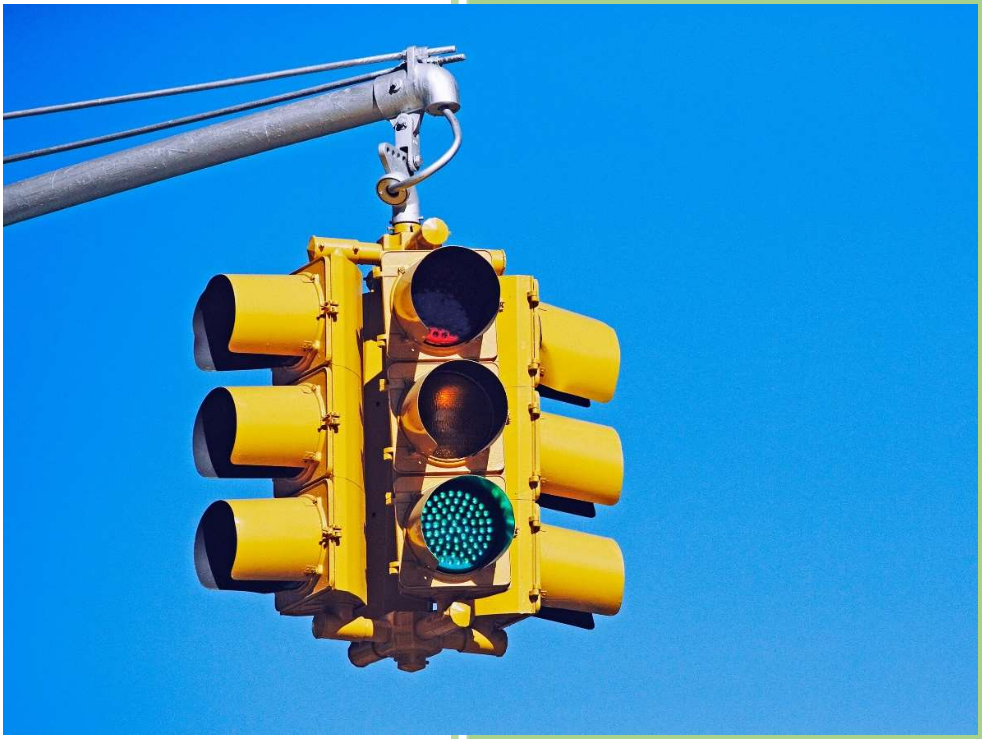
Traffic Light Controller



Submitted by

Kadum Perme

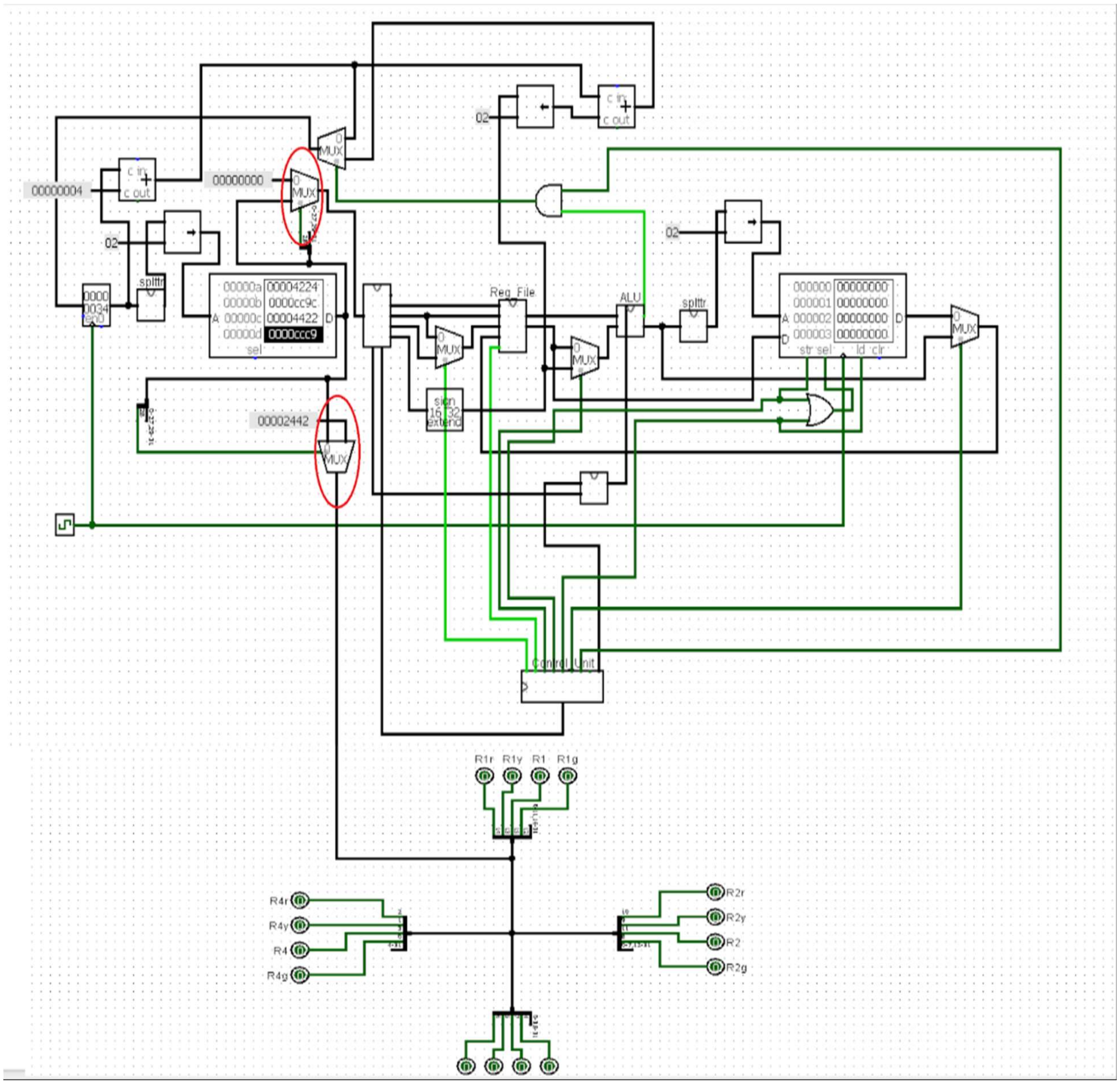
Roll No : 1601CS17

Course : CS322

Aim :

