ECE 2700 – Final Project

By: Ben Fakhoury
Kim Pawlowicz
Yousif Elia
Alex Zufelt

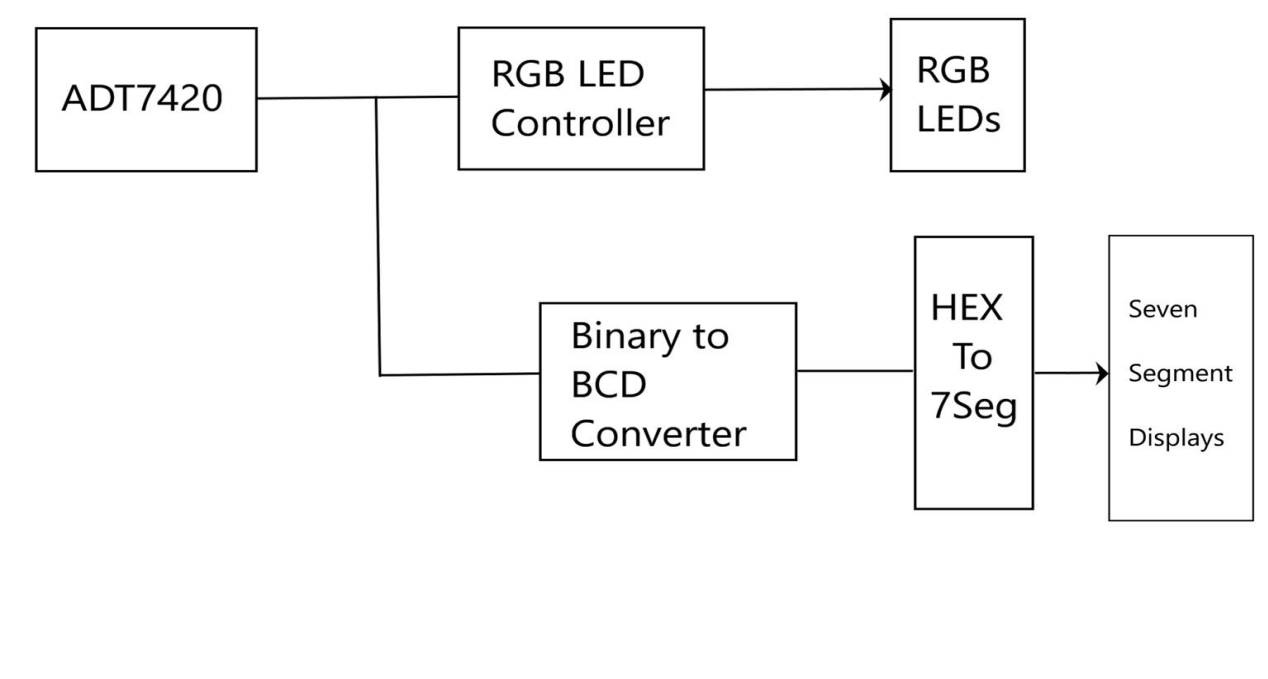
Our Project

Our project makes use of the RGB LEDs, ADT7420 Temperature Sensor, and 7 Segment Display on the Nexys 4 DDR board.

