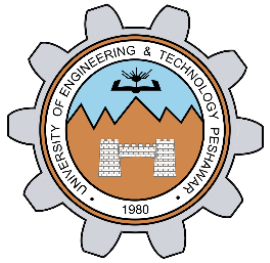
LAB 2:

Basic microcontroller (msp430) programming using C language:

Engr. Shahzada Fahim Jan



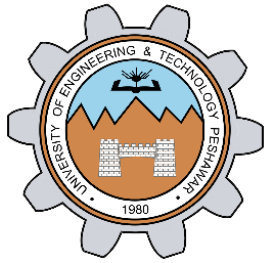
Memory-Mapped I\O

•It means that the ports simply appear to the CPU as particular memory registers called *peripheral registers*

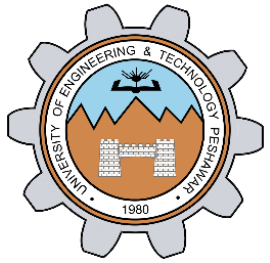
•**P1IN**: Read-Only register. Used to read the input values of P1 port, If it is configured as digital input.

•**P1OUT**: Writing sends the value to be driven onto the pin if it is configured as a digital output.

•**P1DIR**: A bit of 0 configures the pin as an input, which is the default. Writing a 1 switches the pin to become an output.



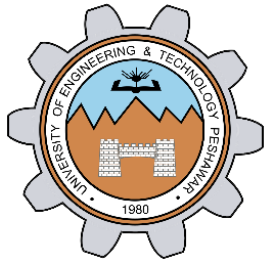

```
#include <msp430x11x1.h>      // Specific device
void main (void)
{
   WDTCTL = WDTPW | WDTHOLD;  // Stop WD timer
   P2DIR = 0x18;              // Sets pins with LEDs to output
   P2OUT = 0x08;              // LED2 (P2.4) on, LED1 (P2.3) off (active low!)
   for (;;) {                 // Loop forever
   }
}
```



Read/Write data from a particular memory location

C language

```
#include <msp430.h>
unsigned char result;

P1OUT = 0xAB;          /*write to the pointed location*/

result = P1OUT;        /* read from the pointed location */
```



Setting , Clearing and Toggling a bit

Setting a bit

C language:

```
#define  BIT7  0x80

unsigned char result;

result |= BIT7;
```

Clearing a bit

C language:

