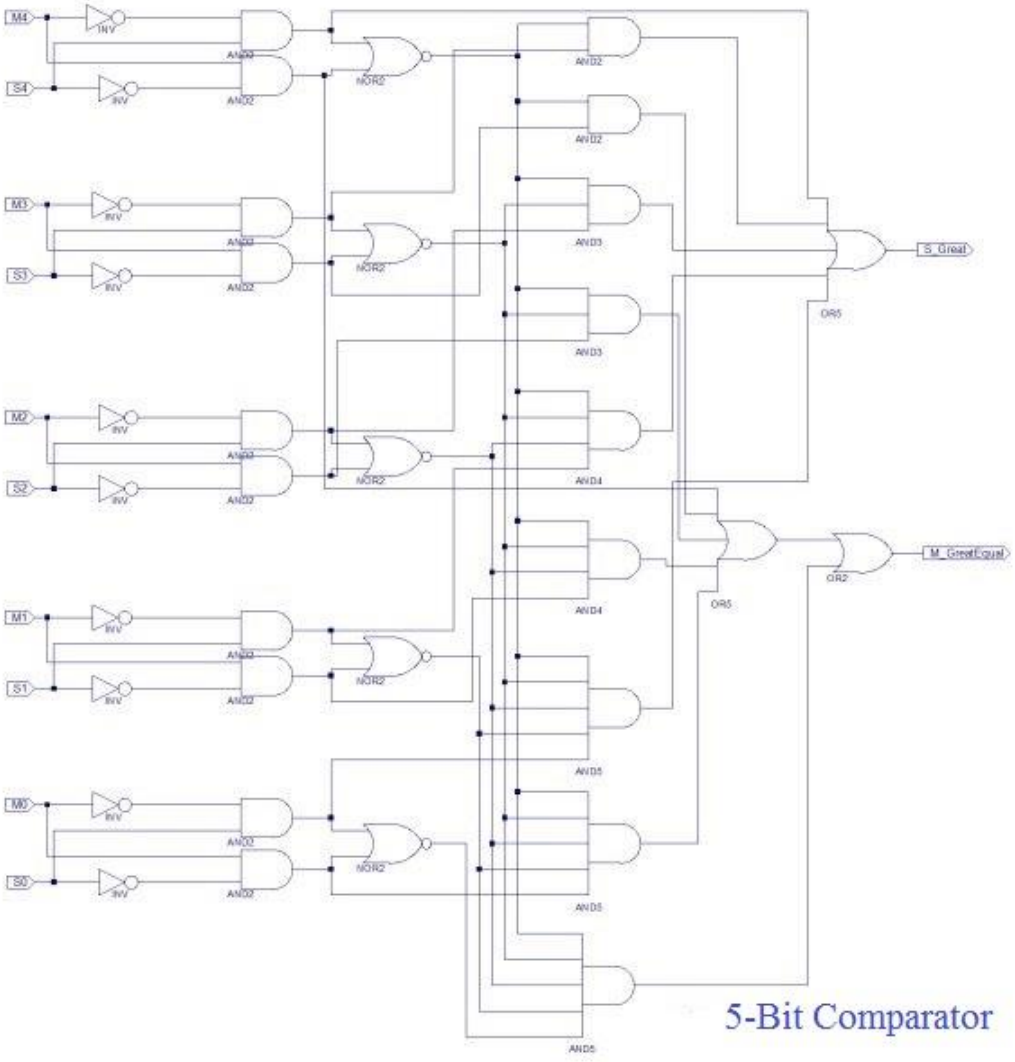
In this project, it is asked that designing a traffic light controller for a crossroad such that it prevents the traffic jam in the main road and in the side road. The procedure for designing such a synchronous sequential circuit can be considered as following:

