Ayush Goyal

190905522 CSE D 62

ES Lab 7(contd.) Week 8 – Programs on Multiplexed Seven Segment Display

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

CODE:

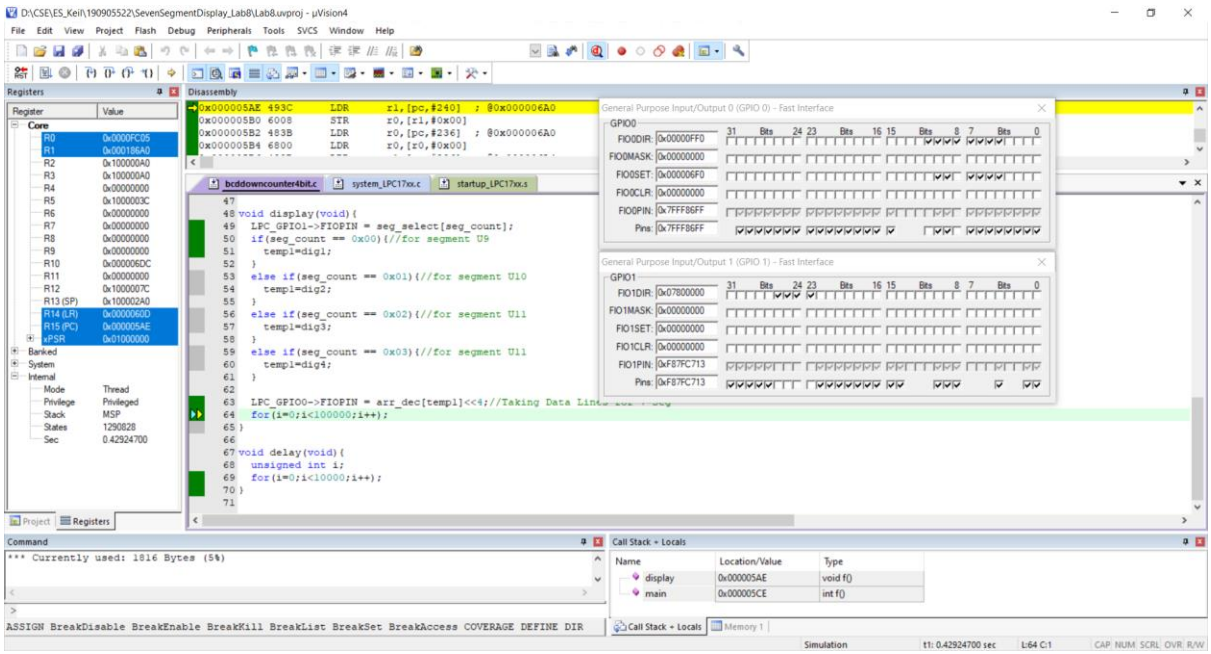
```
#include<LPC17xx.h>
#include<stdio.h>

unsigned int seg_select[4] = {0<<23, 1<<23, 2<<23, 3<<23};
int dig1=0x09, dig2=0x09, dig3=0x09, dig4=0x09;
unsigned int seg_count=0x00, temp1=0x00;
unsigned char arr_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){
    SystemInit();
    SystemCoreClockUpdate();

    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    LPC_PINCON->PINSEL3 &= 0xFFC03FFF;

    LPC_GPIO0->FIODIR |= 0x00000FF0;
    LPC_GPIO1->FIODIR |= 0x07800000;

    while(1){
        delay();
        display();
        seg_count +=1;
        if(seg_count == 0x04){
            seg_count = 0x00;
            dig1-=1;
            if(dig1 < 0){
                dig1=0x09;
                dig2-=1;
                if(dig2 < 0){
                    dig2=0x09;
                    dig3-=1;
                    if(dig3 < 0){
                        dig3=0x09;
                        dig4-=1;
```

```
                        if(dig4 < 0){
                            dig4=0x09;
                        }//eod4
                    }//eod3
                }//eod2
            }//eod1
        }//eosegcount
    }//eowhile
}//eomain

void display(void){
    LPC_GPIO1->FIOPIN = seg_select[seg_count];
    if(seg_count == 0x00){//for segment U9
        temp1=dig1;
    }
    else if(seg_count == 0x01){//for segment U10
        temp1=dig2;
    }
    else if(seg_count == 0x02){//for segment U11
        temp1=dig3;
    }
    else if(seg_count == 0x03){//for segment U11
        temp1=dig4;
    }

    LPC_GPIO0->FIOPIN = arr_dec[temp1]<<4;//Taking Data Lines for 7-Seg
    for(i=0;i<100000;i++);
}

void delay(void){
    unsigned int i;
    for(i=0;i<10000;i++);
}
```

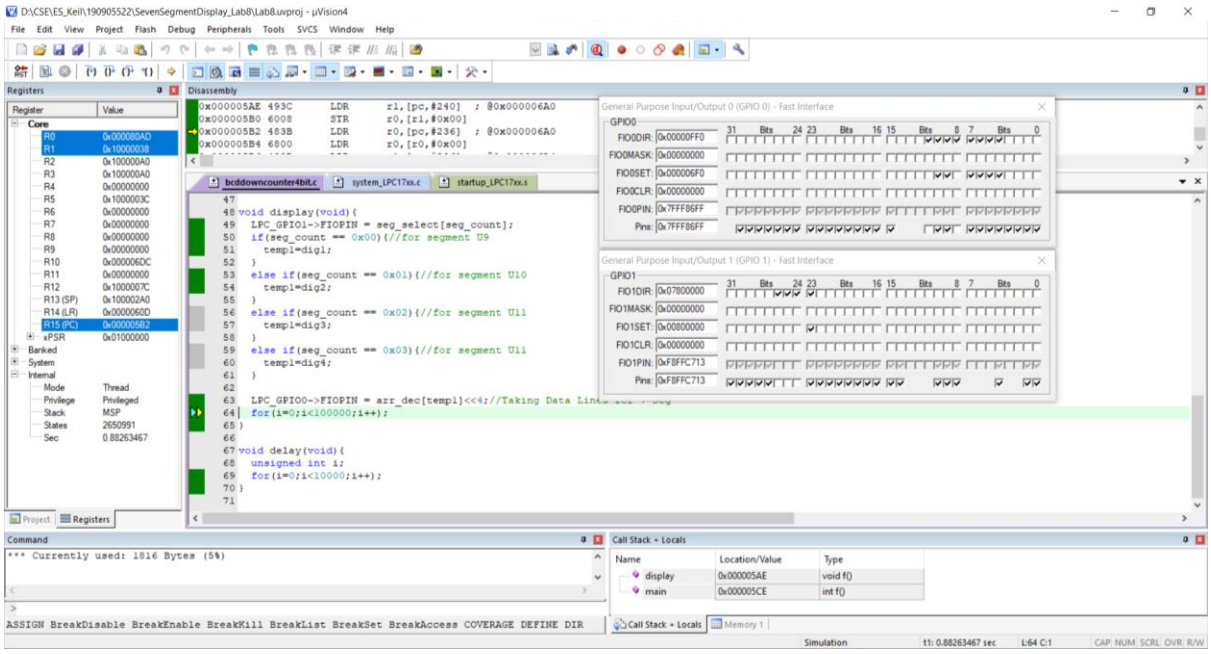
OUTPUT:

In the first iteration of the program the BCD number to be displayed will be 9999 then 9998 and so on and so forth till 0000. So, we can see that for 9999 the digits made 9 will be bits 0, 1, 2 and 3:

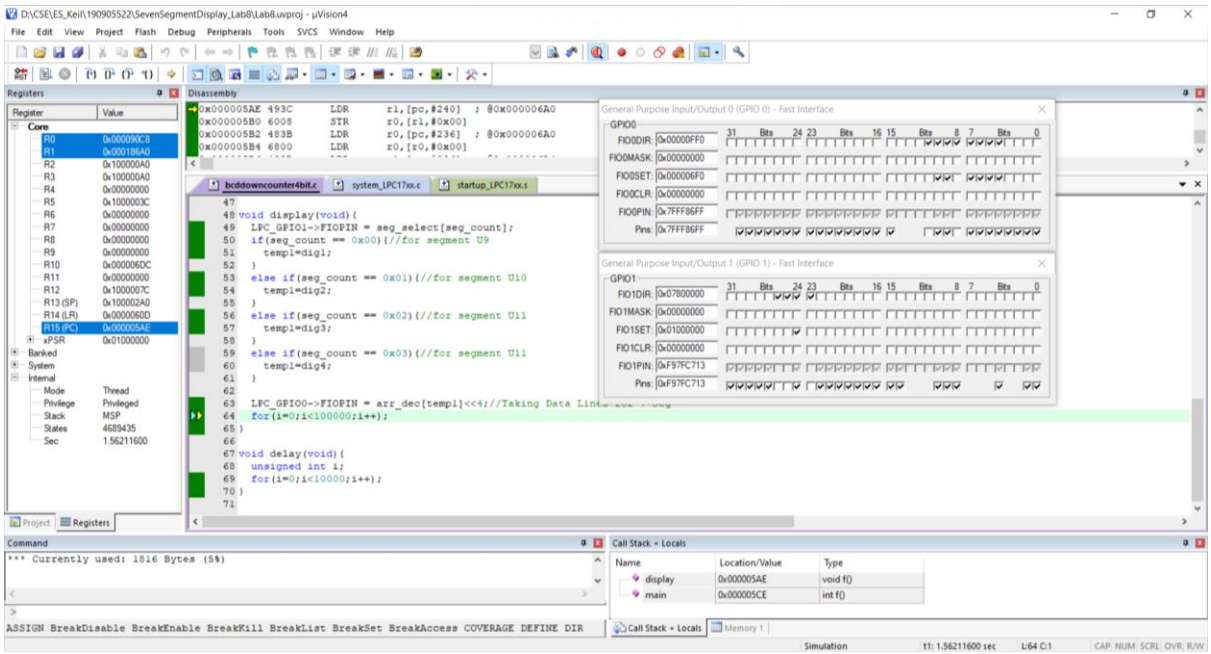
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



