Embedded Systems Design
(630470)

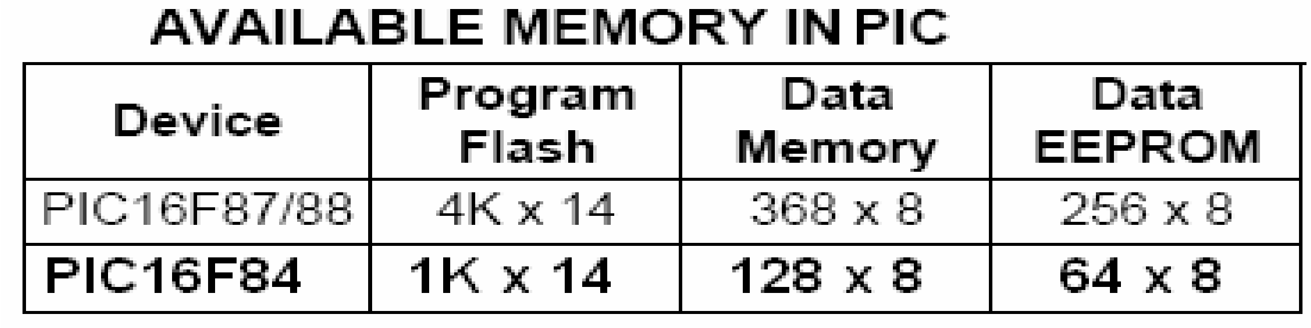
Lecture 7

Memory Organization

Prof. Kasim M. Al-Aubidy

Computer Eng. Dept.

Memory Organization:



AVAILABLE MEMORY IN PIC

Device	Program Flash	Data Memory	Data EEPROM
PIC16F87/88	4K x 14	368 x 8	256 x 8
PIC16F84	**1K x 14**	**128 x 8**	**64 x 8**

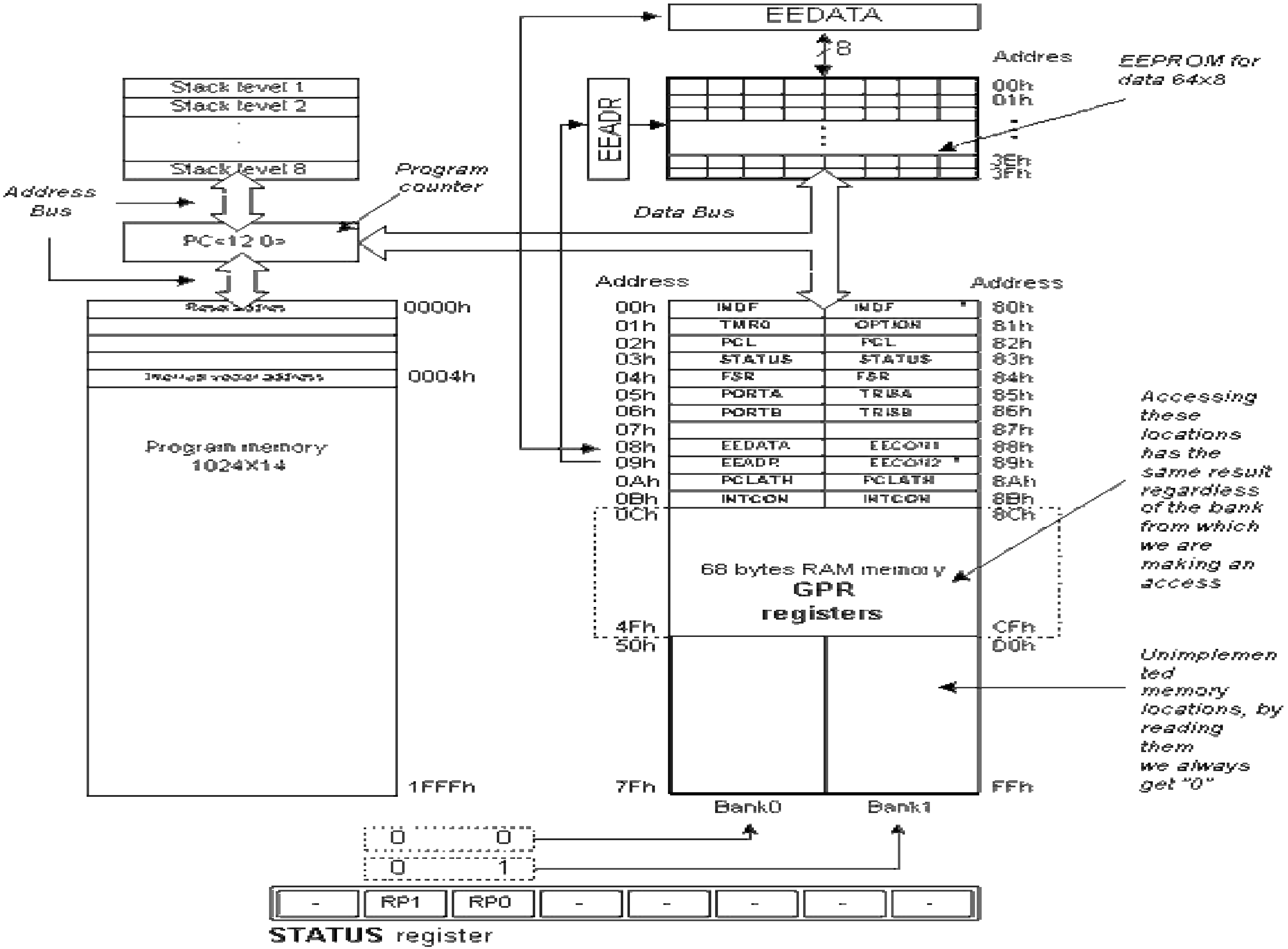
- PIC16F84 has two separate memory blocks, for data and for program.
- EEPROM memory with GPR and SFR registers in RAM memory make up the data block, while FLASH memory makes up the program block.