1. ***Stating the problem:*** In this project, it is asked that to decrease the traffic jam in the main road preferentially and then decrease the traffic jam in the side and main road together. To build such a controller, some elements are needed such as: a synchronous sequential circuit that determines in which road the car passes, and the traffic lights are green if 1 is given or they are red if 0 is given. Naturally, there are some hierarchical rules to be followed such as: firstly, at any instant only one of the traffic lights can be red or green; secondly, the number of cars in the main road has a threshold '10' namely, if that number is above this threshold, the light in the main road should be green, even if the number of cars in the side road is greater than that number; thirdly, in the roads, there can be at most 31 cars; lastly, if the number of cars in the main road is below the threshold, the light in the road with more cars should be green and if the numbers are equal, the light in the main road should be green. Only one car leaves the crossroad in one second if the light is green. And ones one of the light is green, all the cars in that road should leave the road, each car should leave in 1 second, and after that, the other light should be green, after all cars in both road leave, the light in the main road should be green.
2. ***Determining the number of available inputs and required outputs:*** In the project, there are ten available inputs that express the number of cars in each road and 2 other inputs for counters' clock pulse and for set the count 00000 state. And there are also two required outputs that indicate whether the traffic light is green or red in each road.
3. ***Determining the type and the number of elements:*** There should be up counters that count the number of cars which leave the road; subtractors that subtract the number which will be taken from the counters from the number which are given, and one comparator that compares the number of cars in the roads to each other and one comparator that compares the number of cars in the main road with threshold 10.
   a. ***The Comparators:***
      i. *The comparator which compares the number of cars in each road:* The comparator should be able to compare 5 bit binary numbers, namely it should have 10 inputs. It determines relative magnitudes of the number of cars. The outcome of the comparators is defined by two binary variables that indicate whether M≥S or M<S. The design procedure of the comparator [1] was followed as in Digital Design from M. Mano and M. Ciletti.
      $M = M_4M_3M_2M_1M_0$ and $S = S_4S_3S_2S_1S_0$ each letter represents one of the digits in the number.
      $x_i = M_iS_i + M'_iS'_i$ for i = 0, 1, 2, 3, 4
         1. $(M \geq S) = M_4S'_4 + x_4M_3S'_3 + x_4x_3M_2S'_2 + x_4x_3x_2M_1S'_1 + x_4x_3x_2x_1M_0S'_0 + x_4x_3x_2x_1x_0$
         2. $(M < S) = M'_4S_4 + x_4M'_3S_3 + x_4x_3M'_2S_2 + x_4x_3x_2M'_1S_1 + x_4x_3x_2x_1M'_0S_0$
      ii. *The comparator which compares the number of cars in the main road with the threshold 10:* This comparator is the comparator designed above with the input S = 01010.
   b. **The Counters:** The counters should be able to count up to 31 from 0, and they should count up the number of cars whenever a car leaves the road, thus they would count the number of cars which pass from that road. To perform such a counter, the following strategy [2] was considered: Actually, there is no need to go through a sequential logic process. In a

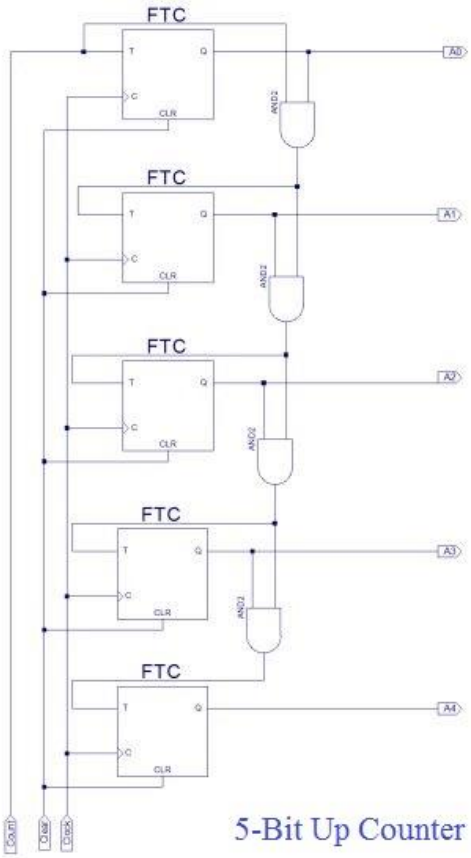
synchronous binary counter, the flip flop in the least significant position is complemented with every pulse. A flip flop in any other position is complemented when all the bits in the lower significant positions are equal to 1. For example, if the present state of a five-bit counter is $A_4A_3A_2A_1A_0 = 00111$, the next count is 01000. A_0 is always complemented. A_1 is complemented because the present state of $A_0 = 1$. A_2 is complemented because the present state of $A_1A_0 = 11$. A_3 is complemented because the present state of $A_2A_1A_0 = 111$. However, A_4 is not complemented because the present state of $A_3A_2A_1A_0 = 0111$, which does not give an all-1's condition. Namely, if the outputs of the previous flip flops are successively connected to each other with and gates, this sequential circuit becomes a counter. In this counter T-flip flops with positive edge clock are used. Whenever T is 1, the counter counts up at the positive edge of the clock; whenever T is 0, the counter keeps the present values, namely T is used as enable in the counter. All the clocks of the flip flops are connected to a common clock. Each flip flop possesses reset to begin with the state 00000. And to design 5-bit counter, 5 flip flops are required.