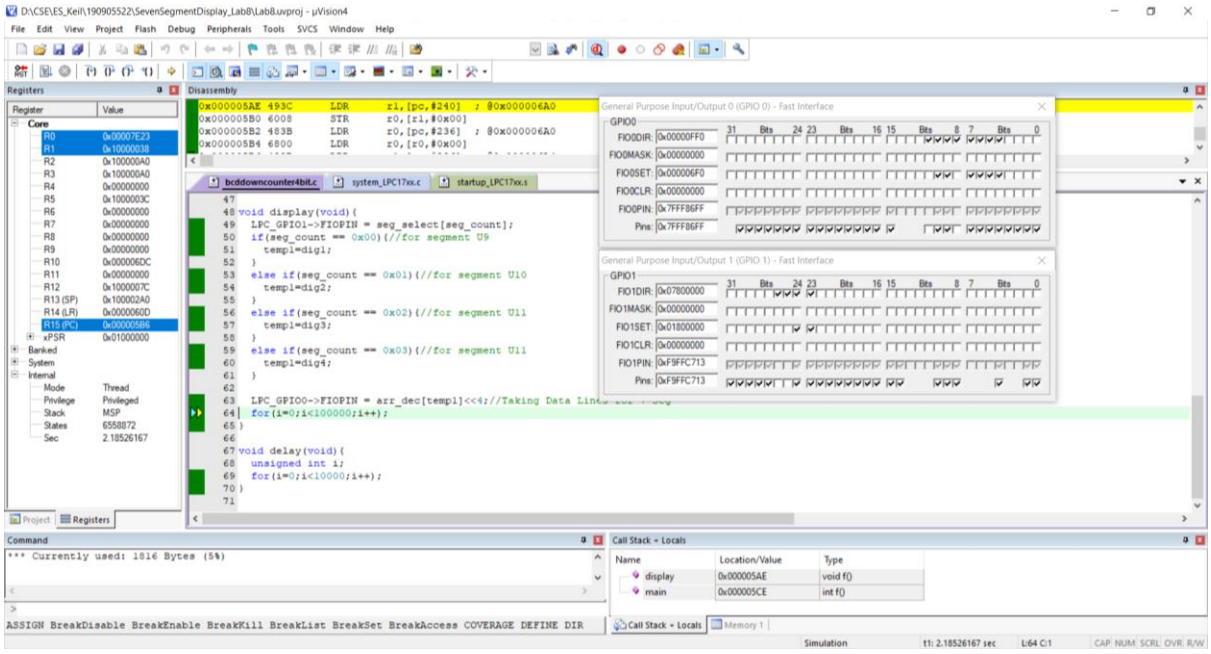
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:

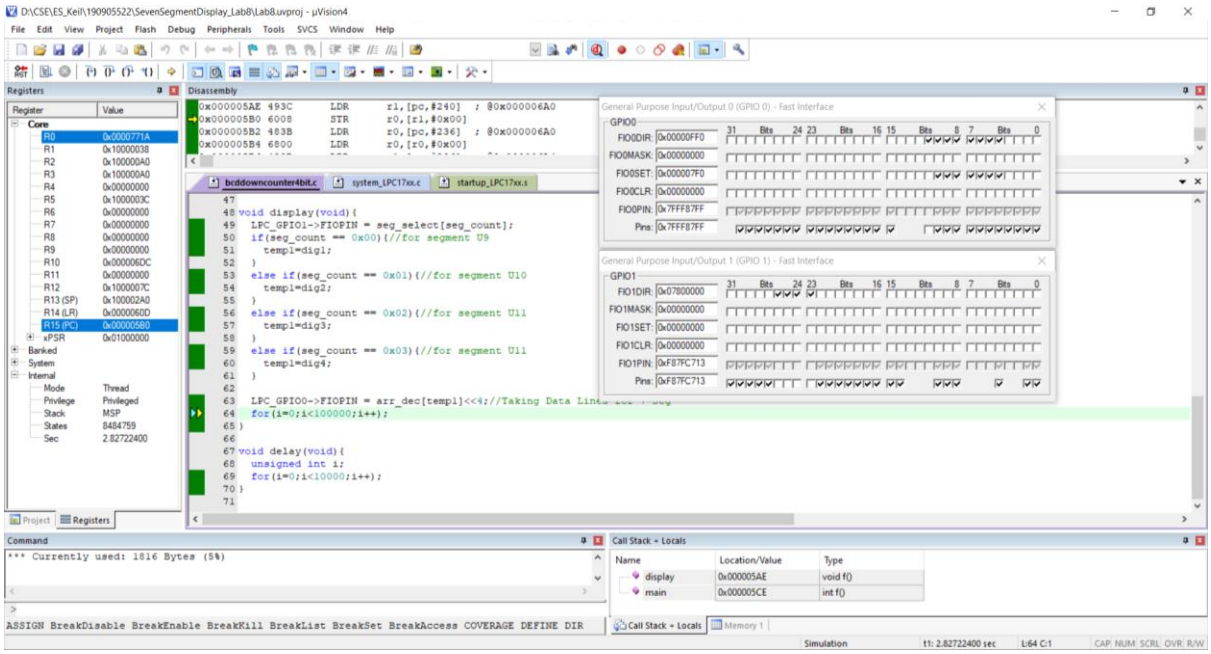


In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:

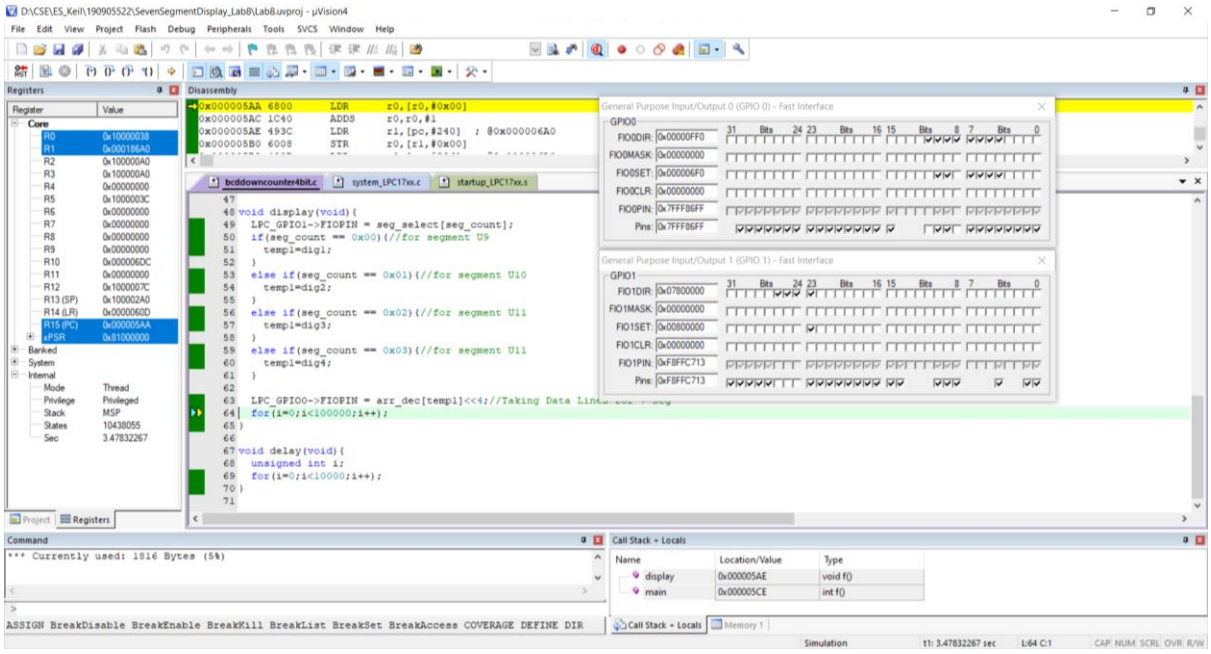


Now in the second iteration the value displayed should be 9998:

