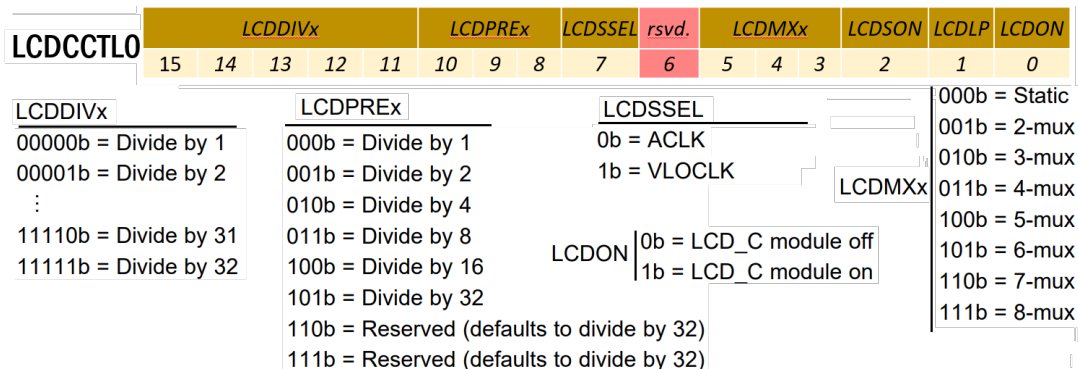


1. (4 points) For the following LCD initialization function, complete the code so that:

- (a) (2 points) Assuming that LCD clock source is connected to 32.768 KHz ACLK, the LCD frequency is set to 33 Hz in 4-mux mode.
- (b) (2 points) Assuming that LCD clock source is connected to 500 KHz VLOCLK, the LCD frequency is set to 625 Hz in 6-mux mode.



```
void Initialize_LCD() {
    PJSEL0 = BIT4 | BIT5;
    LCDCPCTL0 = 0xFFD0;
    LCDCPCTL1 = 0xF83F;
    LCDCPCTL2 = 0x00F8;

    CSCTL0_H = CSKEY >> 8;
    CSCTL4 &= ~LFXTOFF;
    do {
        CSCTL5 &= ~LFXTOFFG;
        SFRIFG1 &= ~OFIFG;
    } while (SFRIFG1 & OFIFG);    fault flag
    CSCTL0_H = 0;

    LCDCTL0 = _____;
    LCDCVCTL = VLCD_1 | VLCDREF_0 | LCDPEN;
    LCDCCPCTL = LCDCPCLKSYNC;
    LCDCMEMCTL = LCDCLRM;
    _____;
    return;
}
```

Answer (part a): ACLK = 32,768 Hz → needs to be (approximately) 1000x slower to be 33 Hz. So, the chosen divider and pre-scaler must create this ratio based on the following equation.

$$LCDCTL0 \text{ sets the LCD frequency} \gg f_{LCD} = \frac{f_{clk}}{2^{LCDPREx(LCDDIVx+1)}}$$

So, the configuration of LCD must be as follows. Also, the last line, should be LCD ON (we make LCD ON once the configuration is done).

```
LCDCTL0 = LCDDIV_31 | LCDPRE_5 | LCDSEL_0 | LDMx_3; // Filled line
LCDCVCTL = VLCD_1 | VLCDREF_0 | LCDPEN;
LCDCCPCTL = LCDCPCLKSYNC;
LCDCMEMCTL = LCDCLRM;
LCDCTL0 = LCDON; // Filled line
```

Answer (part b): Part (b) is similar to Part (a). The clock source is changed (to VLOCLK) and the mux-mode value is changed. Also, based on new clock frequency (500 KHz), the required frequency (625 Hz) is 800x slower. So, divider and pre-scaler should make this 800x slower frequency for the LCD (which is 25x32). Similar to Part (a), the last line should be also LCD ON.

```
LCDCTL0 = LCDDIV_24 | LCDPRE_5 | LCDSEL_1 | LDMx_5; // Filled line
LCDCVCTL = VLCD_1 | VLCDREF_0 | LCDPEN;
LCDCCPCTL = LCDCPCLKSYNC;
LCDCMEMCTL = LCDCLRM;
LCDCTL0 = LCDON; // Filled line
```