c. **Subtractor:** The subtractor is used for the subtraction of the number of cars which the counters count from the number of the cars which are given by the user. To design a subtractor, a full-adder is needed. If the full adder's carry in is given as 1 and the subtrahend number is complemented, the full adder will work as subtractor because doing so, the full adder subtracts with the 1's complement. To obtain a 5-bit subtractor, 5 full adders should be used. And the first full adder's carry in is given 1, the others' carry in are connected to the previous full adder's carry out. [3]

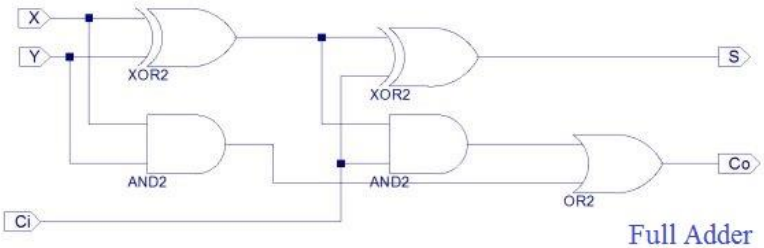
Schematics of The Elements:



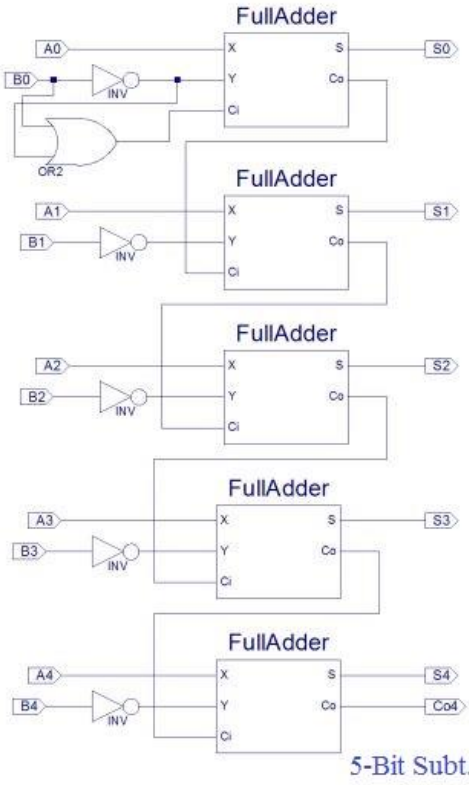
5-Bit Comparator



5-Bit Up Counter

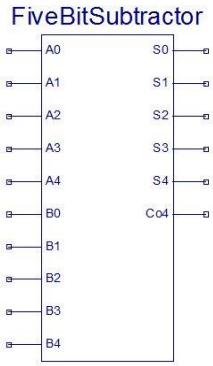


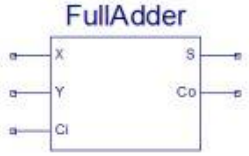
Full Adder

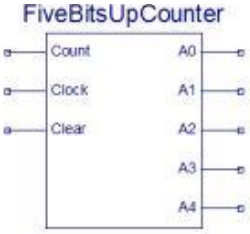


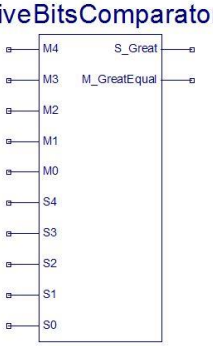
5-Bit Subt.

