

① Indirect Addressing

INDF and FSR Registers

The INDF register is not a physical register.

Addressing INDF actually addresses the register whose address is contained in FSR register.

Examples

- Register file 0x05 contains the value 0x10,

- " " 0x06 " " " 0x0A

- Load the value 0x05 into the FSR register

⇒ A read of the INDF register will return the value of 0x10.

- Increment the value of the FSR register by one (FSR = 0x06)

- A read of the INDF register now will return the value of 0x0A

The assembly code is as follows

movlw 0x10;

movwf 0x05,

movlw 0x0A;

movwf 0x06

movlw 0x05

movwf FSR;

movf INDF, W)

movwf PORTB → 0x10 is displayed at PORTB

lncf FSR, F;

movf INDF, W;

movwf PORTB → 0x0A
is displayed
at PORTB

2

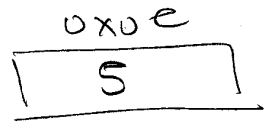
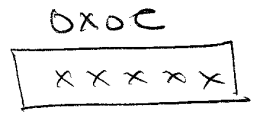
Exe

```

movlw 0x0C;
movwf FSR;

movlw 5;
movwf INDF

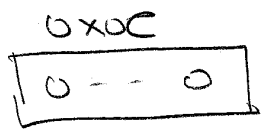
```



```

clrf INDF;

```



Ex3

How to clear RAM using indirect addressing

A simple program segment to clear RAM locations 20h - 2Fh

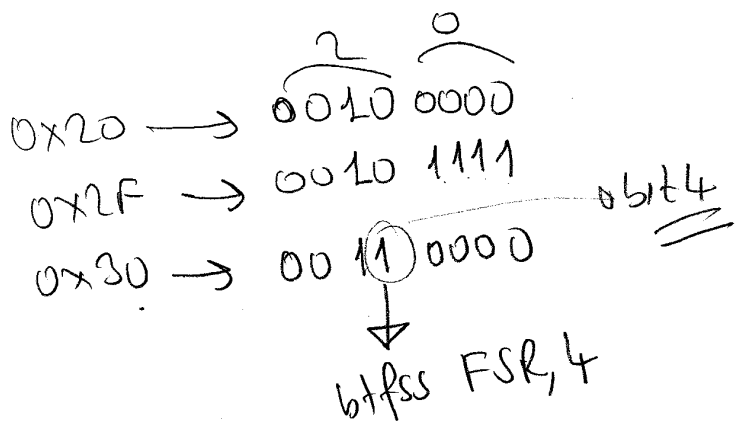
```

movlw 0x20; initialize pointer
movwf FSR; to RAM

NEXT clrf INDF; clear INDF register
      incf FSR; inc pointer
      btfss FSR,4, all done?
      goto NEXT;

```

CONTINUE !



3

Ex^o

get 20 inputs from PORTA
and fill the RAM locations starting
at 0x00 by the values from PORTA

Sn^o

Counter equ 0x0C

movlw .20,

movwf Counter,

movlw 0x00;

movwf FSR;

Loop movf PORTA, W;

movwf INDF;

incf FSR;

call Delay-1sec;

decfsz Counter, F;

goto Loop

Ex^o