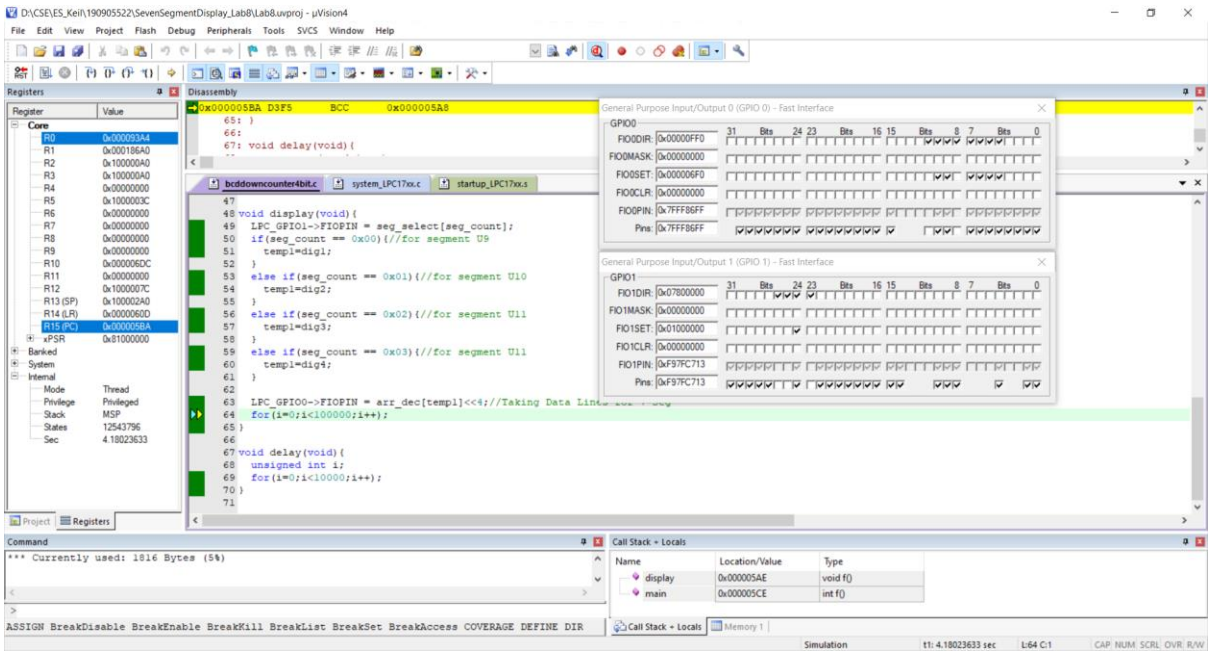
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit 8 in GPIO0 which is of value 0111 1111 in binary:



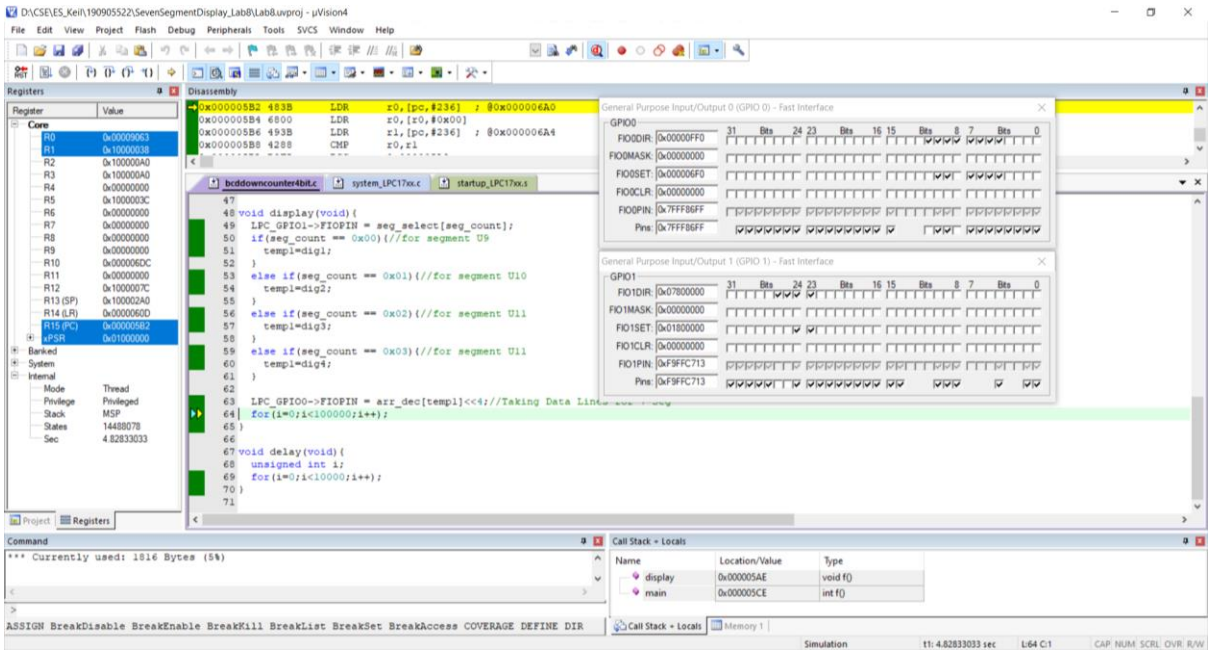
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



Thus we can see the working of a BCD down counter.

3. Write a C program for 4-digit BCD up/down counters on seven segment displays using a switch.

CODE:

```
#include<LPC17xx.h>
#include<stdio.h>

unsigned int seg_select[4] = {0<<23, 1<<23, 2<<23, 3<<23};
int dig1=0x00, dig2=0x00, dig3=0x00, dig4=0x00;
unsigned int seg_count=0x00, temp1=0x00;
unsigned char arr_dec[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
unsigned long int i=0;
unsigned int k;
void delay(void);
void display(void);

int main(void){
    SystemInit();
    SystemCoreClockUpdate();

    LPC_PINCON->PINSEL0 &= 0xFF0000FF;//output
    LPC_PINCON->PINSEL3 &= 0xFFC03FFF;//bit
    LPC_PINCON->PINSEL4 &=0xFCFFFFFF;//switch

    LPC_GPIO0->FIODIR |= 0x00000FF0;//output
    LPC_GPIO1->FIODIR |= 0x07800000;//bit
    LPC_GPIO2->FIODIR &= 0xFFFFEFFF;//switch

    while(1){
        k = LPC_GPIO2->FIOPIN >> 12; //We read input from 2.12
        k &= 0x00000001;
        delay();
        display();
        seg_count +=1;
        if(seg_count == 0x04){
            seg_count = 0x00;

            if(k==1){
                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;
                            }//eod4
                        }//eod3
                    }//eod2
                }//eod1
            }
            else{
                dig1-=1;
                if(dig1 < 0){
                    dig1=0x09;
                    dig2-=1;
                    if(dig2 < 0){
                        dig2=0x09;
                        dig3-=1;
                        if(dig3 < 0){
                            dig3=0x09;
                            dig4-=1;
                            if(dig4 < 0){
                                dig4=0x09;
                            }//eod4
                        }//eod3
                    }//eod2
                }//eod1
            }//eoelse

        }//eosegcount
    }//eowhile
}//eomain

void display(void){
    LPC_GPIO1->FIOPIN = seg_select[seg_count];
    if(seg_count == 0x00){//for segment U9
        temp1=dig1;
    }
    else if(seg_count == 0x01){//for segment U10
        temp1=dig2;
    }
    else if(seg_count == 0x02){//for segment U11
        temp1=dig3;
    }
    else if(seg_count == 0x03){//for segment U11
        temp1=dig4;
    }

    LPC_GPIO0->FIOPIN = arr_dec[temp1]<<4;//Taking Data Lines for 7-Seg
    for(i=0;i<100000;i++);
```

```
}

void delay(void){
    unsigned int i;
    for(i=0;i<10000;i++);
}
```

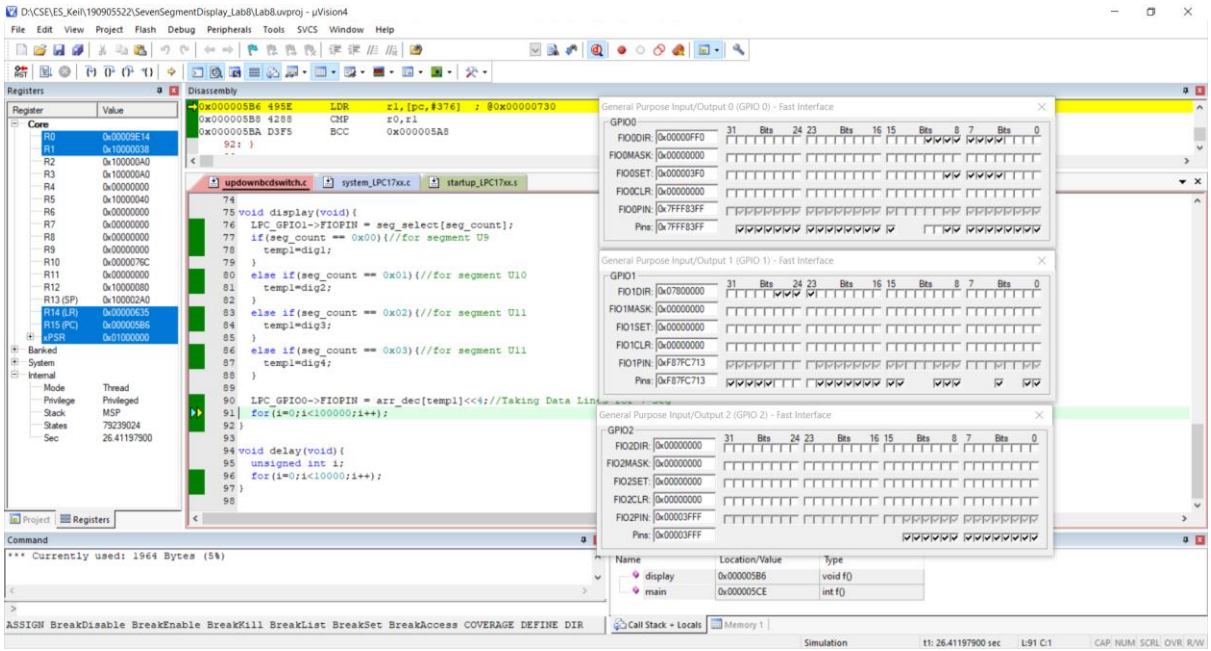
OUTPUT:

We read the value of the switch by reading the 12th bit of the GPIO2 pin which is turned on for the first iteration shown. Therefore, it acts as an up-counter.