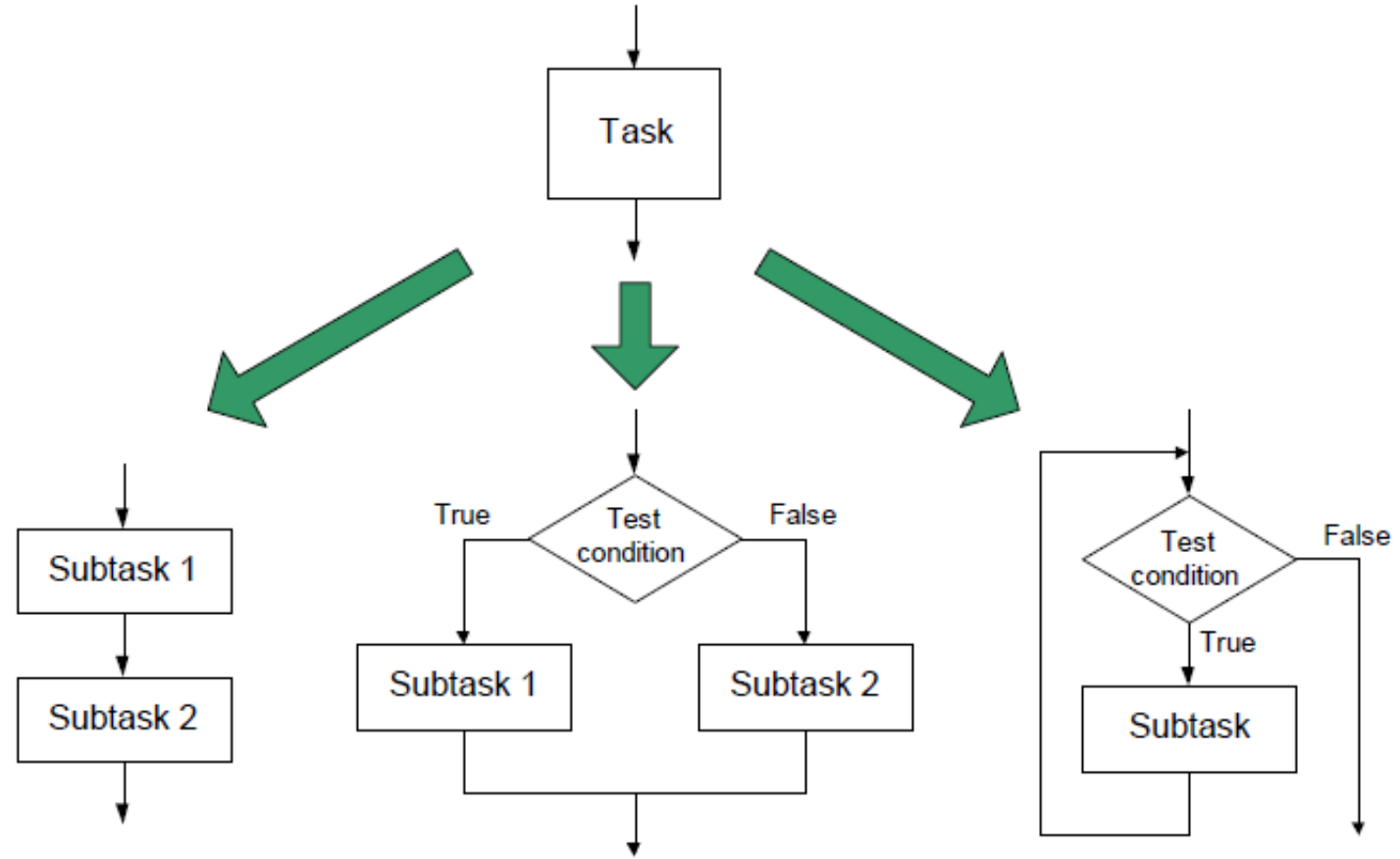
```
#define  BIT7  0x80

unsigned char result;

result &= ~BIT7;
```

