Spring 2022

California State University, Northridge
Department of Electrical & Computer Engineering

Final Project

Seven Segment Display

May 09, 2022

ECE 520

Written By: Christopher Y. Chen

Purpose:

This final project demonstrates how to implement a seven-segment display to display number or alphabets with zybo 7020 board while using the switches to control the output of the display. Additionally, we used an eight high brightness LED on the side just to verify the correct out and the pin mapping of the hardware PMOD ports. The seven-segment display will first display the number of 1 to 9, then every increment of number will continue with A, B, C …. F. The Four out of the eight LED will indicate the switch activation of each respective switches.

Background:

Pmods are defined by Digilent for small input and output interface boards to provides additional support and extension to the programmable development boards or embedded develop board systems. These Pmods module provides the developer to have sensitive signal conditioning circuits and high-power drive circuits to be placed near the sensor or the processing peripherals.

Hardware list:

1. PMOD Eight Liquid Crystal High Brightness LED (Figure 1)
2. Pmod Seven Segment LED Display
3. Pmod Switches
4. Zybo 7020

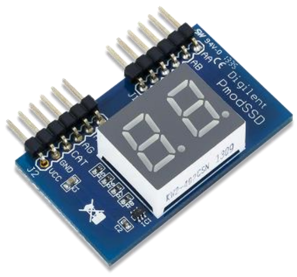
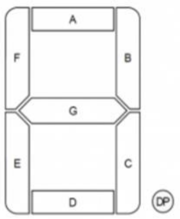
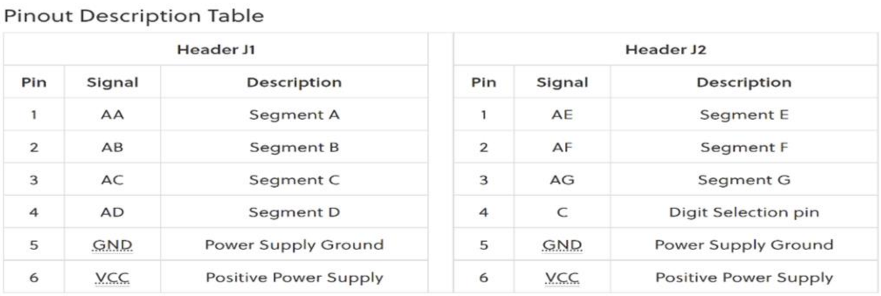


Figure 1

The **Pmod SSD** is a two-digit seven-segment display. To achieve the two digits shows up simultaneity, the developer needs to toggle at a rate of 50Hz to trick the human vision to think that is two digits lits up simultaneously. On the actual board, we used pin 1-6 of the JD and JE high speed Pmod port interfraces for the connect. As shown in the table below, we mapped the code in each respective pins to each respective i/o ports on zybo.



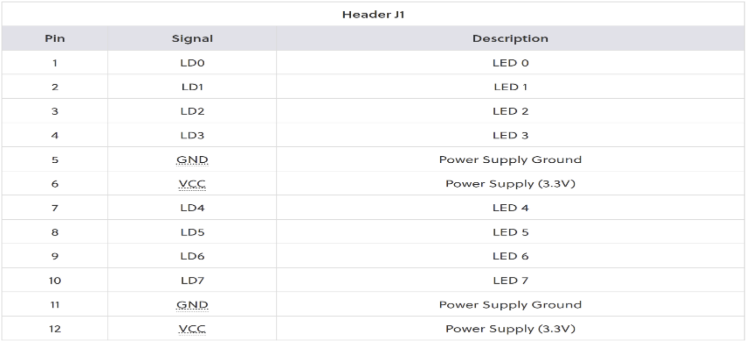


Pinout Description Table

Header J1			Header J2		
Pin	Signal	Description	Pin	Signal	Description
1	AA	Segment A	1	AE	Segment E
2	AB	Segment B	2	AF	Segment F
3	AC	Segment C	3	AG	Segment G
4	AD	Segment D	4	C	Digit Selection pin
5	GND	Power Supply Ground	5	GND	Power Supply Ground
6	VCC	Positive Power Supply	6	VCC	Positive Power Supply

Figure 2

The **Pmod eight liquid crystal high brightness LED** utilizes individual transistors so that each LED can be turned on or off independently. I used the Pmod as the hexadecimal signal display guide for testing and troubleshooting the seven-segment display On the actual board, we used pin 1-6 of the JC high Pmod port interfraces for the connect. As shown in the table below, we mapped the code in each respective pins to each respective i/o ports on zybo.