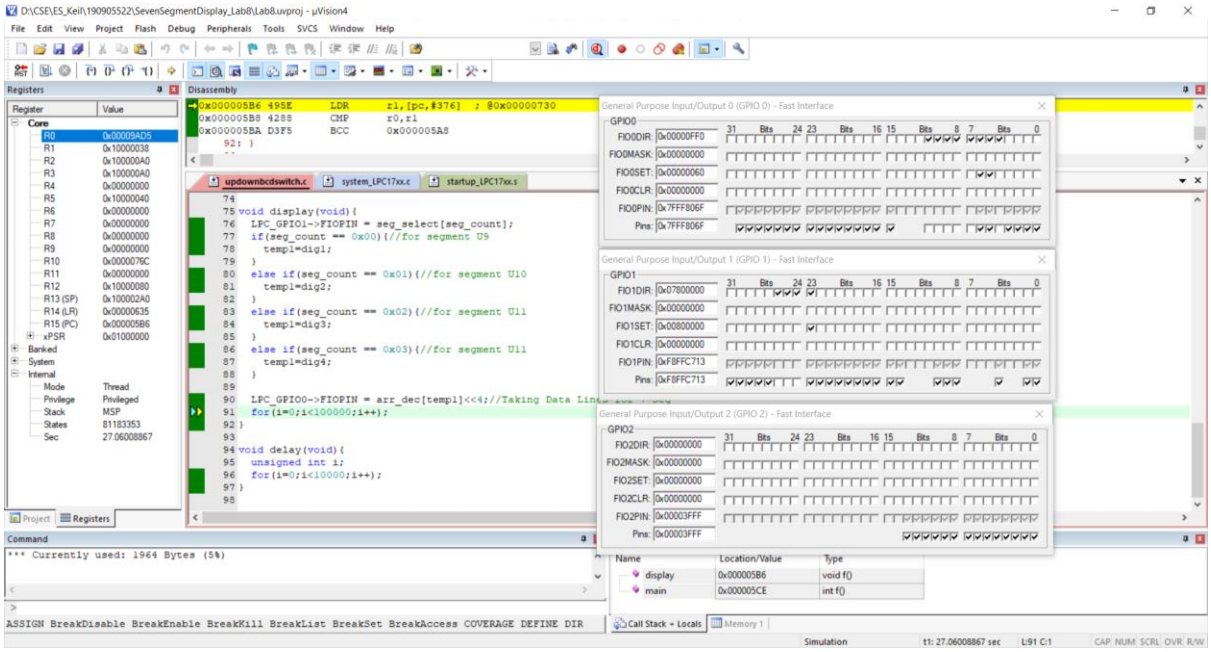
We show the up-counter iteration since the switch is on,

We assume the up-counter's previous iteration was at 0009. Therefore, the next value will be 0010.

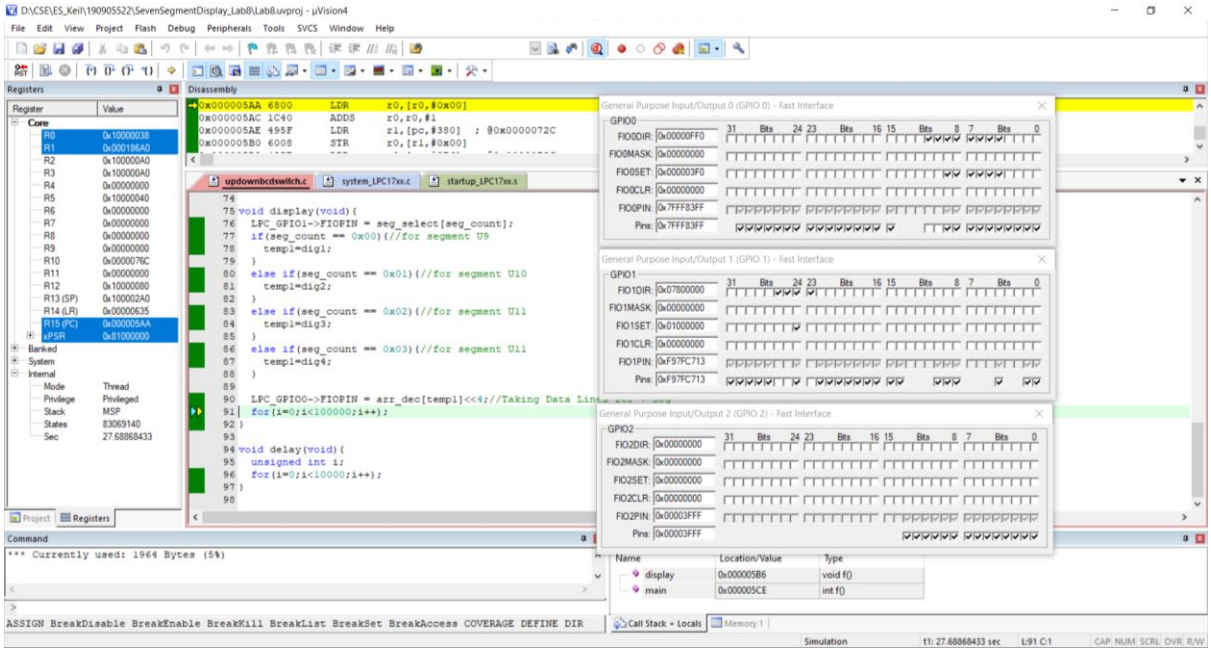
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



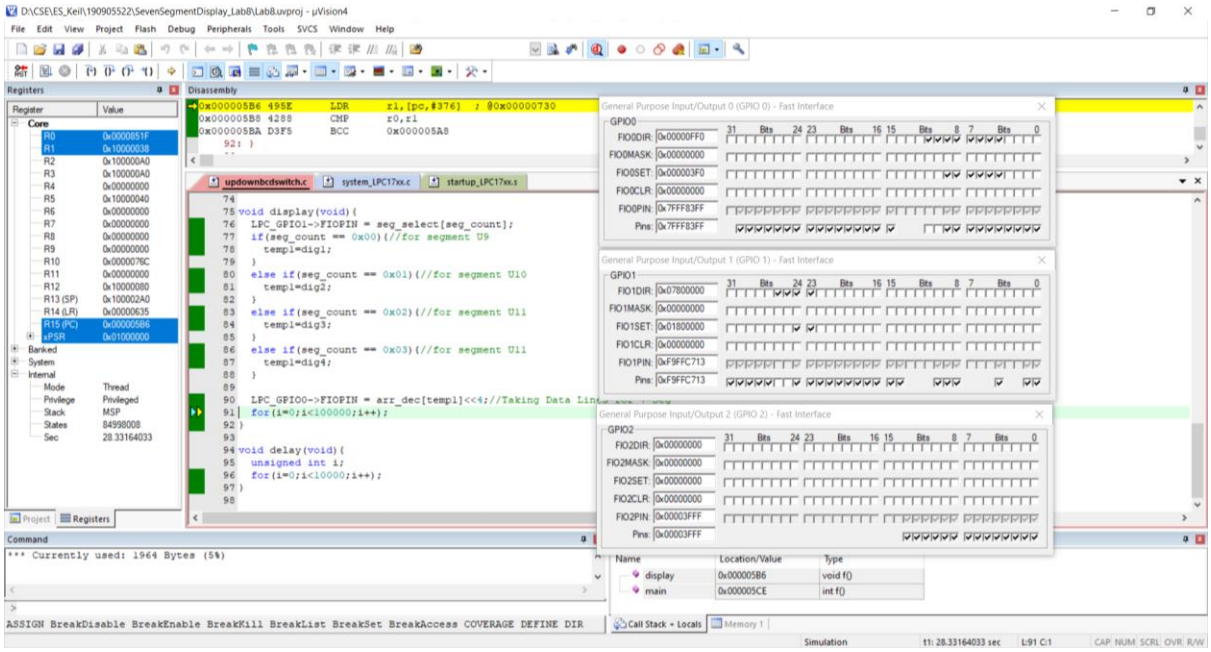
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit 1 in GPIO0 which is of value 0000 0110 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



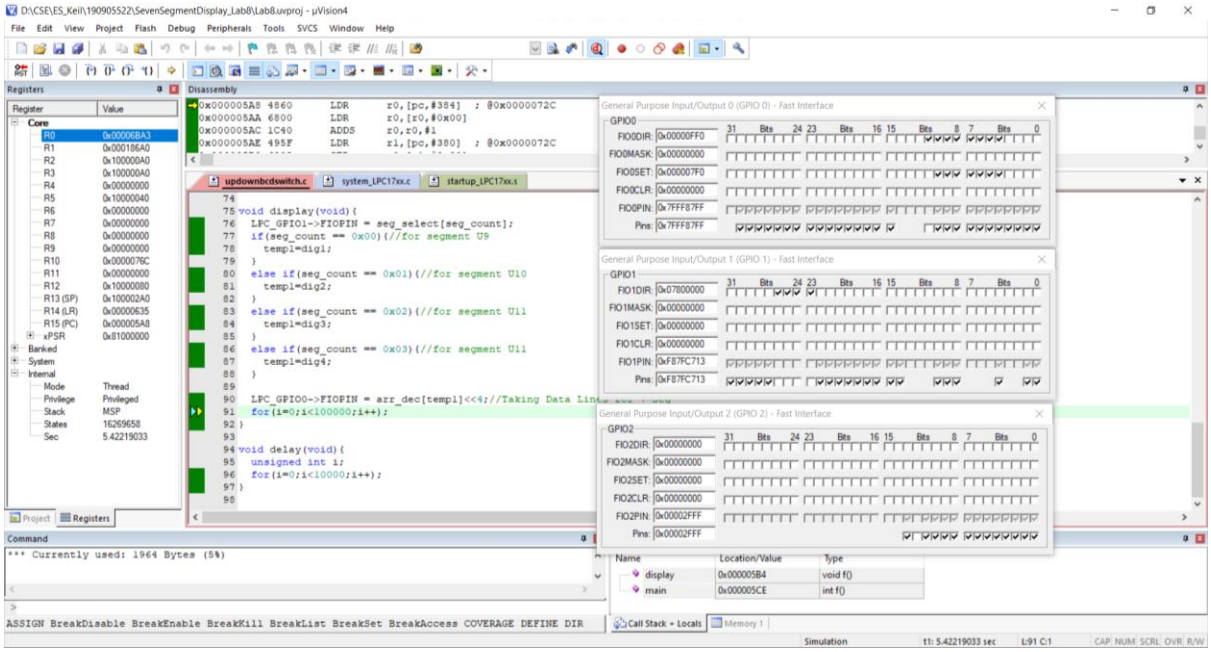
In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



