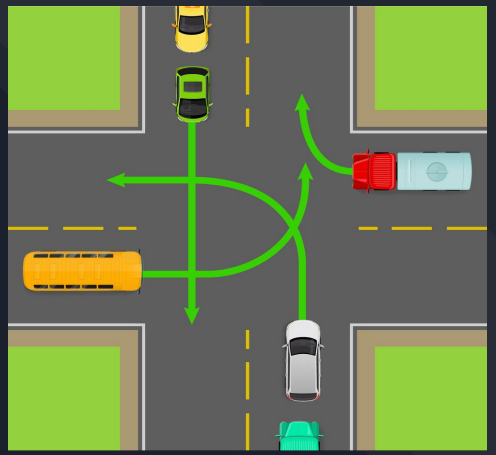
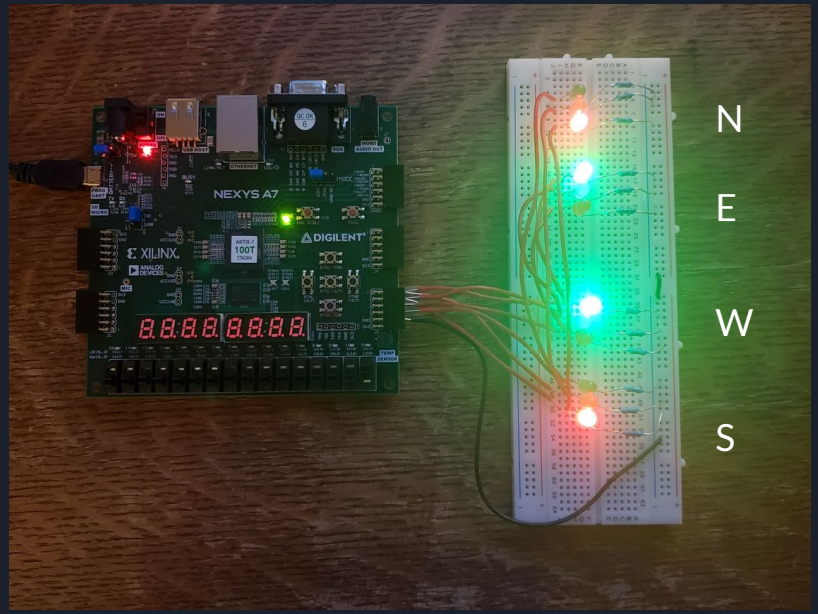
4-Way Traffic Light Controller