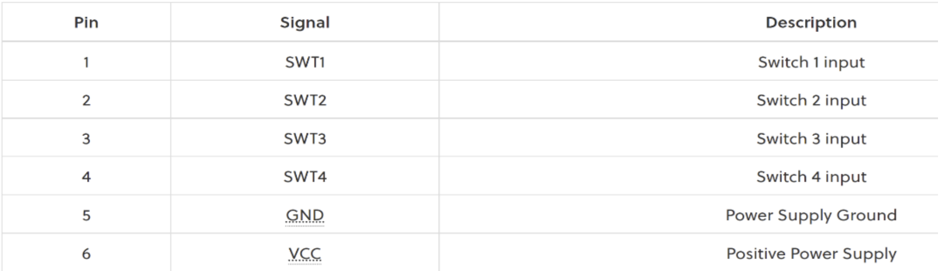


Header J1

Pin	Signal	Description
1	LD0	LED 0
2	LD1	LED 1
3	LD2	LED 2
4	LD3	LED 3
5	GND	Power Supply Ground
6	VCC	Power Supply (3.3V)
7	LD4	LED 4
8	LD5	LED 5
9	LD6	LED 6
10	LD7	LED 7
11	GND	Power Supply Ground
12	VCC	Power Supply (3.3V)

Figure 3

The **Pmod Switch** utilizes four slide switches that users can use as a set of on and off switches or as a set of static binary inputs. Switch 1 will be the LSB and switch 4 will be the MSB.



Pin	Signal	Description
1	SWT1	Switch 1 input
2	SWT2	Switch 2 input
3	SWT3	Switch 3 input
4	SWT4	Switch 4 input
5	GND	Power Supply Ground
6	VCC	Positive Power Supply