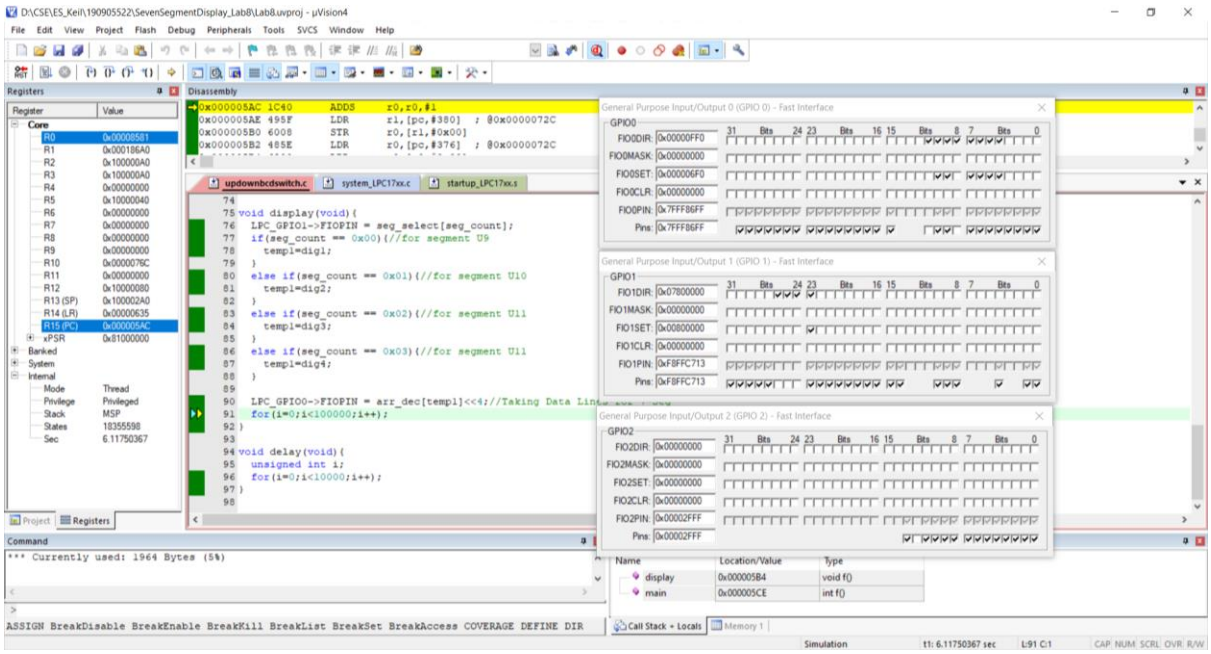
Now to see the down counter we can see that in GPIO2 the 12th bit is unselected representing that the switch is off:

We will show the functioning of the down counter in this iteration. The previous iteration had the value of 9999. Therefore, the next value will be 9998 which is shown as follows:

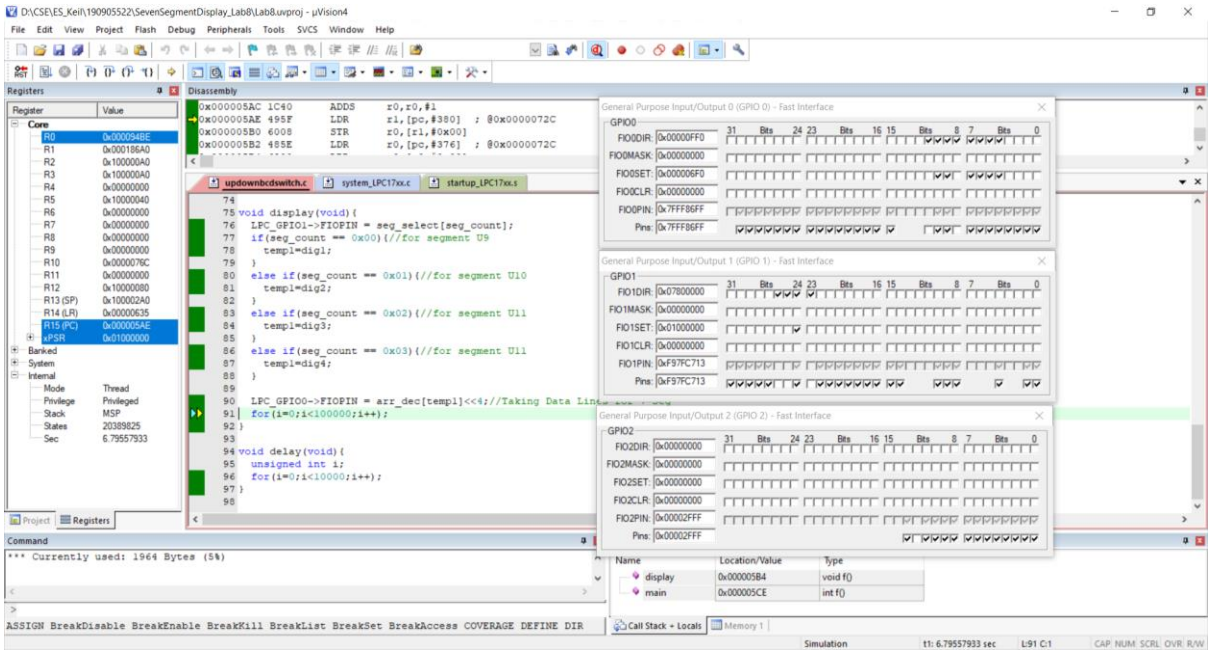
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit 8 in GPIO0 which is of value 0111 1111 in binary:



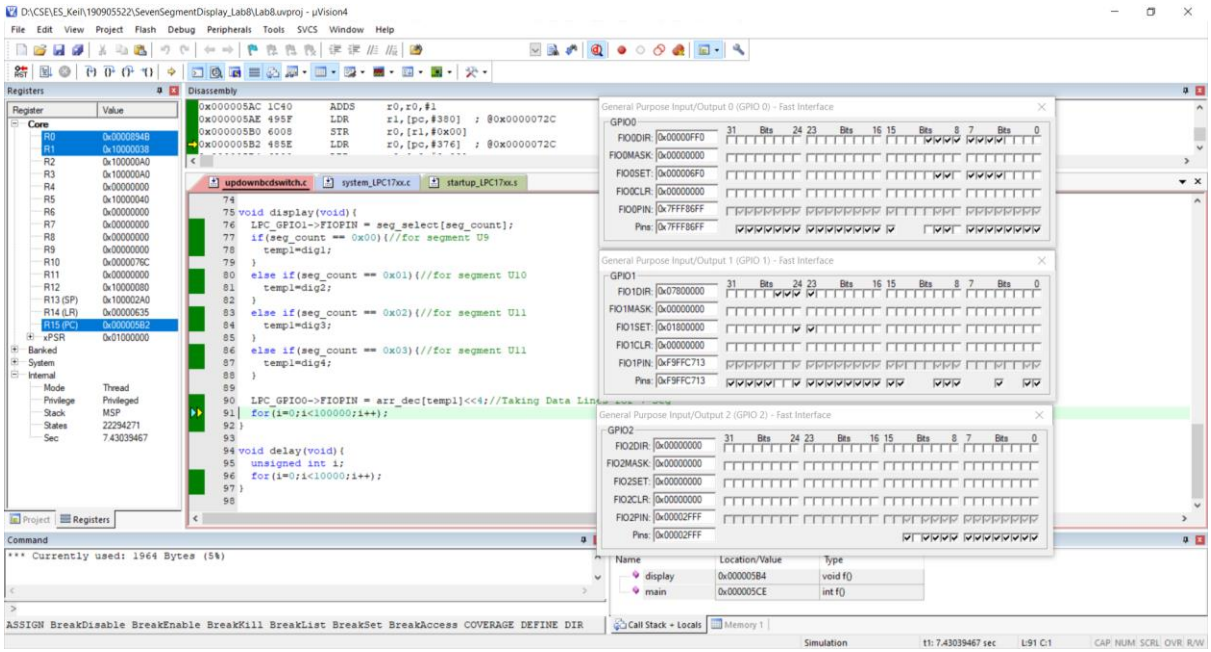
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit 9 in GPIO0 which is of value 0110 1111 in binary:



Thus, the number displayed is 9998, showing the functionality of a down counter.