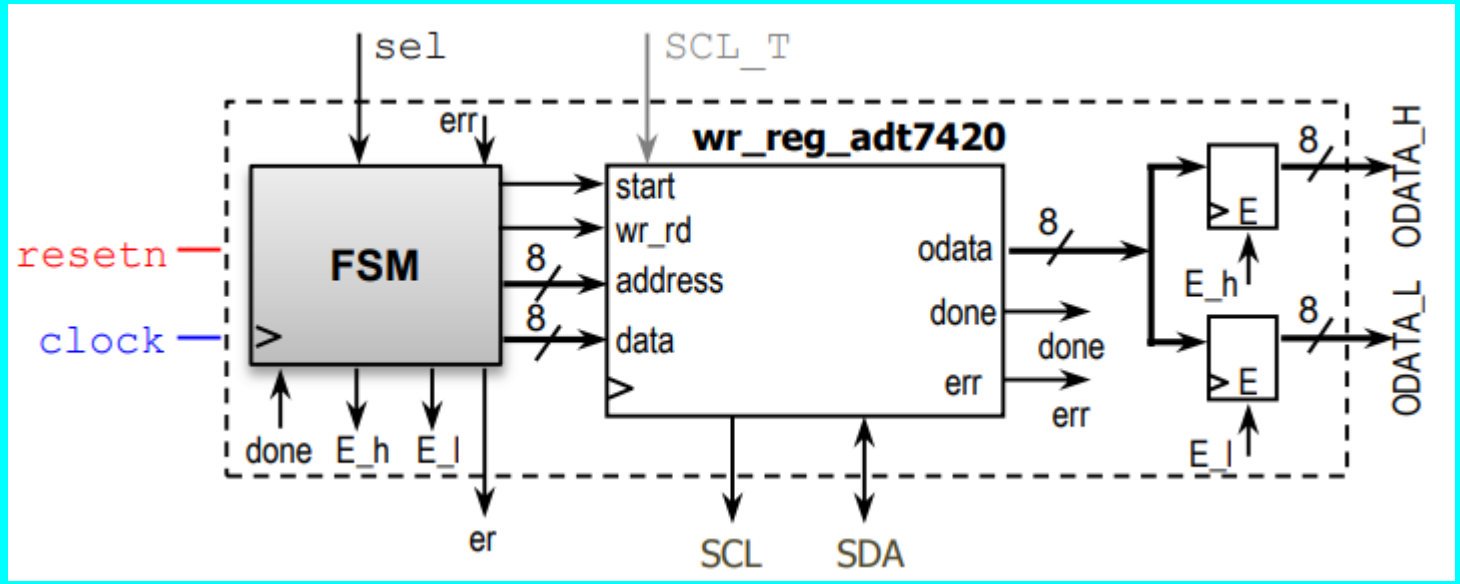
Our idea was to simulate a pool sensor light that would sense what temperature the pool is. Green will be for around room temperature around 15 to 25 degrees Celsius. A Red light will represent a hotter temperature, anything from 26 to 50+ degrees Celsius. Finally. A Blue for a colder temperature, anything from 0 to 15 degrees Celsius.

This device would easily tell people what temperature the pool is before seeing for themselves. Ideally this would be an illuminated light for the entire pool. However this is just a demonstration of the idea, the next slide shows a real life example of our idea.

Block Diagram

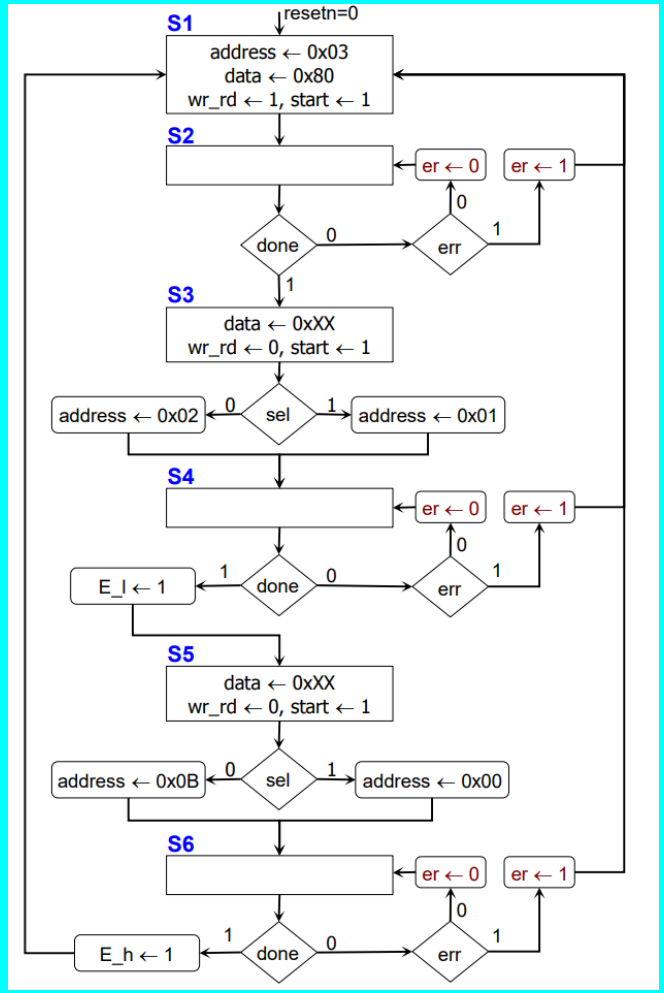


ADT7420 Temperature Sensor *Credit - Dr. Llamocca*



FSM *Credit - Dr. Llamocca*

Reg. Address	Name	Reg. Address	Name
0x00	TEMP_H	0x03	CONFIG
0x01	TEMP_L		
0x02	STATUS	0x0B	ID



Temperature Conversion

- The initial issue we had was interpreting the data from the ADT7420 Temperature Sensor.
- This data is 16 bits data in the format FX.
- By referencing the data sheet, we discovered these 16 bits of data needed to be divided by 128.