Program Memory

- Program memory has been carried out in FLASH technology.
- The size of program memory is 1024 locations with 14 bits width where locations zero and four are reserved for reset and interrupt vector.

Data Memory

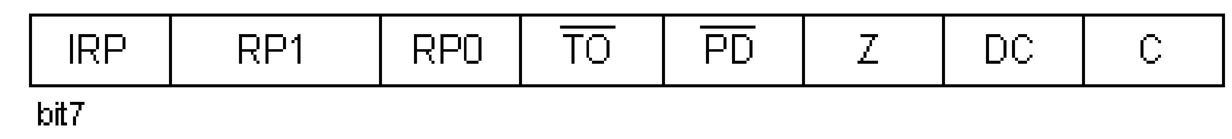
- Data memory consists of EEPROM and RAM memories.
- EEPROM memory consists of 64 eight bit locations.
- EEPROM is not directly addressable, but is accessed indirectly through EEADR and EEDATA registers.
- EEPROM memory usually serves for storing important parameters.
- RAM memory for data occupies space on a memory map from location 0x0C to 0x4F which comes to 68 locations (**GPR registers**).
- **SFR registers** take up first 12 locations in banks 0 and 1.



Memory organization of microcontroller PIC16F84

Memory Banks

- Memory map is divided in 'width' to two areas called 'banks'.
- Selecting one of the banks is done via RP0 bit in STATUS register.



IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit7							

Example:

```
bcf STATUS, RP0
```

Instruction BCF clears bit RP0 in STATUS register and thus sets up bank 0.

```
bsf STATUS, RP0
```

Instruction BSF sets the bit RP0 in STATUS register and thus sets up bank1.