To implement the working of traffic light controller using the fundamental features of MIPS single cycle processor in logisim.

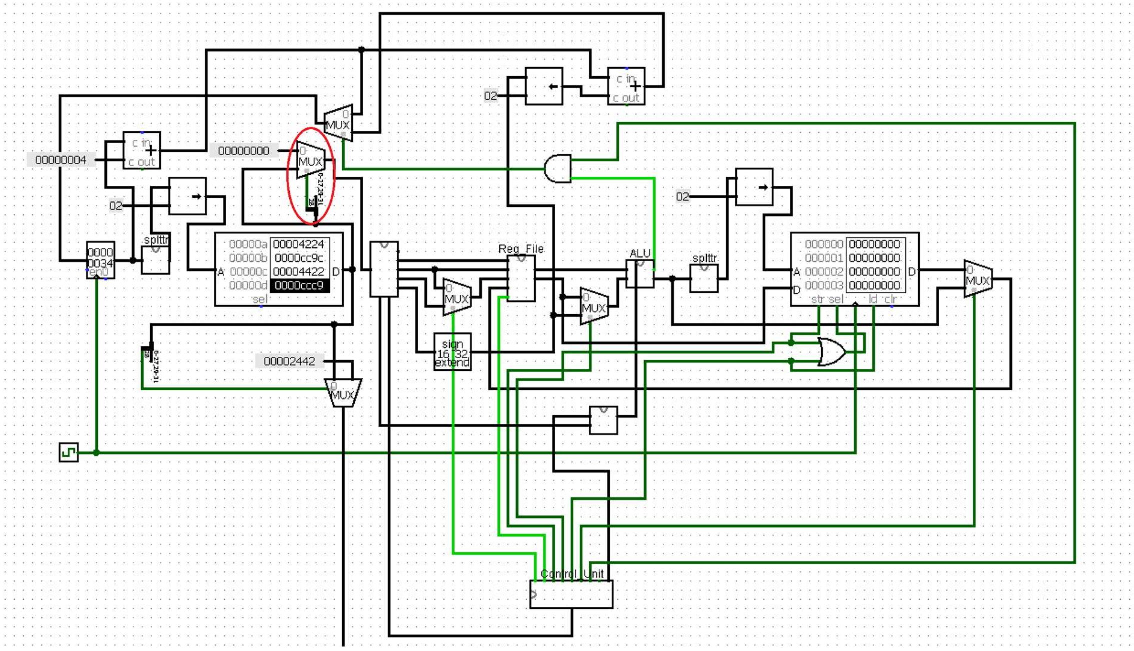
Architecture Details :