TEMPERATURE CONVERSION FORMULAS

16-Bit Temperature Data Format

Positive Temperature = *ADC Code* (dec)/128

Negative Temperature = (*ADC Code* (dec) – 65,536)/128

where *ADC Code* uses all 16 bits of the data byte, including the sign bit.

Negative Temperature = (*ADC Code* (dec) – 32,768)/128

where Bit 15 (sign bit) is removed from the ADC code.

7 Segment Display State Machine

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity FSM is
5      port (resetn,Clock,zC: in std_logic;
6            LEDc: out std_logic_vector(1 downto 0);
7            AN: out std_logic_vector(7 downto 0)
8           );
9
10 end FSM;
11
12 architecture struct of FSM is
13 type state is (s1,s2,s3,S4);
14 signal y: state;
15 begin
16 Transitions: process (resetn, Clock,zC)
17 begin
18 if resetn = '0'
19 then
20 y<=s1;
21 elsif (clock'event and clock = '1') then
22 case y is
23 when S1 =>if zC = '0' then
24     y <= s1;
25     else
26     y <= s2;
27     end if;
28
29     when s2 => if zC = '1' then
30     y <= s3;
31     else
32     y <= s2;
33     end if;
```

```
35     when s3 => if zC = '0' then
36     y<= s3;
37     else
38     y<= s4;
39     end if;
40     when s4 => if zC = '0' then
41         y<= s4;
42         else
43         y<= s1;
44         end if;
45 end case;
46 end if;
47 end process;
48 outputs: process (zC)
49 begin
50
51 AN <= "00000000";
52 LEDc <= "00";
53
54
55 case y is
56 when s1 => LEDc<="00";AN<="11110111";
57
58 when s2 => LEDc<="01";AN<="11111011";
59
60 when s3 => LEDc<="10";AN<="11111101";
61
62 when s4 => LEDc<="11";AN<="11111110";
63
64 end case;
65 end process;
66 end struct;
```

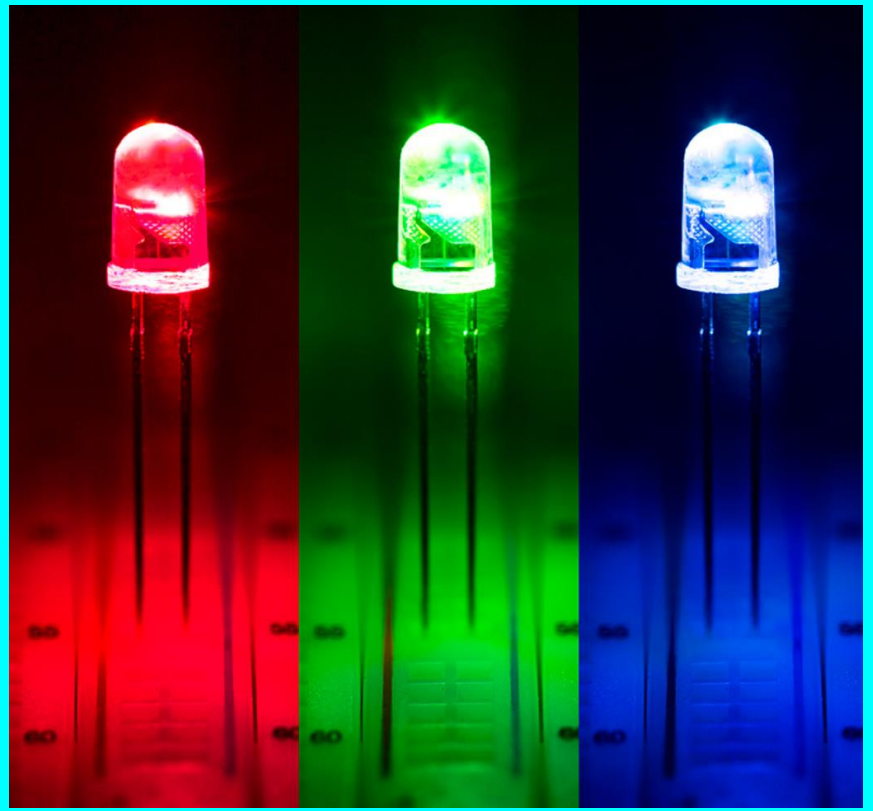
RGB LEDs

Temperature Range in Degrees Celsius

50+ RED LED

30-49 GREEN LED

0-29 BLUE LED



Room Temperature in Degrees Celsius