2. (5 points) Complete the following code in a way that it cycles digits 0 → 9 on LCD position A1 (LCDM10 and LCDM11) every 1 second. Assume that the function `Initialize_LCD()` correctly initializes the LCD in its default mode (ACLK, 4-mux), and the function `delay_ms(int n)` provides a delay of `n` milliseconds.

```
#include <msp430fr6989.h>

const unsigned char LCD_Shapes_LB[10] = {0xFC, 0x60, 0xDB, 0xF3, 0x67, 0xB7, 0xBF, 0xE0, 0xFF, 0xF7};
const unsigned char LCD_Shapes_HB[10] = {0x28, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

void Initialize_LCD(void);
void delay_ms(int n);

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;
    PM5CTL0 &= ~LOCKLPM5;
    Initialize_LCD();
    while (1)
    {
        for (_____) {
            _____;
            _____;
        }
    }
}
```

Answer: The code loops through digits 0–9, writes each digit's segment pattern from `LCD_Shapes[]` to the LCD memory bytes (LCDM10, LCDM11) that control position A1, and waits 1 second using `delay_ms(1000)` before showing the next digit. This creates a continuous counting display on the LCD.

```
unsigned int i;
for (i = 0; i < 10; i++) {
    LCDM10 = LCD_Shapes_LB[i];    // Display digit i on A1 (lower byte)
    LCDM11 = LCD_Shapes_HB[i];    // Display digit i on A1 (higher byte)
    delay_ms(1000);               // Wait 1 second
}
```

3. (5 points) Bonus

Write a function that receives the current clock time as three separate integer inputs: (1) hour, (2) minute, and (3) second; and displays the time on the MSP430 LCD. The function should correctly show the time on the LCD in the format HH:MM:SS. Complete the function below.

```
#include <msp430fr6989.h>

const unsigned char LCD_Shapes_LB[10] = {0xFC, 0x60, 0xDB, 0xF3, 0x67, 0xB7, 0xBF, 0xE0, 0xFF, 0xF7};
const unsigned char LCD_Shapes_HB[10] = {0x28, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

static void lcd_show_time(int hour, int minute, int second){

    LCDM10 = LCD_Shapes_LB[_____];
    LCDM11 = LCD_Shapes_HB[_____];

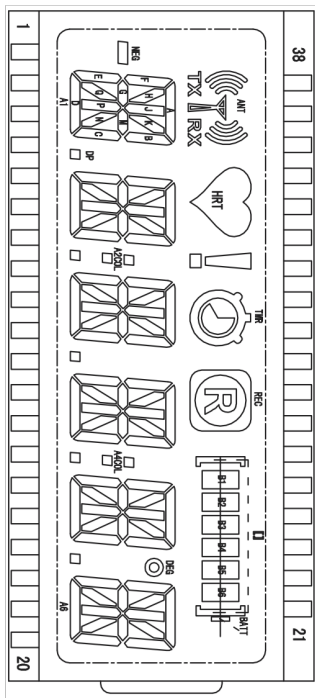
    LCDM6 = _____;
    LCDM7 = _____;

    LCDM4 = LCD_Shapes_LB[_____];
    LCDM5 = LCD_Shapes_HB[_____];

    LCDM19 = _____;
    LCDM20 = _____;

    LCDM16 = LCD_Shapes_LB[_____];
    LCDM15 = LCD_Shapes_HB[_____];

    LCDM8 = _____;
    LCDM9 = _____;
}
```