The architecture is totally based upon the working of a single cycle processor with slight modifications in order to implement the working of a traffic light controller.

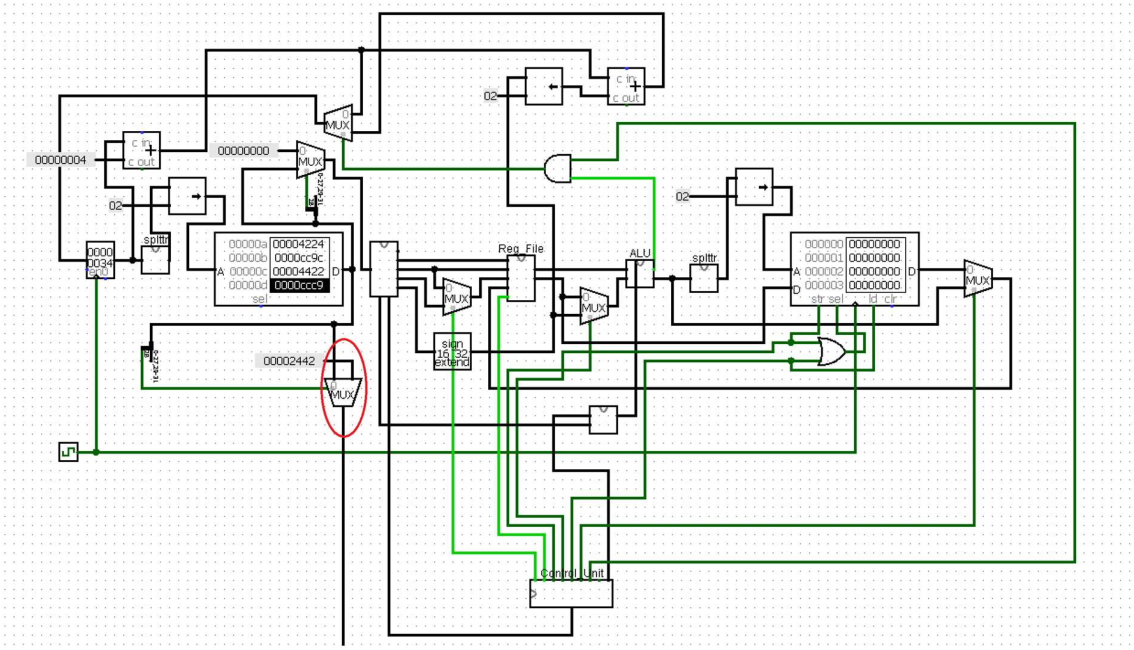
The single-cycle MIPS microarchitecture used for this project uses the same concepts and functionalities taught in lab assignments. All the modifications and extra circuits added are clearly described below.