4. Write a program for 4-digit Hexadecimal up/down counters on seven segment displays using a switch.

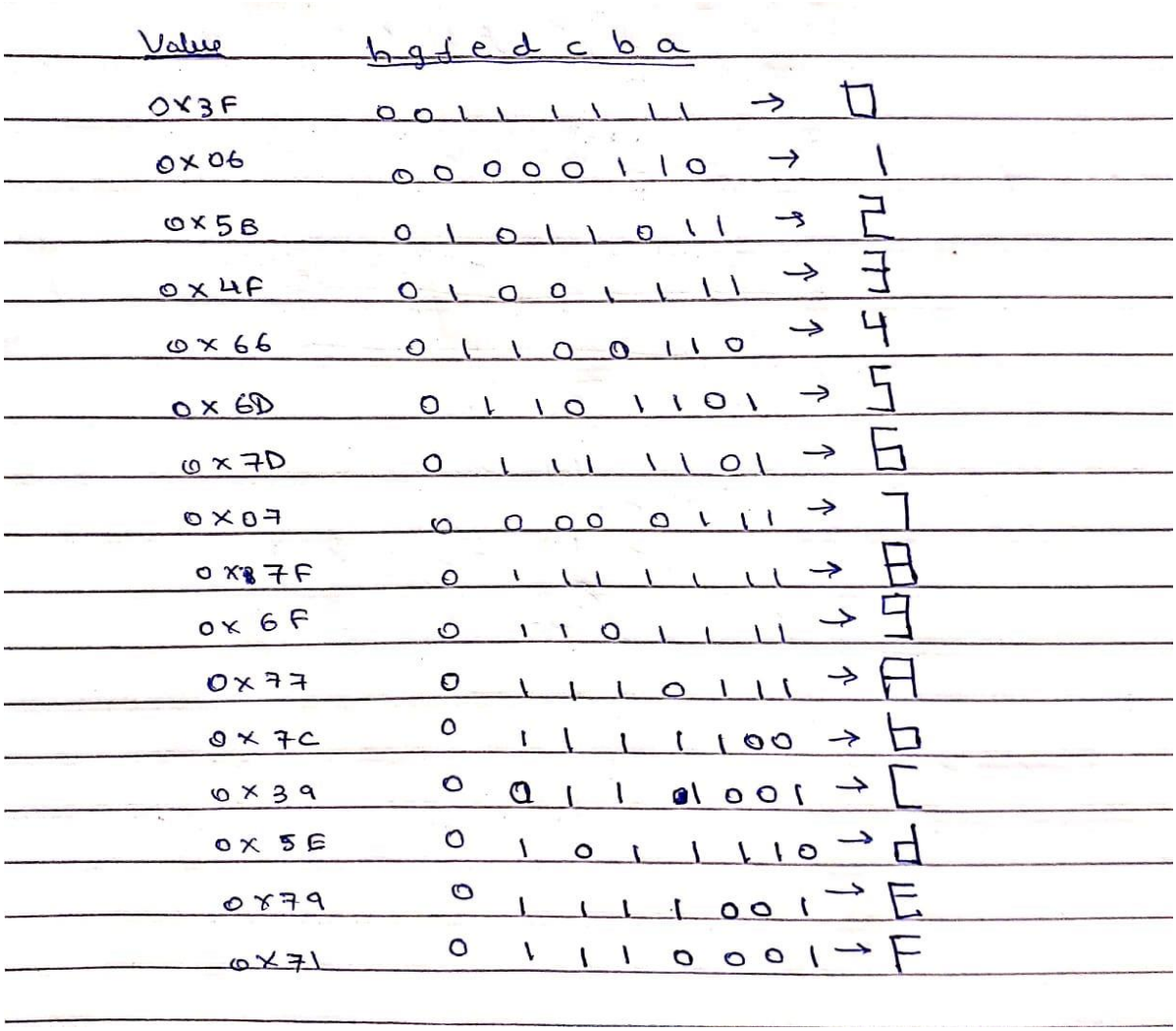
CODE:

```
#include<LPC17xx.h>
#include<stdio.h>

unsigned int seg_select[4] = {0<<23, 1<<23, 2<<23, 3<<23};
int dig1=0x00, dig2=0x00, dig3=0x00, dig4=0x00;
unsigned int seg_count=0x00, temp1=0x00;
unsigned char arr_dec[16]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0
x77,0x7C,0x39,0x5E,0x79,0x71};
unsigned long int i=0;
unsigned int k;
void delay(void);
void display(void);

int main(void){
```

```
    SystemInit();
    SystemCoreClockUpdate();

    LPC_PINCON->PINSEL0 &= 0xFF0000FF;//output
    LPC_PINCON->PINSEL3 &= 0xFFC03FFF;//bit
    LPC_PINCON->PINSEL4 &=0xFCFFFFFF;//switch

    LPC_GPIO0->FIODIR |= 0x00000FF0;//output
    LPC_GPIO1->FIODIR |= 0x07800000;//bit
    LPC_GPIO2->FIODIR &= 0xFFFFEFFF;//switch

    while(1){
        k = LPC_GPIO2->FIOPIN >> 12; //We read input from 2.12
        k &= 0x00000001;
        delay();
        display();
        seg_count +=1;
        if(seg_count == 0x04){
            seg_count = 0x00;

            if(k==1){
                dig1+=1;
                if(dig1 == 0x10){
                    dig1=0;
                    dig2+=1;
                    if(dig2 == 0x10){
                        dig2=0;
                        dig3+=1;
                        if(dig3 == 0x10){
                            dig3=0;
                            dig4+=1;
                            if(dig4 == 0x10){
                                dig4=0;
                            }//eod4
                        }//eod3
                    }//eod2
                }//eod1
            }
            else{
                dig1-=1;
                if(dig1 < 0){
                    dig1=0x0F;
                    dig2-=1;
                    if(dig2 < 0){
                        dig2=0x0F;
                        dig3-=1;
                        if(dig3 < 0){
                            dig3=0x0F;
```

```
                            dig4-=1;
                            if(dig4 < 0){
                                dig4=0x0F;
                            }//eod4
                        }//eod3
                    }//eod2
                }//eod1
            }//eoelse

        }//eosegcount
    }//eowhile
}//eomain

void display(void){
    LPC_GPIO1->FIOPIN = seg_select[seg_count];
    if(seg_count == 0x00){//for segment U9
        temp1=dig1;
    }
    else if(seg_count == 0x01){//for segment U10
        temp1=dig2;
    }
    else if(seg_count == 0x02){//for segment U11
        temp1=dig3;
    }
    else if(seg_count == 0x03){//for segment U11
        temp1=dig4;
    }

    LPC_GPIO0->FIOPIN = arr_dec[temp1]<<4;//Taking Data Lines for 7-Seg
    for(i=0;i<100000;i++);
}

void delay(void){
    unsigned int i;
    for(i=0;i<10000;i++);
}
```

Convention Used:



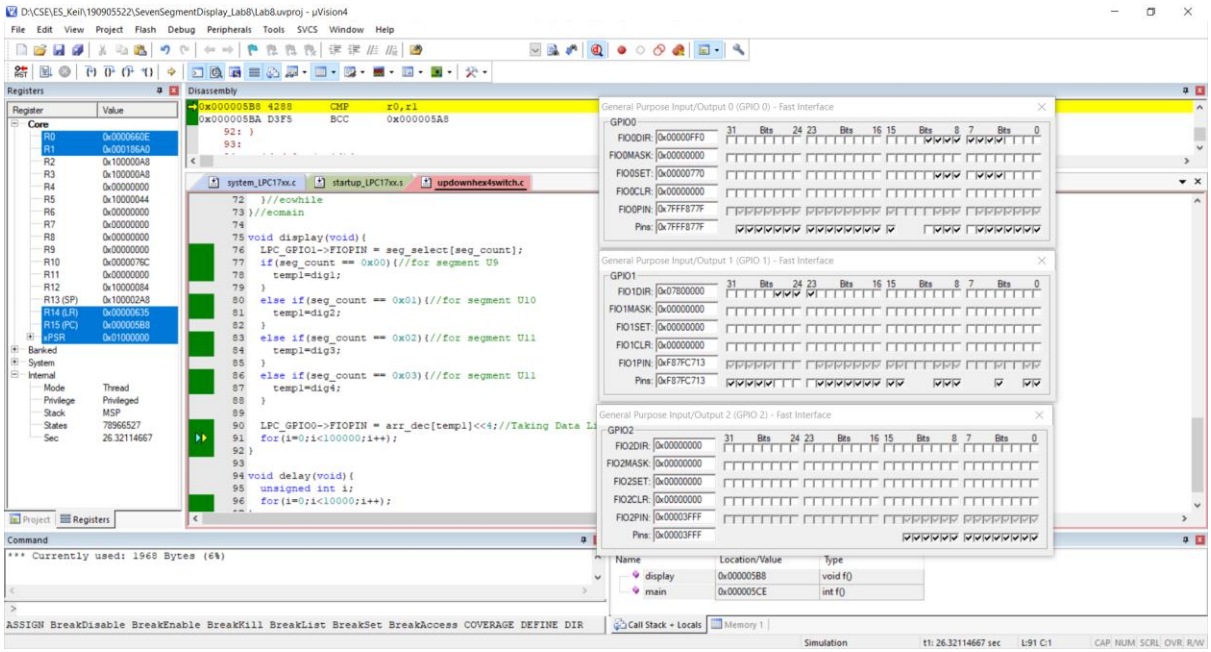
OUTPUT:

We read the value of the switch by reading the 12th bit of the GPIO2 pin which is turned on for the first iteration shown. Therefore, it acts as an up-counter.