Block Designs of The Elements:





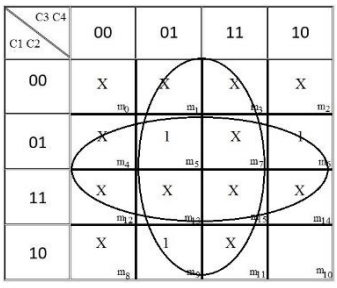


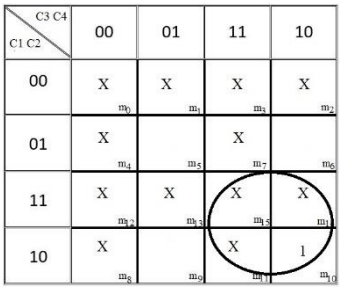


4. ***Strategies used in the synchronous sequential circuit:*** In the project, there are some stages:
   a. *At the beginning,* the comparators determine which light is green which one is red. And to convert this logic to a gate circuit, the truth table of it should be considered, the both the functions of main road's light and side road's light should be specified.
   C1=The Threshold Comparator's SGreat, C2=The Comparator's MGreatEqual
   C3=The Second Comparator's SGreat, C4=The Second Comparator's MGreatEqual,

The Truth Table of The Comparators

C1	C2	C3	C4	M	S
0	0	0	0	X	X
0	0	0	1	X	X
0	0	1	0	X	X
0	0	1	1	X	X
0	1	0	0	X	X
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	X	X
1	0	0	0	X	X
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X





M=C2+C4 S=C1.C3

b. After the comparison and until the last car, there is no need the comparators and the elements that determine the condition of the light are counters and subtractors. The lights' logic values should be counted, so each counter's count input is directly connected to the values of the lights. And the outcomes of the subtractors are the number of cars that wait to leave the road. In the project, the asked condition is the cars' leaving the road whose light is green at the beginning, then whenever all cars leave the road, it should be red and the other light should be green. The determining part is the each subtractors' 0s. If all the bits are 0, then it means that all cars leave the road.

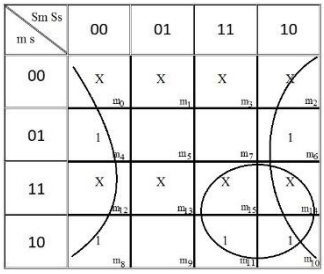
$Sm=Sm_0+Sm_1+Sm_2+Sm_3+Sm_4$ (main road's subt.) $Ss=Ss_0+Ss_1+Ss_2+Ss_3+Ss_4$ (side road's subt.)

c. *When all the cars leave the* ro*ad,* the asked is that the main road's light is green. To obtain this situation, it is needed that the output of the main road's light and the Sm are connected to each other with an and gate because when both 0, it means don't count up.

d. *Combining these observation,* the determining elements of the roads' light are m, and s (from step a); Sm, and Ss (from step b). The truth tables and the karnaugh maps of the roads' lights should be drawn.

The Truth Table of The Comparators

m	s	Sm	Ss	M	S
0	0	0	0	X	X
0	0	0	1	X	X
0	0	1	0	X	X
0	0	1	1	X	X
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X



