**ELEC 204 Digital Design Preliminary Lab Report**

Preliminary Lab 3

Name: Metehan Gelgi

Date:11.07.2019

**Question 1**

For getting 2’s complement of binary number change 0 to 1 (also change 1 to 0) then add 1 to this binary number.

1001 -> -7

0111 -> 7

0011-> 3

1111 -> -1

-3 -> 1101

7 -> 0111

10100 -> there is end carry so

= 0100 = 4

1 -> 0001

-7 ->1001

1010 ->no end carry so (-2’s complement)

= -0110 = -6

7 -> 0111

2’s complement

-7-> 1001

­­-3-> 1101

1 ->0001

**Question 2**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 1.** 4-to-1 one-bit Multiplexer

**Question 3**

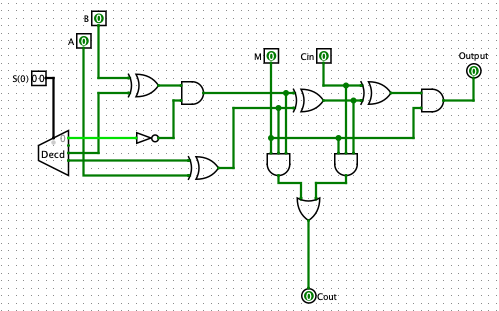
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 2.** Logic Unit with or, and, xor, xnor gates

I used a 4-to-1multiplexer with logic gates. Because I think it is simpler way to do it. First, processing for four gates then choose one of them.

**Question 4**



**Figure 3.** Arithmetic Unit with Decoder, M and full-adder operations (instead of 8-to-1 mux I used Decoder and Full Adder)

**Question 6**

**elektronik eşyalar içeren bir resim

Açıklama otomatik olarak oluşturuldua)**

**saat, nesne içeren bir resim

Açıklama otomatik olarak oluşturuldu**

In my code I am checking A and B before giving them to the 4-bit ALU

That’s why it looks complicated. Left figure is my main part of the program.

**b)**

In the SlowerClock our clock counter is going to increase with “10000000” from starting x“0000000”

Which is goes with 10ns

= 100 Mhz

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**c)**

NET "F[0]" LOC = P16;

NET "F[1]" LOC = P13;

NET "F[2]" LOC = P6;

NET "F[3]" LOC = P3;

NET “SevenSegcontrol[0]” LOC = P50;  
NET “SevenSegcontrol[1]” LOC = P49;  
NET “SevenSegcontrol[2]” LOC = P52;  
NET “SevenSegcontrol[3]” LOC = P56;  
NET “SevenSegcontrol[4]” LOC = P59;  
NET “SevenSegcontrol[5]” LOC = P57;  
NET “SevenSegcontrol[6]” LOC = P60;  
NET “SevenSegcontrol[7]” LOC = P61;  
  
NET “SevenSegbus[7]” LOC = P71;  
NET “SevenSegbus[6]” LOC = P62;  
NET “SevenSegbus[5]” LOC = P65;  
NET “SevenSegbus[4]” LOC = P72;  
NET “SevenSegbus[3]” LOC = P73;  
NET “SevenSegbus[2]” LOC = P98;  
NET “SevenSegbus[1]” LOC = P64;  
NET “SevenSegbus[0]” LOC = P70;  
  
NET “clk” LOC = P40;

F 4-bit output and it is our result for Logic Unit can been on Leds.

SevenSegcontrol and clk is logic input

SevenSegBus logic output for displaying on Seven Segment display screen.

NET "A[0]" LOC = P94;

These are 4-bit inputs which are used in 4-bit ALU

NET "A[1]" LOC = P90;

NET "A[2]" LOC = P88;

NET "A[3]" LOC = P85;

NET "B[0]" LOC = P15;

NET "B[1]" LOC = P12;

NET "B[2]" LOC = P5;

NET "B[3]" LOC = P4;

NET "S[0]" LOC = P34; S is the input for selecting operations on Logic Unit and Arithmetic Unit.

NET "S[1]" LOC = P35;

NET "M" LOC = P78; M is input. We used M to switch between Logic and Arithmetic Units

NET "Cin" LOC = P82; Cin is input for operation to add final result on Arithmetic Unit

NET "Cout" LOC = P77; Cout is the Carry from Arithmetic Operations can be seen on Led