By: Beau Tucker, Ben Bonham, Nick Schneider

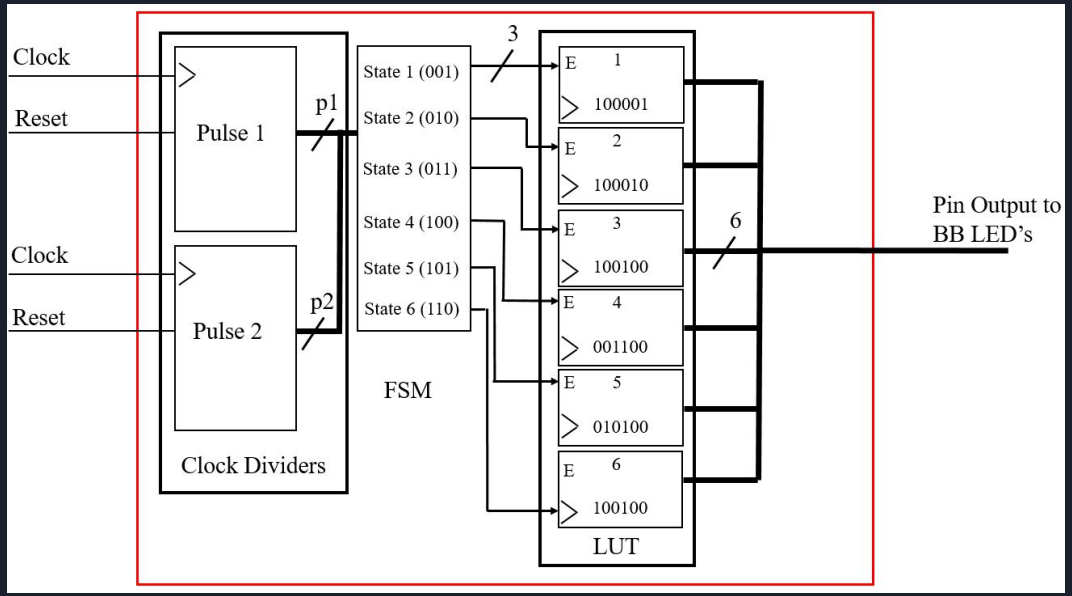




Overview

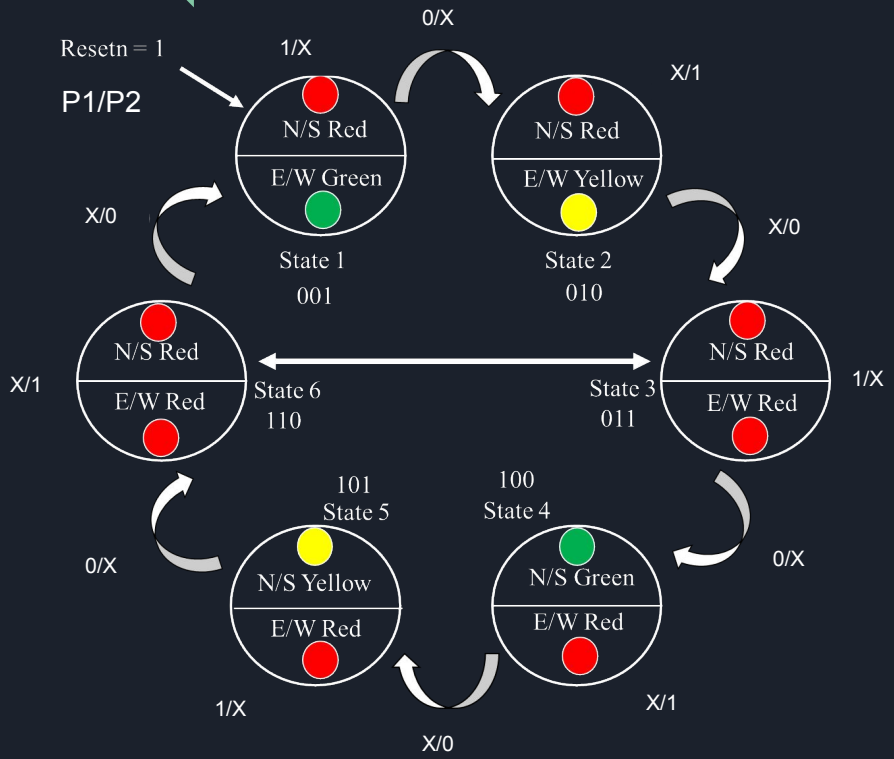
- Design and implementation of a 4-way Traffic Light Controller. To do this we will utilize the software Vivado and the Nexys Artix-7 100T FPGA Board with an external circuit to simulate the results.
- In the four way intersection, North/South and East/West lights will share the same light patterns, creating two groups of traffic lights.
- While one pair of lights is red, the other pair will cycle through green and yellow lights until it reaches red. Once the other pair is red then the next pair of lights will begin to cycle through.

Traffic Control Light - Block Diagram



- Circuit structure is composed of three unique components
  - Pulse 1/2 - two pulses work in unison to provide two 1 bit outputs
    - Output -> P1, P2
  - FSM - A single Mealy type device, provides a 3 bit code depending on state that triggers the LUT
    - Output -> “000” (S1 - S6)
  - LUT - stores a 6 bit digit code used to signal logic high or low for the attached LED’s
    - Output -> “000000”

FSM State Design



Timer settings can be altered so that additional hold times can be added. Each pulse timer is reset after a state change. Intersection timing could easily be changed quickly.