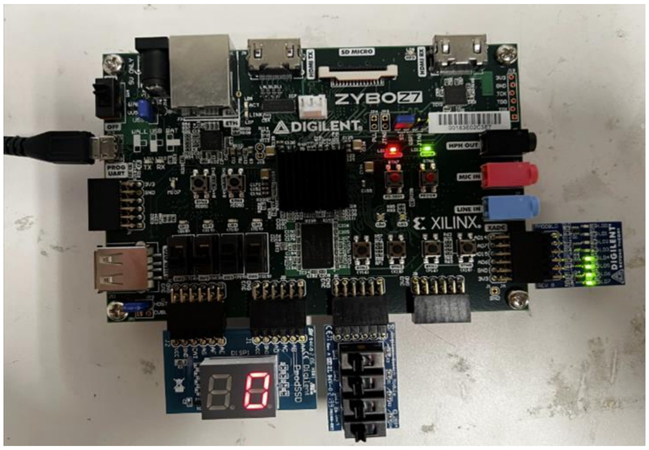
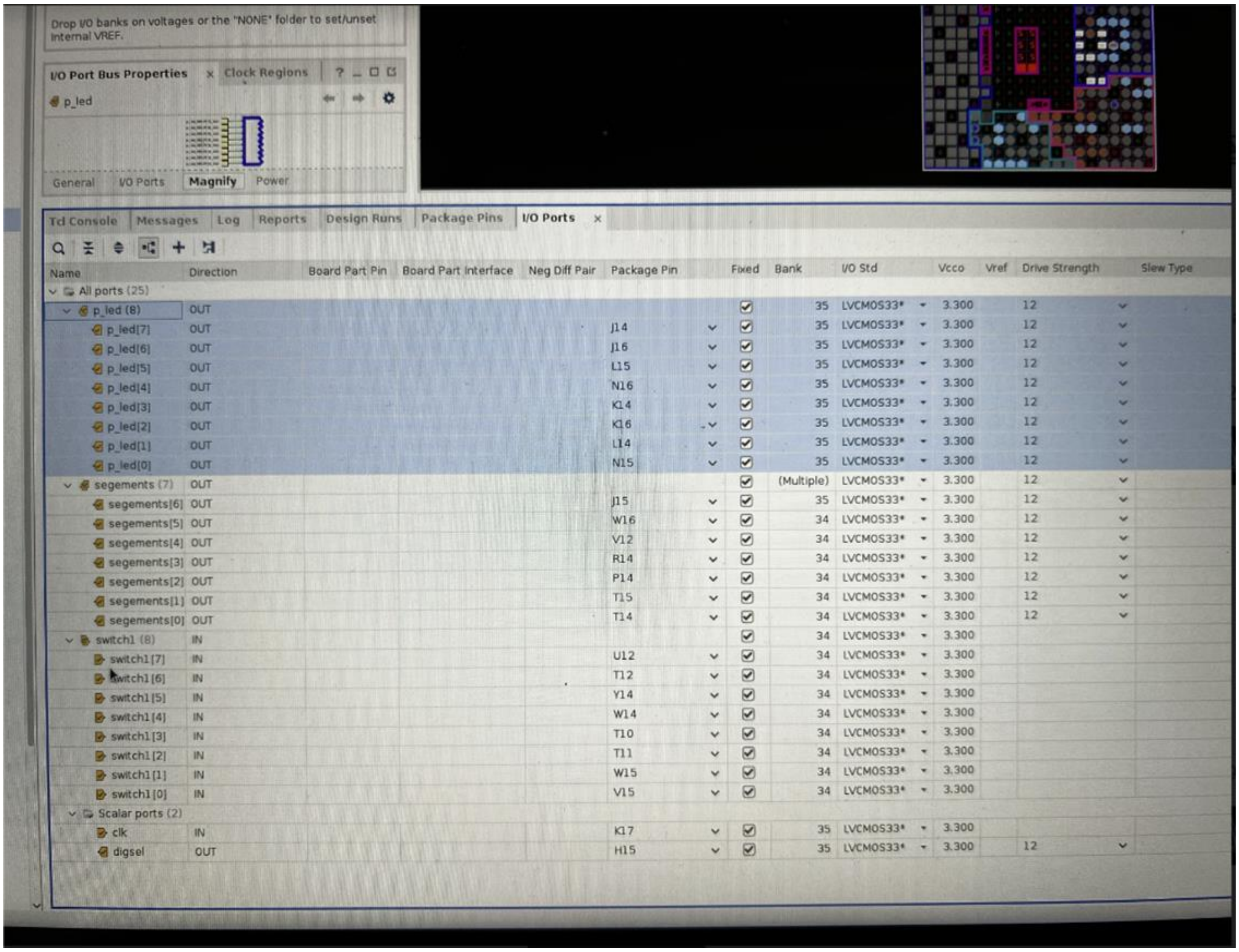


Figure 4

Port Mapping:

	Pmod JA	Pmod JB*	Pmod JC	Pmod JD	Pmod JE	Pmod JF
Pmod Type	**XADC**	**High-Speed**	**High-Speed**	**High-Speed**	**Standard**	**MIO**
Pin 1	N15	V8	V15	T14	V12	MIO-13
Pin 2	L14	W8	W15	T15	W16	MIO-10
Pin 3	K16	U7	T11	P14	J15	MIO-11
Pin 4	K14	V7	T10	R14	H15	MIO-12
Pin 7	N16	Y7	W14	U14	V13	MIO-0
Pin 8	L15	Y6	Y14	U15	U17	MIO-9
Pin 9	J16	V6	T12	V17	T17	MIO-14
Pin 10	J14	W6	U12	V18	Y17	MIO-15