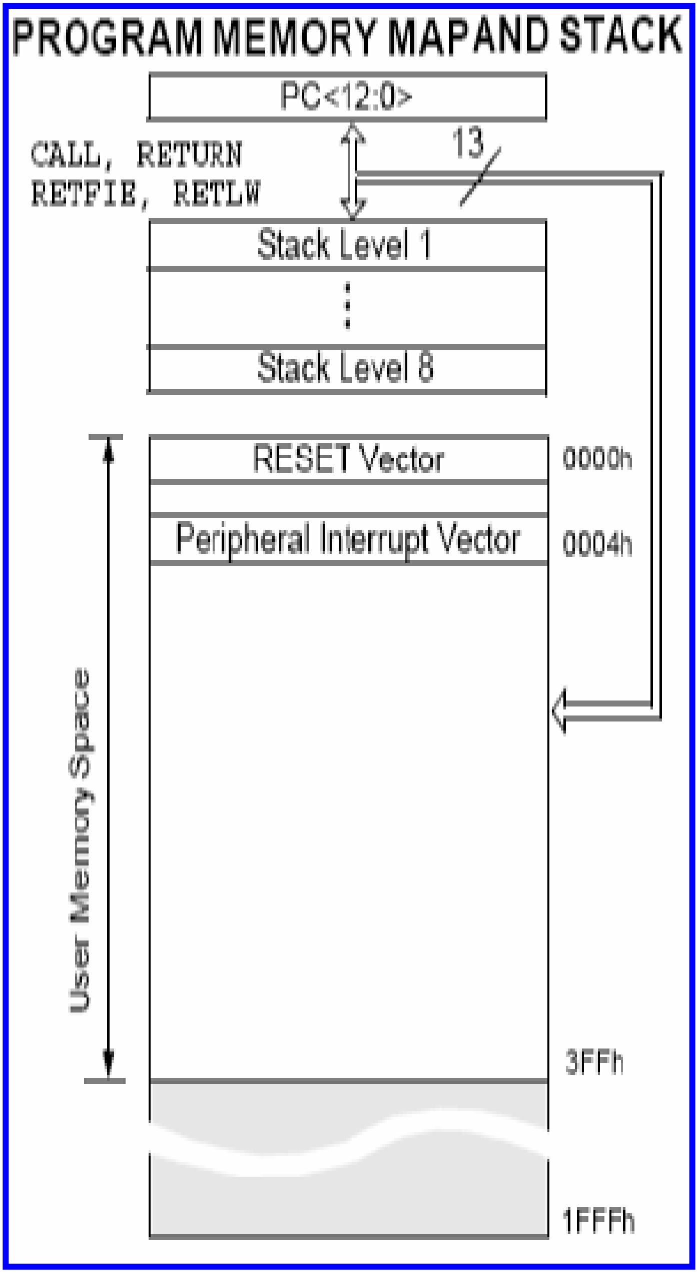
What would happen if the wrong bank was selected?

```
BANK0 macro
    bcf STATUS, RP0   ;Select memory bank 0
    endm

BANK1 macro
    bsf STATUS, RP0   ;Select memory bank 1
    endm
```

Stack:

- PIC16F84 has a 13-bit stack with 8 levels.
- Its basic role is to keep the value of PC after a jump from the main program to an address of a subprogram .
- In order for a program to know how to go back to the point where it started from, it has to return the value of a PC from a stack.
- When moving from a program to a subprogram, PC is being pushed onto a stack. When executing instructions such as RETURN, RETLW or RETFIE which were executed at the end of a subprogram, PC was taken from a stack so that program could continue where was stopped before it was interrupted. These operations of placing on and taking off from a program counter stack are called PUSH and POP.