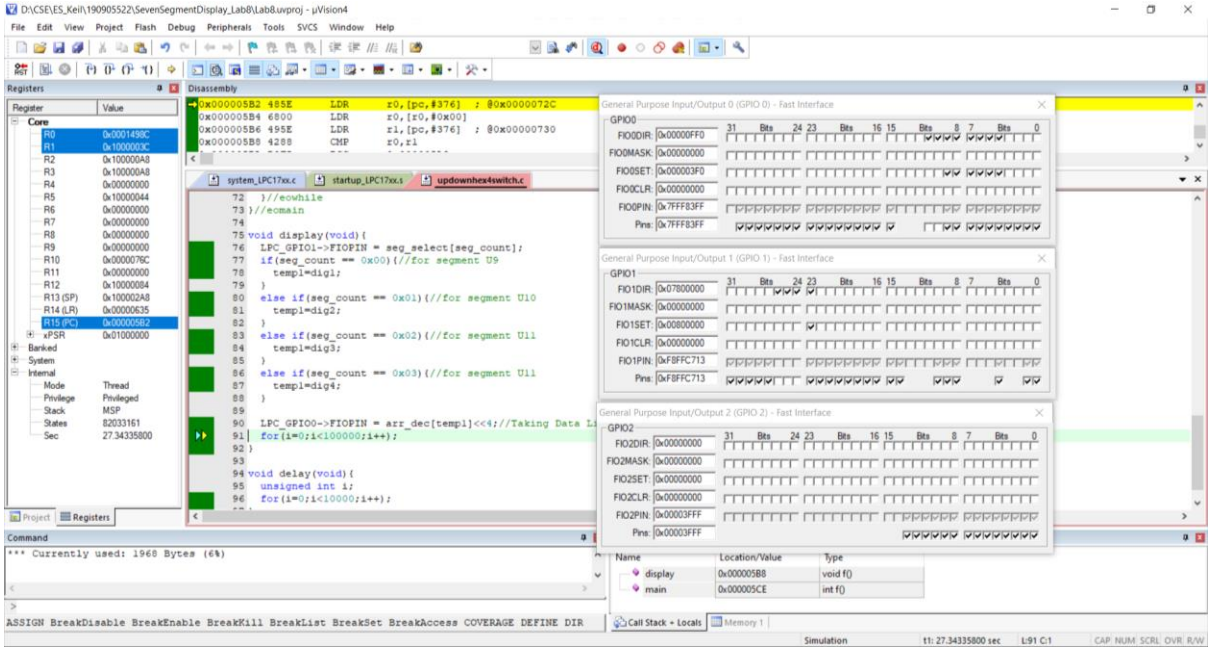
We show the up-counter iteration since the switch is on,

We assume the up-counter's previous iteration was at 0009. Therefore, the next value will be 000A.

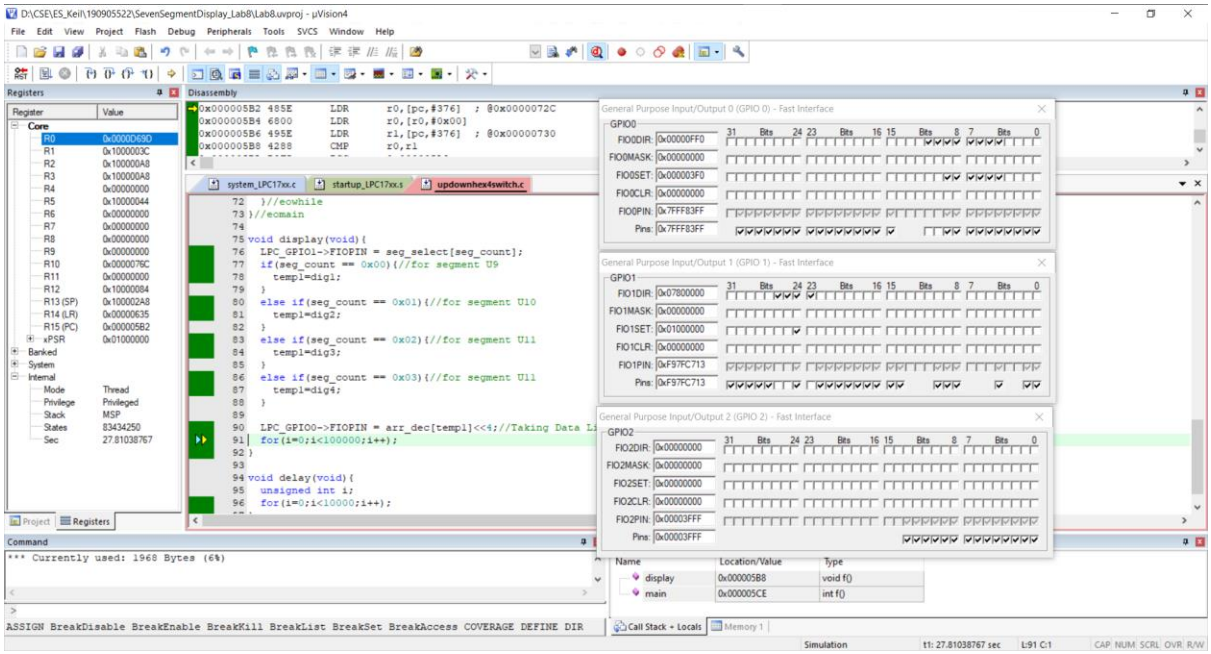
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit 'A' in GPIO0 which is of value 0111 0111 in binary:



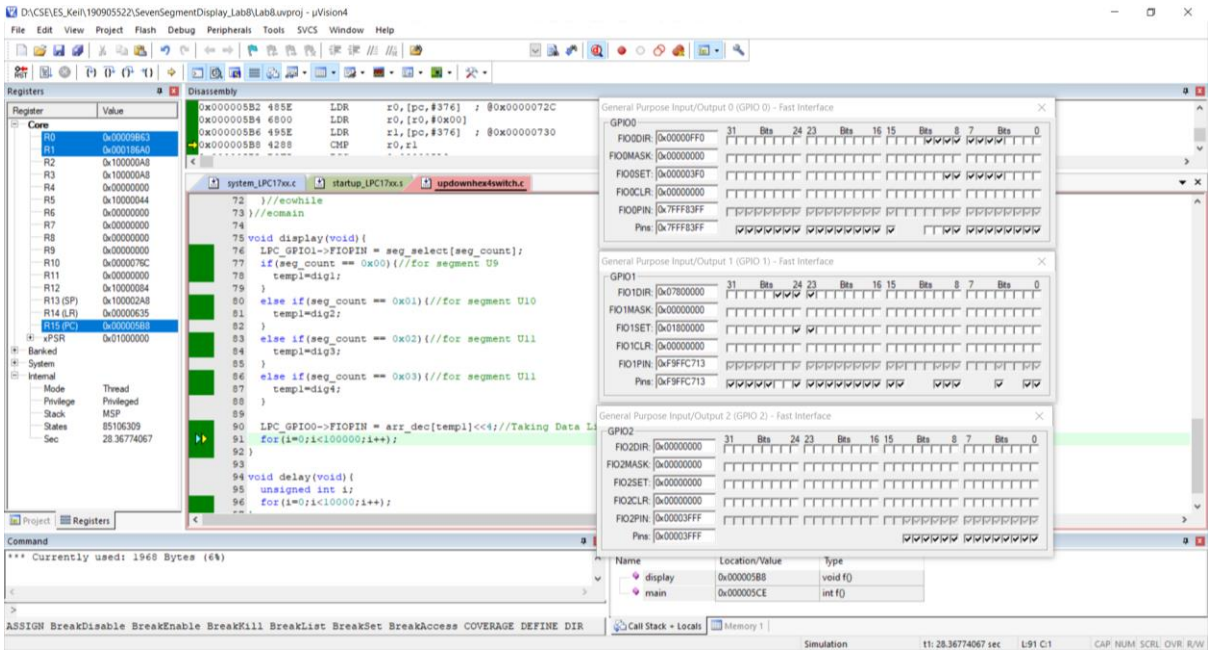
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



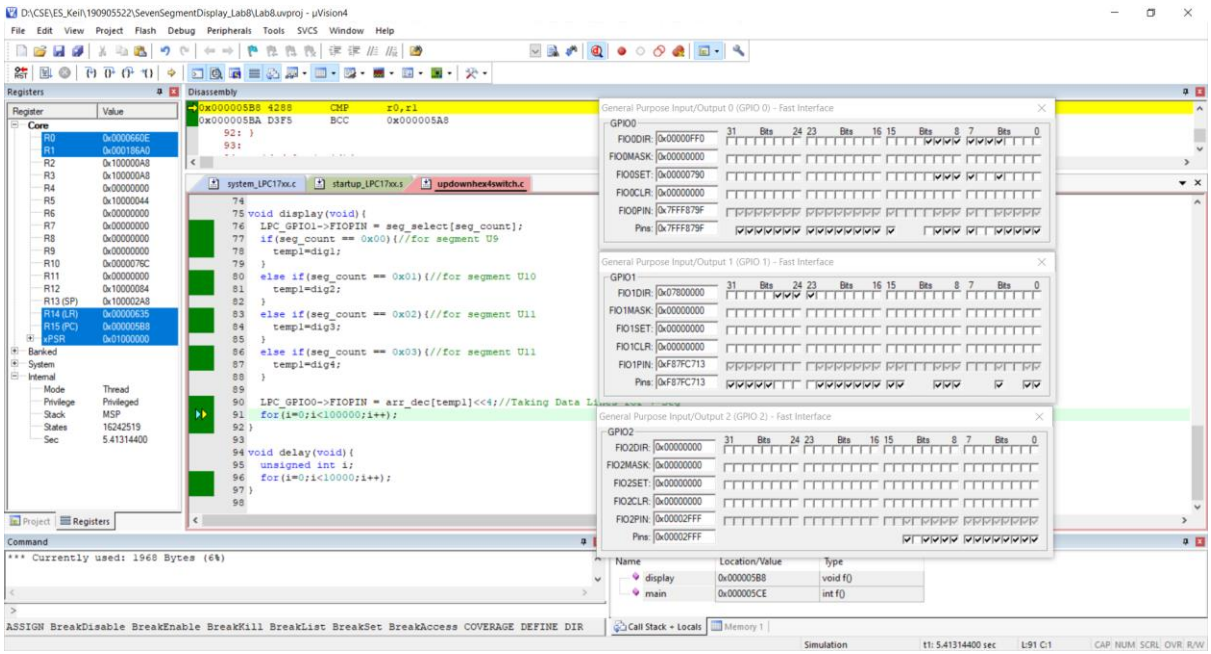
In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit 0 in GPIO0 which is of value 0011 1111 in binary:



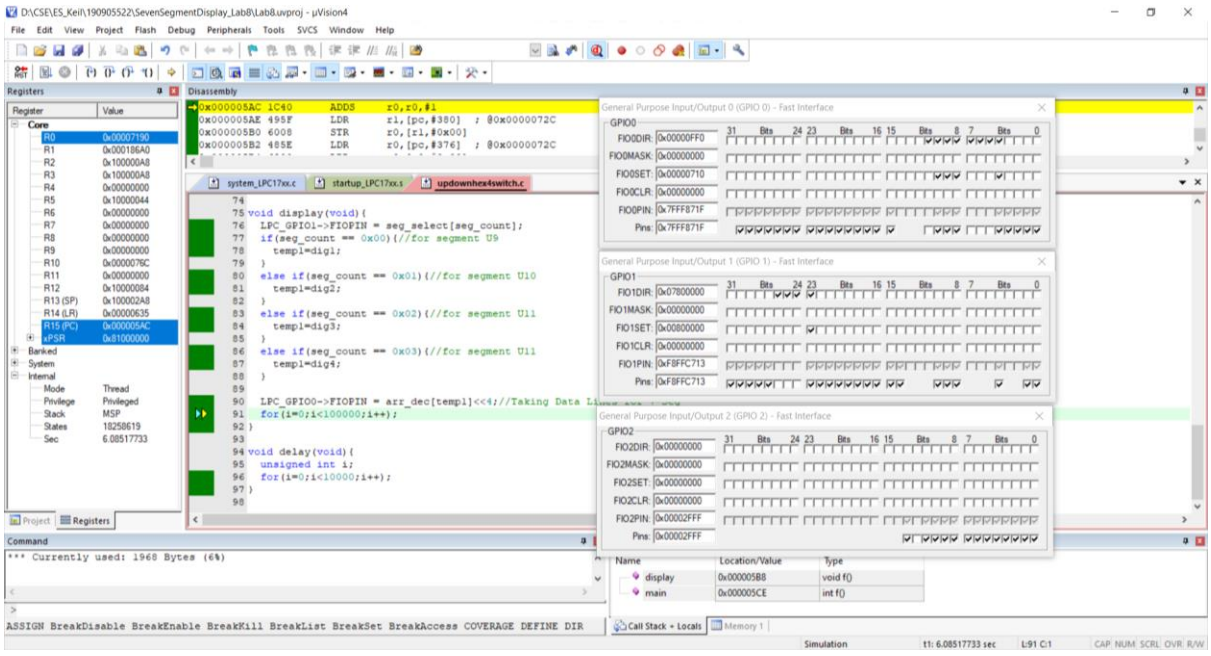
Now to see the down counter we can see that in GPIO2 the 12th bit is unselected representing that the switch is off:

We will show the functioning of the down counter in this iteration. The previous iteration had the value of FFFF. Therefore, the next value will be FFFE which is shown as follows:

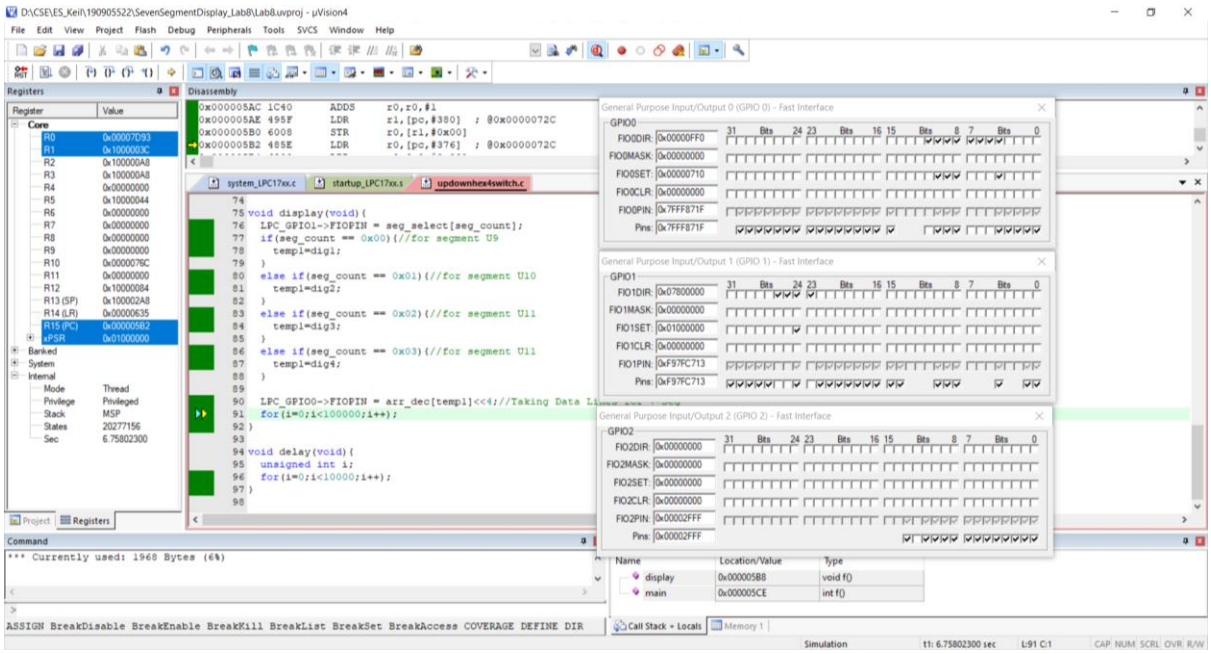
In the first position, the 0th bit shown in GPIO1 we can see output of seven segment display of the digit ‘E’ in GPIO0 which is of value 0111 1001 in binary:



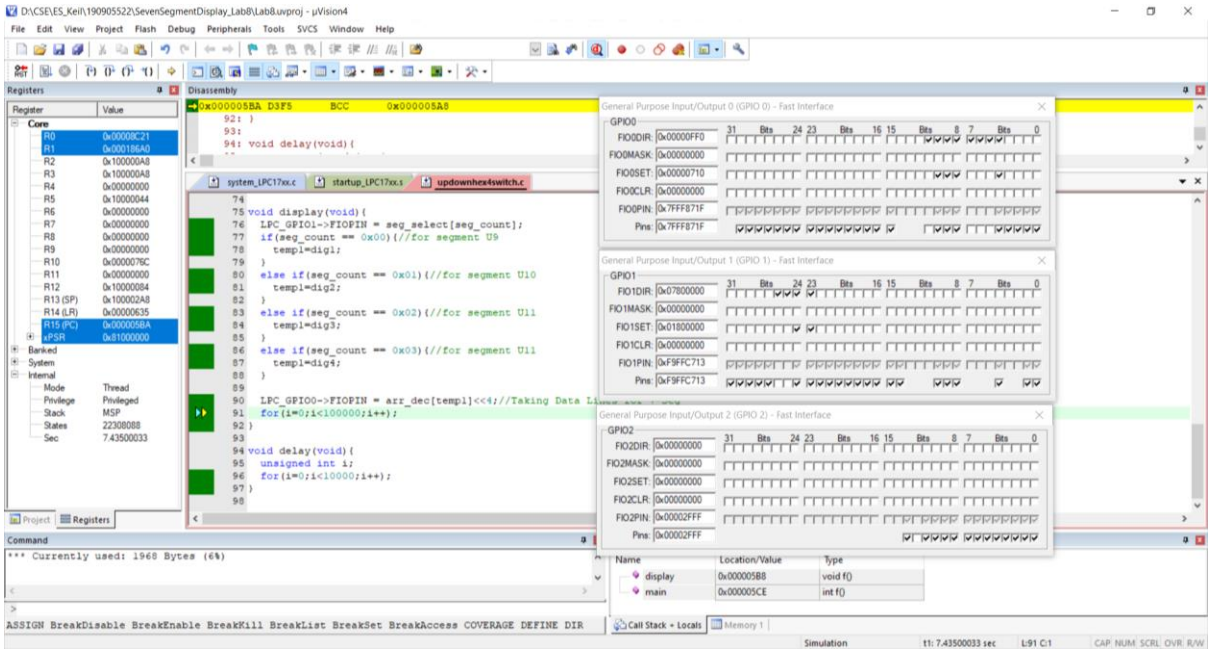
In the second position, the 1st bit shown in GPIO1 we can see output of seven segment display of the digit ‘F’ in GPIO0 which is of value 0111 0001 in binary:



In the third position, the 2nd bit shown in GPIO1 we can see output of seven segment display of the digit ‘F’ in GPIO0 which is of value 0111 0001 in binary:



In the fourth position, the 3rd bit shown in GPIO1 we can see output of seven segment display of the digit ‘F’ in GPIO0 which is of value 0111 0001 in binary:



THE END