Modifications :

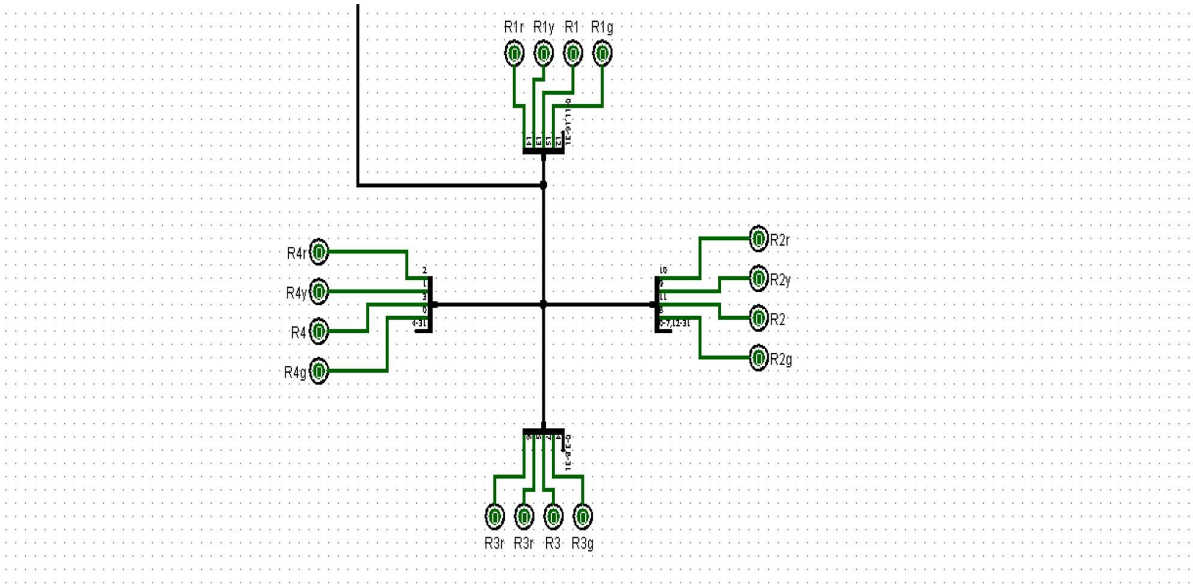


Not all the instructions fetched in the instruction memory is related to any MIPS assembly code, i.e., some of the instructions have nothing to do with the register file, ALU and the data memory. So to filter out those instructions and to ensure that only instructions related to MIPS assembly language are pass forwarded in the datapath, the multiplexer circled in red is used.

All the instructions related to MIPS assembly language has their 28th bit as ‘1’ while those not related to MIPS assembly language has their 28th bit as ‘0’. So, if the 28th bit of an instruction is ‘0’ an instruction for NOP (0x00000000) is pass forwarded to datapath otherwise the MIPS assembly instruction is pass forwarded to the datapath.



The traffic controller which is shown in below diagram is controlled by the instructions which are not MIPS assembly code. So, whenever a MIPS assembly instruction comes, the multiplexer marked in red in above diagram passed a constant instruction for maintaining the sequence of glowing of traffic lights.



The above diagram depicts the traffic controller.

Symbol Meanings:

R_i = Bit assigned for blinking lights on ith road

R_{ir} = Bit assigned for red light for ith road

R_{iy} = Bit assigned for yellow light for ith road

R_{ig} = Bit assigned for green light for ith road

Let us first try to understand how the instructions from the instruction memory cotrols the traffic controller by using a table.

Bit No. →	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Instructions ↓	R_1	R_{1r}	R_{1y}	R_{1g}	R_2	R_{2r}	R_{2y}	R_{2g}	R_3	R_{3r}	R_{3y}	R_{3g}	R_4	R_{4r}	R_{4y}	R_{4g}
0x00007777	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00007777	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00007777	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00009CCC	1	0	0	1	1	1	0	0	1	1	0	0	1	1	0	0
0x00002244	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0
0x0000C9CC	1	1	0	0	1	0	0	1	1	1	0	0	1	1	0	0
0x00004224	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0
0x0000CC9C	1	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0
0x00004422	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0
0x0000CCC9	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1
0x00002442	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0
0x1000FFF1 = beq \$0, \$0, -9																

Now, let me explain the whole process briefly and also rule out some processes that couldn't be shown via the table.

Brief Explaination :

The mips single cycle processor takes 32-bit instructions. I have used the first 16 bits from the LSB (Least Significant bit) side for turning the LED's on and off. There are four roads and each road requires 4 bits, i.e., one bit for red light, one for red, one for green and the fourth for blinking lights on the road also called 'road stud' or 'road turtles'.

