

Solved Exercise:

Write a program to simulate 4-digit BCD up counter on the multiplexed seven segment display.

```
#include <LPC17xx.h>
#include <stdio.h>
unsigned int seg_select[4] = {0<<23, 1<<23, 2<<23, 3<<23};
unsigned int dig1=0x00, dig2=0x00, dig3=0x00, dig4=0x00;
unsigned int seg_count=0x00, temp1=0x00;
unsigned char array_dec[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
unsigned long int i=0;
void delay(void);
void display(void);
int main(void)
{

LPC_PINCON->PINSEL0 &= 0xFF0000FF; //P0.4 to P0.11 GPIO data lines
LPC_PINCON->PINSEL3 &= 0xFFC03FFF; //P1.23 to P1.26 GPIO enable lines
LPC_GPIO0->FIODIR |= 0x00000FF0; //P0.4 to P0.11 output

LPC_GPIO1->FIODIR |= 0x07800000; //P1.23 to P1.26 output
while(1)
{
    delay();
    display();
    seg_count +=1;
    if(seg_count == 0x04)
    {
        seg_count = 0x00;
        dig1 +=1;
        if(dig1 == 0x0A)
        {
            dig1 = 0;
            dig2 +=1;
            if(dig2 == 0x0A)
            {
                dig2 = 0;
                dig3+=1;
                if(dig3 == 0x0A)
                {
                    dig3 = 0;
                    dig4 += 1;
                    if(dig4 == 0x0A)
                    {
                        dig4 = 0;
                    } //end of dig4
                }
            }
        }
    }
}
```

```

        } //end of dig3
        } //end of dig2
        } //end of dig1
        } //end of seg_count
    } //end of while(1)

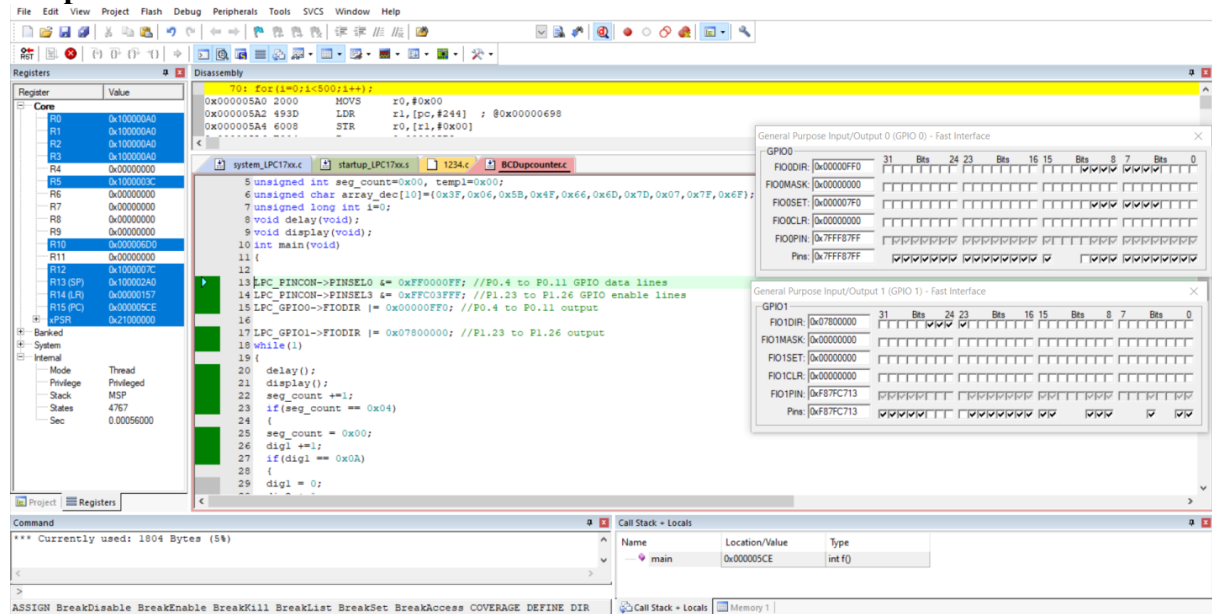
} //end of main

void display(void) //To Display on 7-segments
{
    LPC_GPIO1->FIOPIN = seg_select[seg_count];
    if(seg_count == 0x00) // For Segment U8
    {
        temp1 = dig1;
    }
    else if(seg_count == 0x01) // For Segment U9
    {
        temp1 = dig2;
    }
    else if(seg_count == 0x02) // For Segment U10
    {
        temp1 = dig3;
    }
    else if(seg_count == 0x03) // For Segment U11
    {
        temp1 = dig4;
    }
    LPC_GPIO0->FIOPIN = array_dec[temp1]<<4; // Taking Data Lines for 7-Seg
    for(i=0;i<500;i++);
}

void delay(void)
{ unsigned int i;
  for(i=0;i<60000;i++);
}

