Answer scheme for Mid-term test

SUB: Embedded Systems SEM: 5th IT

1.	Counter 1 is loaded with 7, input=100KHz, duty cycle 68% Since output is pinned on MAT1.2 Therefore MR register used here is MR2 of timer 1 MR2 is 7 it means outpur toggles for every 8 input count. Output frequency will have 16 input pulses. And output frequency is 100K/16=6250Hz. Irrespective of input duty cycle output of MAT is 50% duty cycle. Since counter at CAP pin counts only the pulses at rising/falling edge.	0.5+1+0.5
2.	Post index with example Pre-indexed with proper example	1+1
3.	Logic Copy 8 digit number from memory Load a counter for 8 digits Extract digit Add it to previous digit Adjust the sum to bcd Decrement the loop and repeat Store the result	3
4.	GPIO interrupt is asked Therefore enable P2.10 on rising edge and P2.11 on falling edge When the switch is pressed low signal will appear on port pin else high signal will appear on the port pin. (Push to OFF switch)	1 Marks for configuration 2 Marks for code in the interrupt

```
int main(void)
{
      LPC_GPIO0->FIODIR=0x1<<4;
        LPC_GPIOINT->IO2IntEnR=1<<10;
         LPC_GPIOINT->IO2IntEnF=1<<11;

      LPC_GPIOINT->IO2IntEnR=0x1<<10;
     LPC_GPIOINT->IO2IntEnF=0x1<<11;
      NVIC_EnableIRQ(EINT3_IRQn);

      while(1)   ;

}
void EINT3_IRQHandler(void)
 { unsigned long i,j;
       //i=;

       i=LPC_GPIOINT->IO2IntStatR;
       j=LPC_GPIOINT->IO2IntStatF;
       i=i&0x01<<10;
         i=i>>10;
       j=j&0x01<<11;
      j=j>>11;
       if (i==1)
       {
                  LPC_GPIO0->FIOCLR=1<<4
```

```
            LPC_GPIOINT->IO2IntClr |=0x01<<10; //cleares the
interrupt At P2.10
           }
      else if    (j==1)

      {
            LPC_GPIO0->FIOSET=1<<4;
      LPC_GPIOINT->IO2IntClr |=0x01<<11; //clears the interrupt at
P2.11
      }
}
```

5. Given For multiplexer
Select line P0.4 and P0.5, inputs P0.0-P0.3
And output at P0.6

```
Main()
{
   LPC_PINCON->PINSEL0&=0xFFFFF000;
   LPC_GPIO0->FIODIR=0x0000040;
   While(1)
   {
     (X=LPC_GPIO0->FIOPIN&0x03<<4)>>4;
     (Y=LPC_GPIO0->FIOPIN&1<<X)>>X;

       LPC_GPIO0->FIOPIN=y<<6;
}
```

0.5 Marks for configuration 1.5 Marks for logic+code

6. Given input connected at P2.12 seven segment connected at P0.0 to P0.7.

```
Unsigned char X,Y=0;
Main()
{ LPC_PINCON->PINSEL0&=FFFF0000;
  LPC_PINCON->PINSEL4|=1<<24;
Unsigned char a[10]={0x3f,0x06,……}; seven segment value
LPC_GPIO0->FIODIR=0xFF<<0;
LPC_SC->EXTMODE=0x01<<2; enable edge trigger
LPC_SC->EXTPOLAR=0x01<<2;
Timer_init(); enable the timer for 1 second
NVIC_EnableIRQ(EINT2_IRQn);
NVIC_EnableIRQ(TIMER0_IRQn);
While(1)
{LPC_GPIO0->FIOPIN=a[X]; display value of x on seven segment.
}
}
Timer_init()
{ LPC_TIM0->TCR=0x02;
          ->CTCR=0x0;
           ->MR0= for 1second
```

Initialization of ext.int+timer+GPIO= 1+1+0.5
Handler timer+ext+display= 1+0.5+0.5

```
            ->PR=0x02
          ->MCR=0x03; enable timer and reset upon match
          ->TCR=0x01; start the timer
}


void EINT2_IRQHandler(void)

{
   LPC_SC->EXTINT=1<<2; clear interrupt
   Y++;   increment count
 If(Y==10)
 Y=0;
}


void TIMER0_IRQHandler(void)
{
   LCP_TIM0->IR=0x01;
   X=Y; update X count to display after 1 second
}
```

7 Given input waveform connected at CAP0.0 means need to use timer0 as counter
Output MAT1.1 so Timer 1 as timer to generate waveform of width 10ms
Logic:
Use counter interrupt upon a positive edge at counter 0; and in handler start the timer1 for 10ms and stop it.

```
Main()
{
LPC_PINCON->PINSEL3=0xf<<18; enable CAP0.0 and MAT1.1 as given in
question
Timer_inint();
 While(1);

}
Timer_init()
{ LPC_TIM1->TCR=0x02;
          ->CTCR=0x0;
           ->MCR=0x0x20; upon match of MR1 stop (MAT1.1).
            ->MR1= ___for 10 ms
             ->PR=___
             ->EMR=0x40; upon match reset
  LPC_TIM0->TCR=0x02;
          ->CTCR=0x01; detect rising edge of CAP0.0
          ->CCR=0x05; generate interrupt on rising edge of CAP0.0
          ->TCR=0x01; start counter
NVIC_EnableIRQ(TIMER0_IRQn);
}
```

Config 1+counter part1+ timer part 1+ interrupt concept1

```
Void TIMER0_IRQHandlre(void)
{
  LPC_TIM0->IR=0x1<<4; clear interrupt on Capture channel
  LPC_TIM1->EMR=0x42; set EM1 bit high and reset upon match
  LPC_TIM1->TCR=0x01; start timer 1
  While((LPC_TIM1->EMR&0x01<<01)); wait for 10ms after 10ms EM1
bit resets to zero.
  }
```