Toggling a bit

C language:

```
#define  BIT7  0x80

unsigned char result;

result ^= BIT7;
```

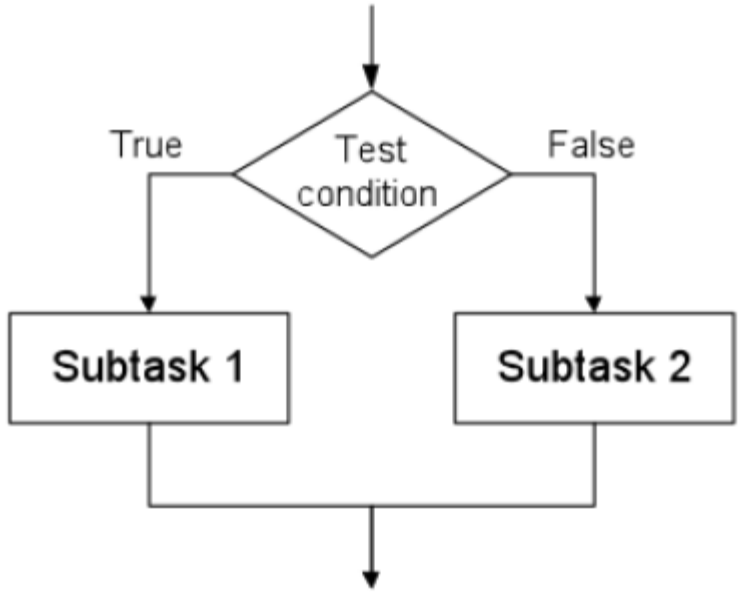
Three Basic Constructs



Sequential *Conditional* *Iterative*

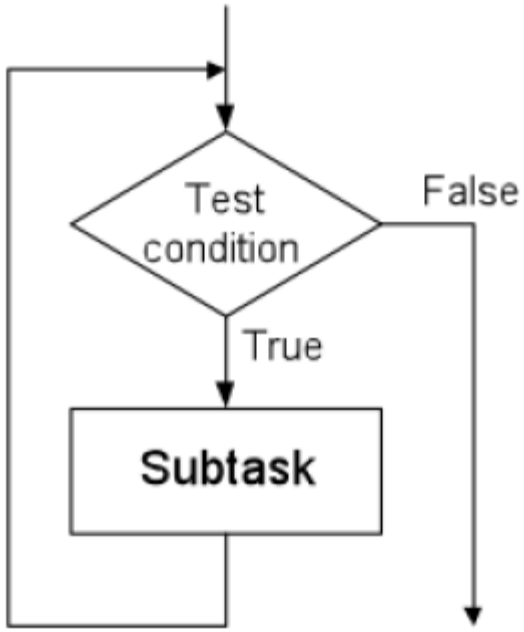
if-then-else Translation

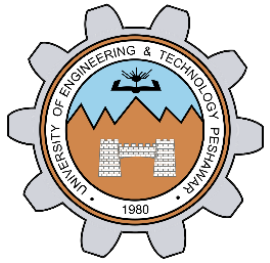
```
if (buzzerON == 1)
{
  pulse_buzzer();
  turn_on_LED();
}
else
{
  turn_off_LED();
}
```



while Translation

```
#define TRUE 1

while (TRUE)
{
   LED_ON();
   delay();
   LED_OFF();
   delay();
}
```





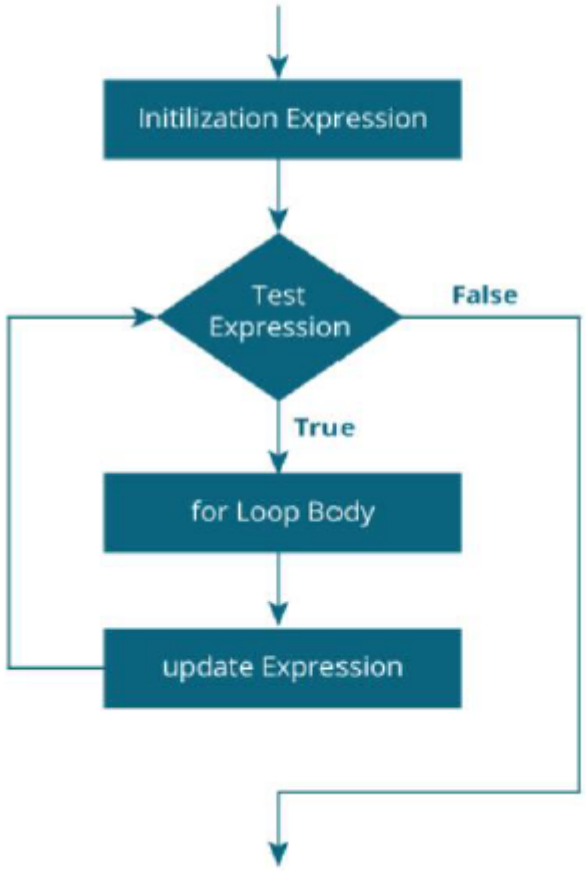
for-loop Translation

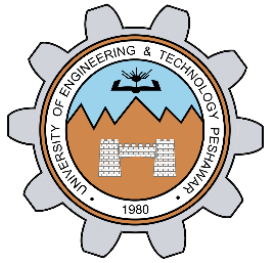
```
int i;

for(i=0; i<10; i++)
{

  do_dot();
  delay();
  do_dash();
  delay();

}
```

Draw the flow diagram

for-loop Translation

```
int i;

for(i=0; i<10; ++i)
{

  do_dot();
  delay();
  do_dash();
  delay();

}
```