Bits in the range 0-3 are used for providing signals to road 4

0th bit is used for R_{1g}

1st bit is used for R_{1y}

2nd bit is used for R_{1r}

3rd bit is used for R_1

Bits in the range 4-7 are used for providing signals to road 3

4th bit is used for R_{2g}

5th bit is used for R_{2y}

6th bit is used for R_{2r}

7th bit is used for R_2

Bits in the range 8-11 are used for providing signals to road 2

8th bit is used for R_{3g}

9th bit is used for R_{3y}

10th bit is used for $R3_{r}$

11th bit is used for R_3

Bits in the range 12-15 are used for providing signals to road 1

12th bit is used for R_{4g}

13th bit is used for R_{4y}

14th bit is used for R_{4r}

15th bit is used for R_4

Bits in the range 16-31 are not used for any instruction except the last instruction which is the beq(branch equal instruction) and hence, all the instructions except the last instruction have value as ‘0’ in their first 16 bits from MSB(Most Significant Bit) ,i.e, bits in the range 31-16 have value ‘0’ for all instructions except the last instruction.

Now consider the following instructions that I have fed to my instruction memory:

Instruction 1 : 0x00007777

Instruction 2 : 0x00000000

Instruction 3 : 0x00007777

Instruction 4 : 0x00000000

Instruction 5 : 0x00007777

Instruction 6 : 0x00000000

The first six instructions are used for denoting that the system is starting for the first time.These instructions allow all the LEDs to turn ON and OFF at the same time ,i.e., let the LEDs to blink uniforlmly.

Instruction 7 : 0x00009CCC

0x9CCC = 1001 1100 1100 1100

R_{1g}, R_{2r}, R_{3r}, R_{4r} are turned ON showing that the green light is ON only for road 1 and rest of the roads are shown a signal of red light, i.e, the signal for stop.

R_1, R_2, R_3, R_4 are turned ON for turning ON the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 8 : 0x00002244

0x2244 = 0010 0010 0100 0100

R_{1y}, R_{2y}, R_{3r}, R_{4r} are turned ON showing that the yellow lights for road 1 and road 2 are ON, indicating that a signal change is going to happen for road 1 and 2. Rest of the roads are shown a signal of red light, i.e., the signal for stop.

R_1, R_2, R_3, R_4 are turned OFF for turning OFF the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 9 : 0x0000C9CC

0xC9CC = 1100100111001100

R_{1r}, R_{2g}, R_{3r}, R_{4r} are turned ON showing that the green light is ON only for road 2 and rest of the roads are shown a signal of red light, i.e, the signal for stop.

R_1, R_2, R_3, R_4 are turned ON for turning ON the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 10 : 0x00004224

0x4224 = 0100 0010 0010 0100

R_{1r}, R_{2y}, R_{3y}, R_{4r} are turned ON showing that the yellow lights for road 2 and road 3 are ON, indicating that a signal change is going to happen for road 2 and 3. Rest of the roads are shown a signal of red light, i.e., the signal for stop.

R_1, R_2, R_3, R_4 are turned OFF for turning OFF the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 11 : 0x0000CC9C

0xCC9C = 1100 1100 1001 1100

R_{1r}, R_{2r}, R_{3g}, R_{4r} are turned ON showing that the green light is ON only for road 3 and rest of the roads are shown a signal of red light, i.e, the signal for stop.

R_1, R_2, R_3, R_4 are turned ON for turning ON the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 12 : 0x00004422

0x4422 = 0100 0100 0010 0010

R_{1r}, R_{2r}, R_{3y}, R_{4y} are turned ON showing that the yellow lights for road 3 and road 4 are ON, indicating that a signal change is going to happen for road 3 and 4. Rest of the roads are shown a signal of red light, i.e., the signal for stop.

R_1, R_2, R_3, R_4 are turned OFF for turning OFF the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 13 : 0x0000CCC9

0xCCC9 = 1100 1100 1100 1001

R_{1r}, R_{2r}, R_{3r}, R_{4g} are turned ON showing that the green light is ON only for road 4 and rest of the roads are shown a signal of red light, i.e, the signal for stop.