LCDMEM	Port Pin	FR6989 Pin	LCD Pin	COM3	COM2	COM1	COM0	Port Pin	FR6989 Pin	LCD Pin	COM3	COM2	COM1	COM0
LCDM22	P2.4	S43						P2.5	S42					
LCDM21	P2.6	S41						P2.7	S40					
LCDM20	P10.2	S39	16	A4H	A4J	A4K	A4P	P5.0	S38	15	A4Q	A4COL	A4N	A4DP
LCDM19	P5.1	S37	14	A4A	A4B	A4C	A4D	P5.2	S36	13	A4R	A4F	A4G	A4M
LCDM18	P5.3	S35	34	B5	B3	B1	⏏	P3.0	S34					
LCDM17	P3.1	S33						P3.2	S32					
LCDM16	P6.7	S31	20	A5H	A5J	A5K	A5P	P7.5	S30	19	A5Q	DEG	A5N	A5DP
LCDM15	P7.6	S29	18	A5A	A5B	A5C	A5D	P10.1	S28	17	A5E	A5F	A5G	A5M
LCDM14	P7.7	S27	33	B6	B4	B2	BATT	P3.3	S26					
LCDM13	P3.4	S25						P3.5	S24					
LCDM12	P3.6	S23						P3.7	S22					
LCDM11	P8.0	S21	4	A1H	A1J	A1K	A1P	P8.1	S20	3	A1Q	NEG	A1N	A1DP
LCDM10	P8.2	S19	2	A1A	A1B	A1C	A1D	P8.3	S18	1	A1E	A1F	A1G	A1M
LCDM9	P7.0	S17	38	A6H	A6J	A6K	A6P	P7.1	S16	37	A6Q	TX	A6N	RX
LCDM8	P7.2	S15	36	A6A	A6B	A6C	A6D	P7.3	S14	35	A6E	A6F	A6G	A6M
LCDM7	P7.4	S13	8	A2H	A2J	A2K	A2P	P5.4	S12	7	A2Q	A2COL	A2N	A2DP
LCDM6	P5.5	S11	6	A2A	A2B	A2C	A2D	P5.6	S10	5	A2E	A2F	A2G	A2M
LCDM5	P5.7	S9	12	A3H	A3J	A3K	A3P	P4.4	S8	11	A3Q	ANT	A3N	A3DP
LCDM4	P4.5	S7	10	A3A	A3B	A3C	A3D	P4.6	S6	9	A3R	A3F	A3G	A3M
LCDM3	P4.7	S5						P10.0	S4	32	TMR	HRT	REC	!
LCDM2	P4.0	S3						P4.1	S2					
LCDM1	P1.4	S1						P1.5	S0					

Answer: The digits are provided separately. Each of the time components, i.e., hour, minute, and second, is displayed using two digits on the LCD (The tens digit and ones digit of each value should be handled separately). For each digit, the corresponding low-byte (LCD_Shapes_LB) and high-byte (LCD_Shapes_HB) patterns are written to two consecutive LCD memory registers. Also, colons (:) between them (one between hours–minutes and one between minutes–seconds) should be on by the A2COL and A4COL segments. The function is as follows:

```
#include <msp430fr6989.h>
```

```
const unsigned char LCD_Shapes_LB[10] = {0xFC, 0x60, 0xDB, 0xF3, 0x67, 0xB7, 0xBF, 0xE0, 0xFF, 0xF7};
const unsigned char LCD_Shapes_HB[10] = {0x28, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

```
static void lcd_show_time(int hour, int minute, int second){
```

```
    LCDM10 = LCD_Shapes_LB[hour/10];
    LCDM11 = LCD_Shapes_HB[hour/10];
```

```
    LCDM6 = LCD_Shapes_LB[hour%10];
    LCDM7 = LCD_Shapes_LB[hour%10] | 0x04; // Turn ON A2COL;
```

```
    LCDM4 = LCD_Shapes_LB[minute/10];
    LCDM5 = LCD_Shapes_HB[minute/10];
```

```
    LCDM19 = LCD_Shapes_LB[minute%10];
    LCDM20 = LCD_Shapes_LB[minute%10] | 0x04; // Turn ON A4COL;
```

```
    LCDM16 = LCD_Shapes_LB[second/10];
    LCDM15 = LCD_Shapes_HB[second/10];
```

```
    LCDM8 = LCD_Shapes_LB[second%10];
    LCDM9 = LCD_Shapes_LB[second%10];
```