switch/case Translation

- switch/case

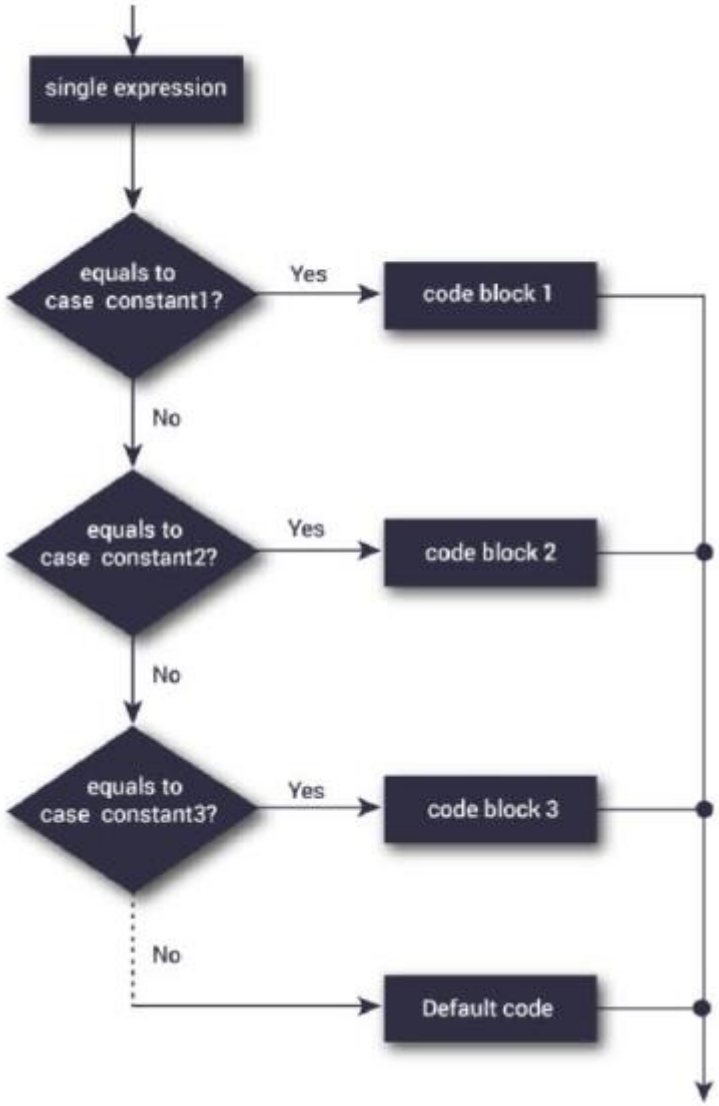
```
switch (myByte)
{
  case DOT:
    do_dot();
    break;

  case DASH:
    do_dash();
    break;

  default:
}
```

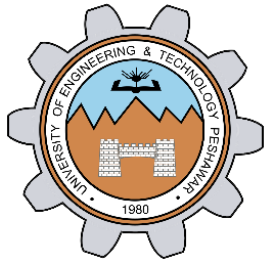
Draw the flow diagram

switch/case Translation

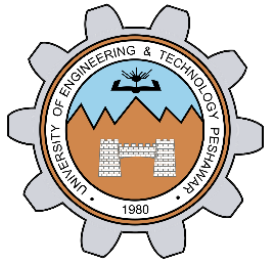
- switch/case

```
switch (myByte)
{
  case DOT:
    do_dot();
    break;

  case DASH:
    do_dash();
    break;

  default:
}
```






```c
#include <msp430fr4133.h>

int main(void) {
    WDTCTL = WDTPW | WDTHOLD;                   // Stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;

    P8DIR |= 0x0F;                              // Set P8.0 to P8.3 to output
direction

    for(;;) {
        volatile unsigned int i;                // volatile to prevent optimization

        P8OUT ^= 0x0F;

        i = 100000;
        do i--;
        while(i != 0);
    }

    return 0;
}
```




```c
#include <msp430fr4133.h>

int main(void) {
    WDTCTL = WDTPW | WDTHOLD;
    PM5CTL0 &= ~LOCKLPM5;

    P8DIR |= 0x0F;

    unsigned char pattern = 0x01;

    while (1) {
        P8OUT = pattern;

        __delay_cycles(1000000);
        pattern <<= 1;                          // Shift the bit left
        if (pattern == 0x10) {
            pattern = 0x01;
        }
    }
}
```

TASKS:

TASK 1

Write C program for Msp430 which toggle P1.0 or any other Pin of Msp430 MCU.

TASK2:

Write C program which toggle the LEDS attached with P1.0 and P1.7 at the same time with different delays.

TASK3:

Write C program which toggle all the LEDs attached with P1 or any other PORT

TASK4:

Display the pattern using C language

00000001

00000010

00000100

….

10000000

00000001

00000010

Continuously