- S1 -> initiated when resetn = 1
  - Input -> P1 = 1, P2 = 0
  - Output -> Y = 001, sig = 100001
- S2 -> initiated when resetn = 1
  - Input -> P1 = 0, P2 = 1
  - Output -> Y = 010, sig = 100010
- S3 -> initiated when resetn = 1
  - Input -> P1 = 1, P2 = 0
  - Output -> Y = 011, sig = 100100
- S4 -> initiated when resetn = 1
  - Input -> P1 = 0, P2 = 1
  - Output -> Y = 100, sig = 001100
- S5 -> initiated when resetn = 1
  - Input -> P1 = 1, P2 = 0
  - Output -> Y = 101, sig = 010100
- S6 -> initiated when resetn = 1
  - Input -> P1 = 0, P2 = 1
  - Output -> Y = 110, sig = 100100

Excitation Table

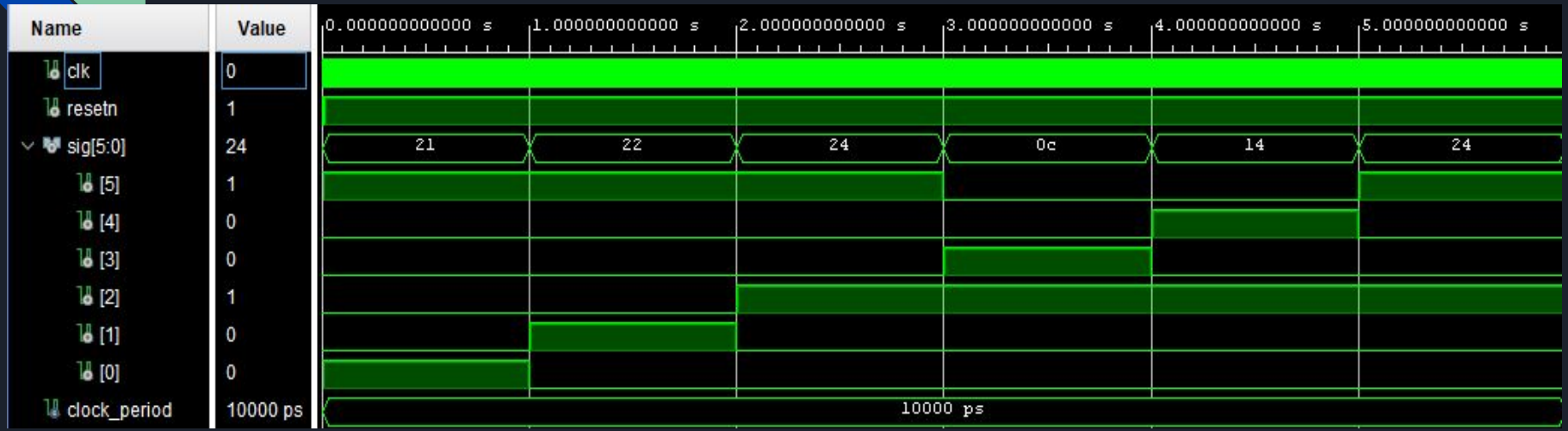
State Assignment

S1	001
S2	010
S3	011
S4	100
S5	101

P1	P2	Present State	Next State	Output	LUT Output
0	x	S1	S2	010	100010
x	0	S2	S3	011	100100
0	x	S3	S4	100	001100
x	0	S4	S5	101	010100
0	x	S5	S6	011	100100
x	0	S6	S1	001	100001
1	x	S1	S1	001	100001
x	1	S2	S2	010	100010
1	x	S3	S3	011	100100
x	1	S4	S4	100	001100
1	x	S5	S5	101	010100
x	1	S6	S6	011	100100

- Inputs P1 and P2 control the current states
- State will be the output of the FSM
- FSM state picks appropriate LUT according

Simulation Results



- Sig [5] = N/S Red
- Sig [4] = N/S Yellow
- Sig [3] = N/S Green
- Sig [2] =E /W Red
- Sig [1] = E/W Yellow
- Sig [0] = E/W Green

- Each have one second intervals

References

1. Llamocca “4 to 1 LUT” VHDL coding for FPGAs http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html
2. “Clock Divider In VHDL.” Stack Overflow. https://stackoverflow.com/questions/61878127/clock-divider-in-vhdl-from-100mhz-to-1hz-code
3. “Design of a VHDL LUT Module” Stack Overflow https://stackoverflow.com/questions/21976749/design-of-a-vhdl-lut-module
4. Nexys A-7 FPGA Trainer Board-Reference Manuel https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual?redirect=1