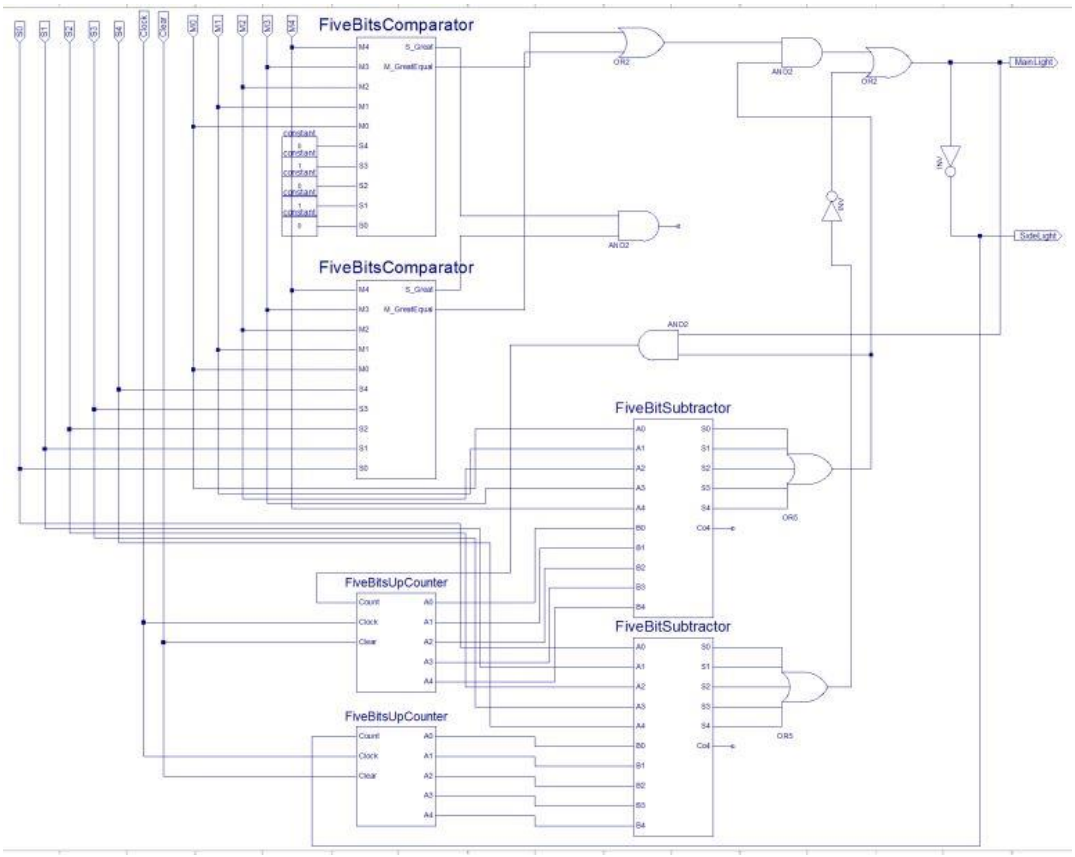
M=Ss'+m.Sm

S =(Ss'+m.Sm)' (The complement of The Main Road)

5. ***Drawing the logic circuit:*** In the project, Xilinx ISE Design Suite 14.7 is used to draw the circuit. And during the drawing, the least number of gates and block designs should be used due to effectiveness. The logic circuit of the project is below.

6. ***Implementing the logic circuit:*** Users should enter 2 5-bit binary numbers to each road, then they should clear the counters by setting the clear input 1 and just after that setting it as 0. Finally, in the project the positive edge clock is used, this should be considered, and the clock should be given as a process to obtain an accurate result. To check the result whether it is true or false, the simulation of the logic circuit should be done via Xilinx.

The Schematic of The Project



Conclusion: Through the project, analyzing the problem according to the logical thinking, and binary algebra, designing a synchronous sequential circuit, and converting the daily issues to the digital design system are expected to be learned. In the project, counters, comparators, and subtractors are studied and it is learned how to complement, how to design these symbols and block designs and how they work in circuits.

References:

[1] M. Mano and M. Ciletti, "Magnitude Comparator", in Digital Design, 5th ed., International ed., 2013, ch. 4, sec. 8, pp. 148-150.

[2] M. Mano and M. Ciletti, "Synchronous Counters", in Digital Design, 5th ed., International ed., 2013, ch. 6, sec. 4, pp. 271-272.

[3] M. Mano and M. Ciletti, "Binary Adder-Subtractor", in Digital Design, 5th ed., International ed., 2013, ch. 4, sec. 5, pp. 141-142.