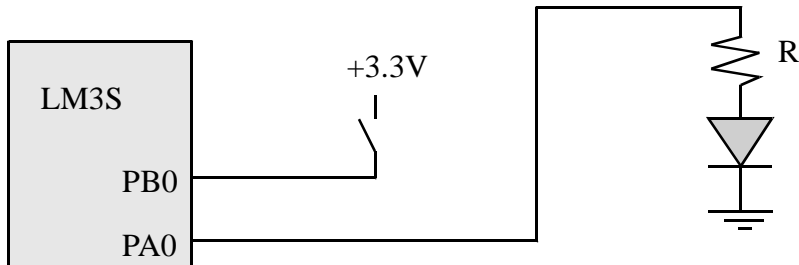


(3) **Question 1.** B) we could decrease the frequency of the bus clock

(4) **Question 2.**  $3^3 3^2 3^1 3^0 = 27, 9, 3, 1$

(3) **Question 3.** Convert both to unsigned. -100 represents 156. 50-156 is -106, which doesn't fit, so **C=0** (mistake). R3 is 150, which is  $128+22=128+16+6=128+16+4+2 = \mathbf{0x96}$ .  
Convert both to signed 50 - -100 = 150, which doesn't fit, so **V=1**.

(20) **Question 4.** Since LED current is less than 8mA, we can drive it from the microcontroller.  $R = (3.2-1V)/2mA = 2.2V/2mA = 1100 \Omega$ . The switch doesn't need a resistor because of the internal pull down.



(10) **Question 5.** Write an assembly subroutine that selects bit 8. The input to the subroutine is a 32-bit number in R0. The output in R0 is 0 if the input bit 8 is 0, and the output is 1 if the input bit 8 is 1.

```
Mask8 AND R0,R0,#0x00000100 ;remove all bits except bit 8
      LSR R0,R0,#8          ;move bit 8 into bit 0 position
      BX LR
```

(10) **Question 6.** Write C function that selects bit 8. The input to the function is an unsigned 32-bit number. The output of the function is 0 if the input bit 8 is 0, and the output is 1 if the input bit 8 is 1.

```
unsigned long Mask8(unsigned long input){ unsigned long output;
  output = input&0x00000100; // select bit 8
  output = output>>8;      // move bit 8 into bit 0 position
  return output;
}
```

(10) **Question 7.** Fill in the boxes with hexadecimal numbers that initializes Port B. Bits 0, 1, and 2 are input. Bits 3 and 5 are output.

```
PortB_Init
  LDR R1, =SYSCTL_RCGC2_R
  LDR R0, [R1]
  ORR R0, R0, #0x02
  STR R0, [R1]
  NOP
  NOP
  LDR R1, =GPIO_PORTB_DIR_R
  LDR R0, [R1]
  ORR R0, R0, #0x28
  BIC R0, R0, #0x07
  STR R0, [R1]
```

```

LDR R1, =GPIO_PORTB_AFSEL_R
LDR R0, [R1]
BIC R0, R0, #0x2F
STR R0, [R1]
LDR R1, =GPIO_PORTB_DEN_R
LDR R0, [R1]
ORR R0, R0, #0x2F
STR R0, [R1]
BX LR

```

**(30) Question 8.** Write an assembly language main program

```

Start  BL   PortB_Init
       LDR  R0,=GPIO_PORTB_DATA_R
Loop   LDR  R1,[R0]      ;read Port B
       ANDS R1,R1,#0x07 ;mask
       BEQ  Toggle3     ;branch if 000
       CMPS R1,#0x07
       BEQ  Toggle3     ;branch if 111
Toggle5 LDR  R1,[R0]      ;read Port B
       EOR  R1,R1,#0x20 ;bit 5
       STR  R1,[R0]      ;write to Port B
       B    Loop
Toggle3 LDR  R1,[R0]      ;read Port B
       EOR  R1,R1,#0x08 ;bit 3
       STR  R1,[R0]      ;write to Port B
       B    Loop

```

**(10) Question 9.** Write a C language main program that first calls the initialization. E.g., execute `PortB_Init()`; defined in Question 7. Then perform this infinite loop: read the inputs; if all three inputs are equal to each other (inputs are 000 or 111) then toggle output bit 3, otherwise (inputs are 001, 010, 011, 100, 101, 110) toggle output bit 5. Write friendly code. With this definition

```
#define PORTB (*(volatile unsigned long *)0x400053FC)
```

You will be able to read and write to Port B. For example

```

    n = PORTB; // reads all 8 bits of Port B into variable n
    PORTB = m; // write all 8 bits of Port B with data from m
void main(void){ unsigned long input;
  PortB_Init();
  while(1){
    input = PORTB&0x07;
    if((input==0)|| (input==0x07)){
      PORTB = PORTB^0x08; // toggle bit 3
    } else{
      PORTB = PORTB^0x20; // toggle bit 5
    }
  }
}

```