Figure 5 Zybo PMOD I/O port interfaces



Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type
All ports (25)												
p_led (8)	OUT					☑	35	LVCMOS33*	3.300		12	
p_led[7]	OUT				J14	☑	35	LVCMOS33*	3.300		12	
p_led[6]	OUT				J16	☑	35	LVCMOS33*	3.300		12	
p_led[5]	OUT				L15	☑	35	LVCMOS33*	3.300		12	
p_led[4]	OUT				N16	☑	35	LVCMOS33*	3.300		12	
p_led[3]	OUT				K14	☑	35	LVCMOS33*	3.300		12	
p_led[2]	OUT				K16	☑	35	LVCMOS33*	3.300		12	
p_led[1]	OUT				L14	☑	35	LVCMOS33*	3.300		12	
p_led[0]	OUT				N15	☑	35	LVCMOS33*	3.300		12	
segements (7)	OUT					☑	(Multiple)	LVCMOS33*	3.300		12	
segements[6]	OUT				J15	☑	35	LVCMOS33*	3.300		12	
segements[5]	OUT				W16	☑	34	LVCMOS33*	3.300		12	
segements[4]	OUT				V12	☑	34	LVCMOS33*	3.300		12	
segements[3]	OUT				R14	☑	34	LVCMOS33*	3.300		12	
segements[2]	OUT				P14	☑	34	LVCMOS33*	3.300		12	
segements[1]	OUT				T15	☑	34	LVCMOS33*	3.300		12	
segements[0]	OUT				T14	☑	34	LVCMOS33*	3.300		12	
switch1 (8)	IN					☑	34	LVCMOS33*	3.300			
switch1[7]	IN				U12	☑	34	LVCMOS33*	3.300			
switch1[6]	IN				T12	☑	34	LVCMOS33*	3.300			
switch1[5]	IN				Y14	☑	34	LVCMOS33*	3.300			
switch1[4]	IN				W14	☑	34	LVCMOS33*	3.300			
switch1[3]	IN				T10	☑	34	LVCMOS33*	3.300			
switch1[2]	IN				T11	☑	34	LVCMOS33*	3.300			
switch1[1]	IN				W15	☑	34	LVCMOS33*	3.300			
switch1[0]	IN				V15	☑	34	LVCMOS33*	3.300			
Scalar ports (2)												
clk	IN				K17	☑	35	LVCMOS33*	3.300			
digsel	OUT				H15	☑	35	LVCMOS33*	3.300		12	

Code:

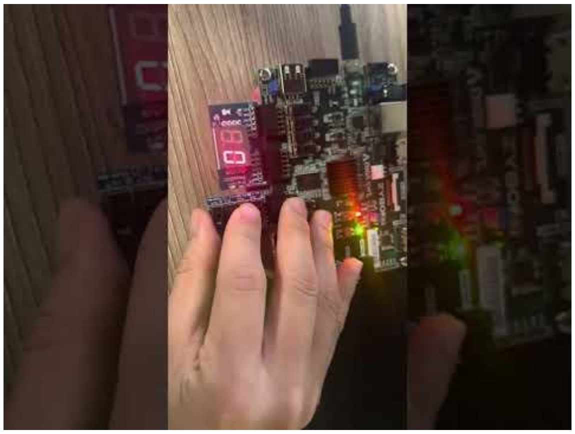
```
`timescale 1ns / 1ps

module pmod_LED(
        input clk,
        input [7:0] switch1,
        output reg [7:0] p_led,
        output reg [6:0] segements,
        output reg digsel
    );


    always @(posedge clk)
        case(switch1 [3:0])
         0: segements <= 7'b0111111;
         1: segements <= 7'b0000110;
         2: segements <= 7'b1011011;
         3: segements <= 7'b1001111;
         4: segements <= 7'b1100110;
         5: segements <= 7'b1101101;
         6: segements <= 7'b1111101;
         7: segements <= 7'b0000111;
         8: segements <= 7'b1111111;
         9: segements <= 7'b1100111;
         10: segements <= 7'b1110111;
         11: segements <= 7'b1111111;
         12: segements <= 7'b0111001;
         13: segements <= 7'b0111111;
         14: segements <= 7'b1111001;
         15: segements <= 7'b1110001;
         endcase

         always @(posedge clk) p_led <= switch1;
         always @(*) digsel = 0;

endmodule
```



The Code will check the switch status at every positive clock cycle. It will insert respective binary numbers into the segments. The respective segements for seven segements display are as follows: GFEDCBA to bits 87654321. At every posigive clock, the switch value will be also be inputed into the eight led and depends on the status of the switch, lights up each leds respectively.

Conclusion:

This is a very fun lab that helped me enforcing understanding of port mapping and simulating the project. The application of the seven-segment display are very common and can be found in most of the digital electronic that requires a display indication of the decimal data.