```

Output:



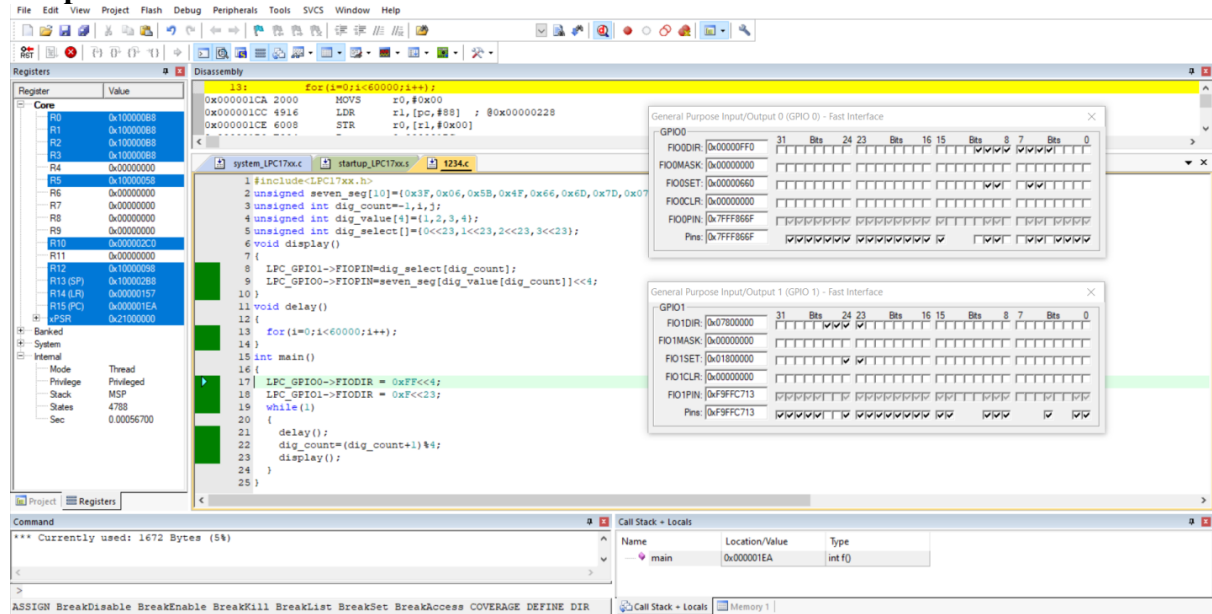
Lab Exercises:

1. Write a C program to display the number “1234” serially in the seven segment display.

Program:

```
#include<LPC17xx.h>
unsigned seven_seg[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
unsigned int dig_count=-1,i,j;
unsigned int dig_value[4]={1,2,3,4};
unsigned int dig_select[]={0<<23,1<<23,2<<23,3<<23};
void display()
{
    LPC_GPIO1->FIOPIN=dig_select[dig_count];
    LPC_GPIO0->FIOPIN=seven_seg[dig_value[dig_count]]<<4;
}
void delay()
{
    for(i=0;i<60000;i++);
}
int main()
{
    LPC_GPIO0->FIODIR = 0xFF<<4;
    LPC_GPIO1->FIODIR = 0xF<<23;
    while(1)
    {
        delay();
        dig_count=(dig_count+1)%4;
        display();
    }
}
```