R_1, R_2, R_3, R_4 are turned ON for turning ON the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 14 : 0x00002442

0x2442 = 0010 0100 0100 0010

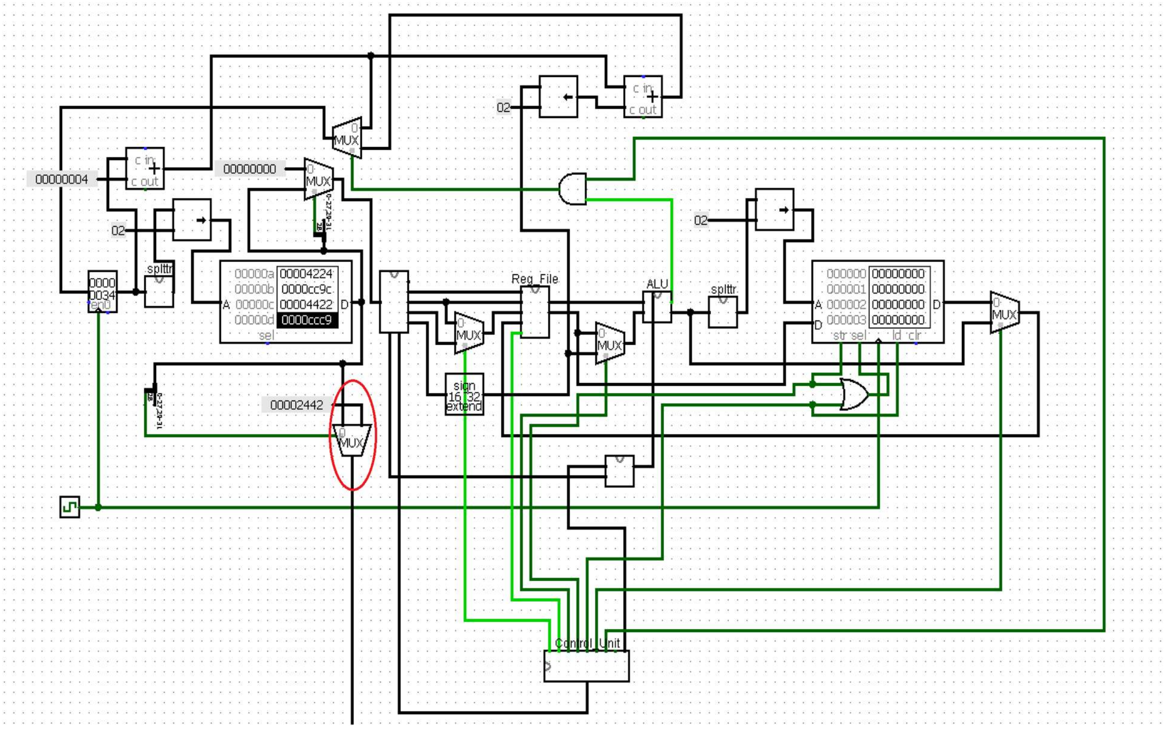
R_{1y}, R_{2r}, R_{3r}, R_{4y} are turned ON showing that the yellow lights for road 1 and road 4 are ON, indicating that a signal change is going to happen for road 1 and 4. Rest of the roads are shown a signal of red light, i.e., the signal for stop.

R$_1$, R$_2$, R$_3$, R$_4$ are turned OFF for turning OFF the road turtles on road 1, road 2, road 3 and road 4 respectively.

Instruction 15 : 0x1000FFF7

0X1000FFF7 (Machine Code) == beq $0, $0, -9 (MIPs Assembly Code)

This instruction is for jumping 9 instructions backward as Instruction 7 is 9 instructions behind from the next address after Instruction 15. After the machine execute this instruction, the execution jumps back to Instruction 7 and the cycle continues.



While executing Instruction 15, since the 28th bit of Instruction 15 has a value 1, the red-circled mux in the above diagram allows the constant 0x00002442 to regulate the traffic lights, which regulate the traffic lights as in Instruction 14. If this is not done the lighting sequence of the traffic lights will be disrupted.

Thank You