Output:



2. Write a C program to simulate a 4-digit BCD down counter.

Program:

```

#include<LPC17XX.h>
void delay(void);
int main()
{
    unsigned int i;
    unsigned int c_flag=0;
    unsigned int digit_value[4]={9,9,9,9};
    unsigned long seven_seg[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
    LPC_PINCON->PINSEL0 =0X00FF000FF;
    LPC_PINCON->PINSEL3 =0xFFFC03FFF;
    LPC_GPIO0->FIODIR=0xFF<<4;
    LPC_GPIO1->FIODIR=0xFF<<23;
    while(1)
    {
        delay();
        for(i=0;i<4;i++)
        {
            LPC_GPIO1->FIOPIN=i<<23;
            LPC_GPIO0->FIOPIN=seven_seg[digit_value[i]]<<4;
        }
        c_flag=0;
        for(i=0;i<4;i++)
        {
            if(i==0)
            {
                if(digit_value[i]<c_flag+1)
                {
                    digit_value[i]=digit_value[i]-1-c_flag+10;
                    c_flag=1;
                }
            }
        }
    }
}

```

```

}
else
{
digit_value[i]=digit_value[i]-1-c_flag;
c_flag=0;
}
}
else
{
if(digit_value[i]<c_flag)
{
digit_value[i]=digit_value[i]-c_flag+10;
c_flag=1;
}
else
{
digit_value[i]=digit_value[i]-c_flag;
c_flag=0;
}
}
}
}
return 2;
}
void delay(void)
{ unsigned int i;
for(i=0;i<1000;i++);
}

```

Output:

The screenshot shows the Keil uVision IDE with the following components:

- Registers:** A list of registers (R0-R15, PC, PSR) with their current values. R0 is 0x10000000, R1 is 0x10000000, R2 is 0x10000000, R3 is 0x10000000, R4 is 0x00000000, R5 is 0x10000000, R6 is 0x00000000, R7 is 0x00000000, R8 is 0x00000000, R9 is 0x00000000, R10 is 0x00000000, R11 is 0x00000000, R12 is 0x10000000, R13 (SP) is 0x10000000, R14 (LR) is 0x00000000, R15 (PC) is 0x00000000, and PSR is 0x10000000.
- Disassembly:** The assembly code for the delay function is shown. It starts with a loop that decrements a counter (R0) until it reaches zero. The code is as follows:


```

4: {
5: unsigned int i;
6: unsigned int c_flag=0;
7: while(i)
8: {
9: i--;
10: if(i==0)
11: {
12: c_flag++;
13: }
14: }
15: }
16: }
17: }
18: }
19: }
20: }
21: }
22: }
23: }
24: }
25: }

```
- Command:** The command window shows the current memory usage: "Currently used: 1780 Bytes (5%)".
- Call Stack - Locals:** The call stack shows the current function call stack. The top entry is "main" at location 0x000001B2, with type "int f()". Below it is "i" at location 0x00000000, with type "auto - unsigned int". Below that is "c_flag" at location 0x10000004, with type "auto - unsigned int".
- GPIO0 and GPIO1:** Two windows showing the GPIO configuration. GPIO0 is configured with FIODIR=0x00000000, FIOMASK=0x00000000, FIOSET=0x00000000, FIOCLR=0x00000000, FIOPIN=0x7FFF82F0, and Pins=0x7FFF82FF. GPIO1 is configured with FIODIR=0x07800000, FIOMASK=0x00000000, FIOSET=0x01800000, FIOCLR=0x00000000, FIOPIN=0x0180C713, and Pins=0xC99FC713.