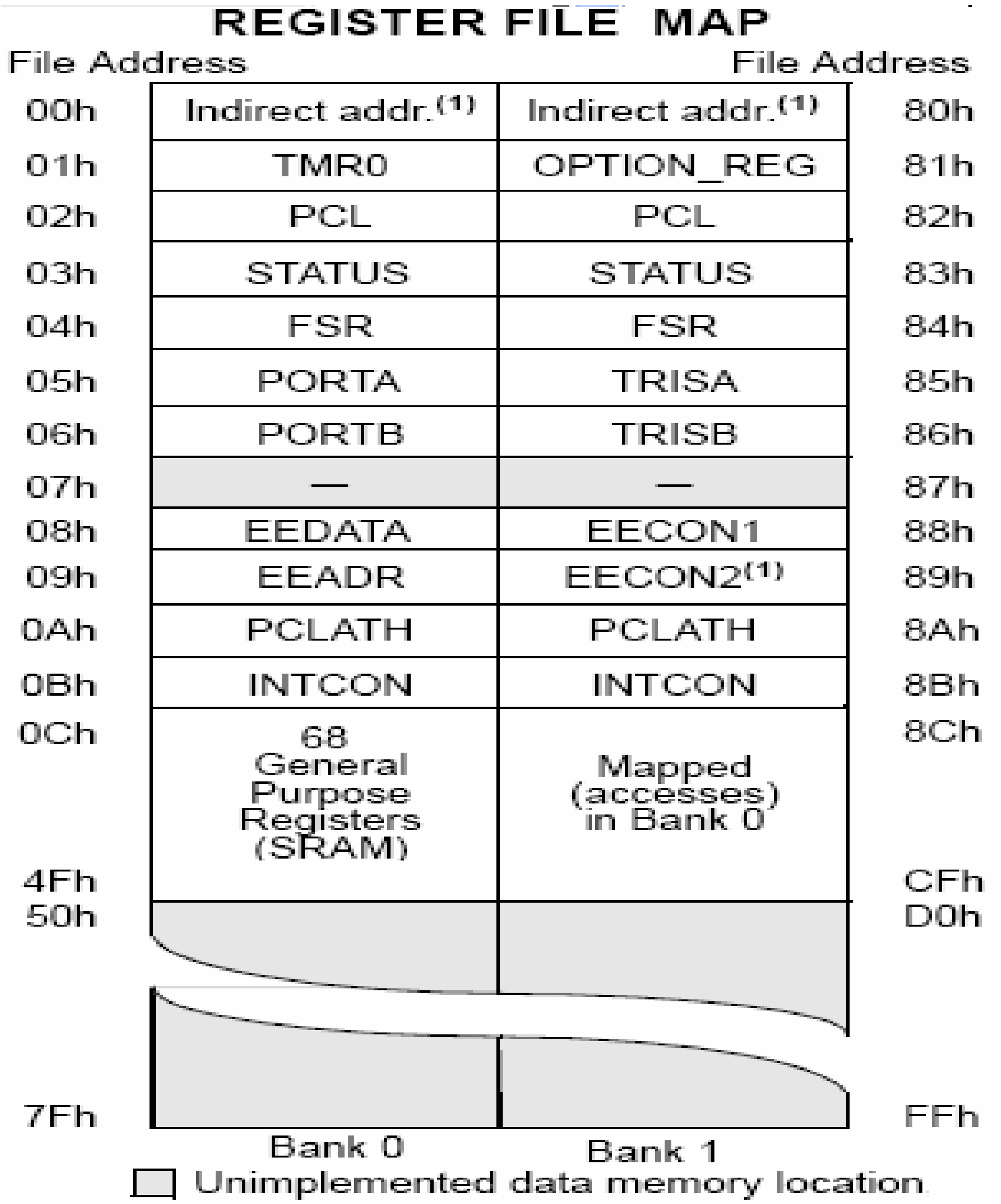


Data Memory:

Data memory is partitioned into two areas:
- SFR Area.
- GPR Area.

The GPR area allow greater than 116 bytes of GP RAM.

The data memory can be accessed either directly using the absolute address of each register file or indirectly through the file select register (FSR).



REGISTER FILE MAP

File Address			File Address
00h	Indirect addr.[1]	Indirect addr.[1]	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2[1]	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch 4Fh	68 General Purpose Registers (SRAM)	Mapped (accesses) in Bank 0	8Ch CFh
50h 7Fh	▨	▨	D0h FFh
	Bank 0	Bank 1	

▨ Unimplemented data memory location

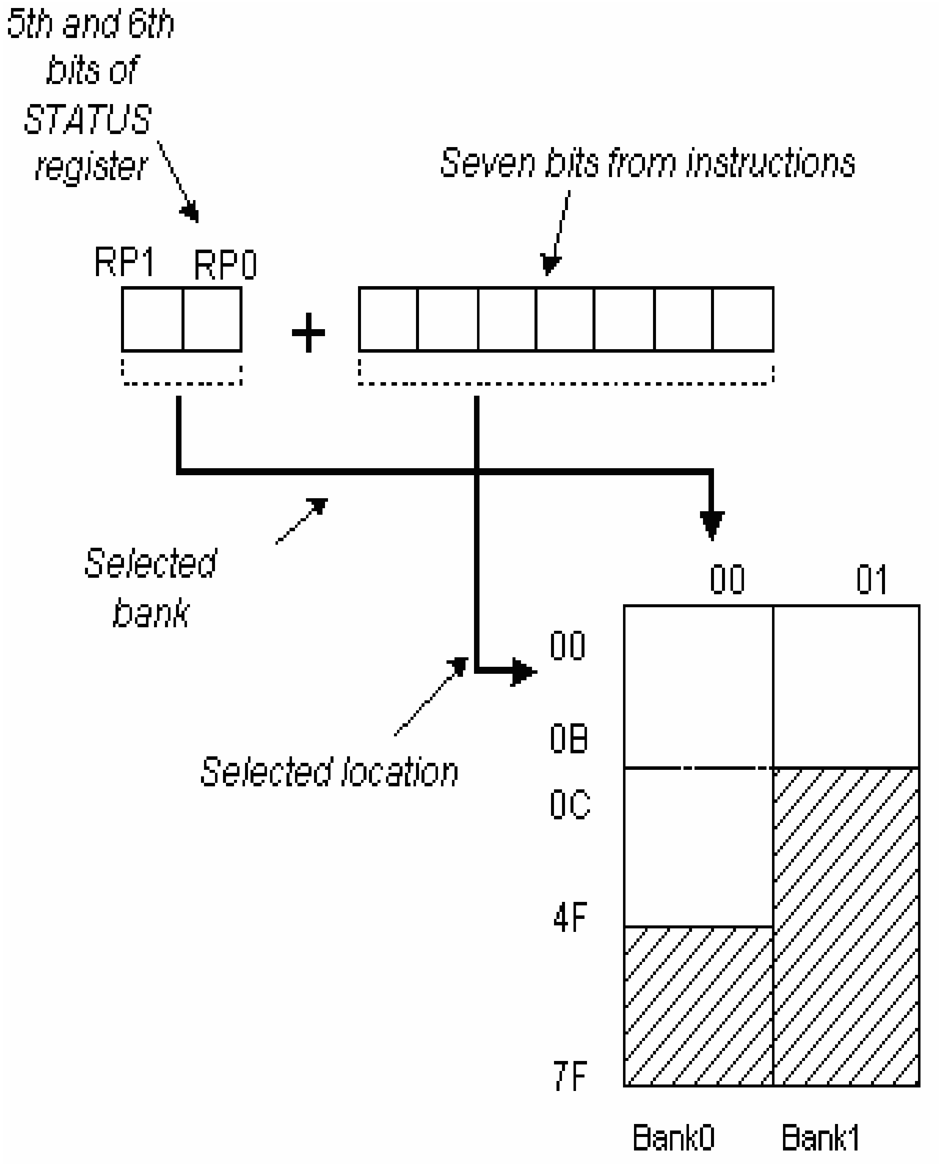
Addressing Modes:

- RAM memory locations can be accessed directly or indirectly.

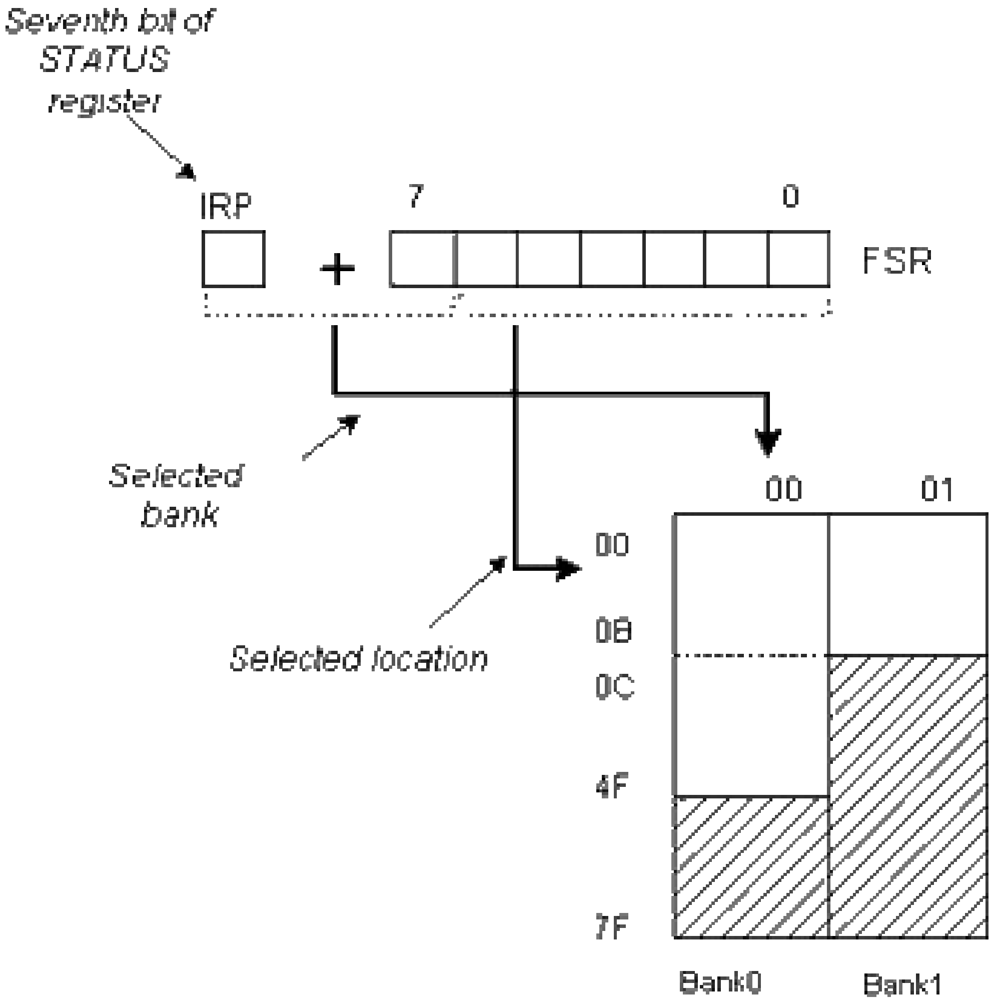
Direct Addressing:

- Direct Addressing is done through a 9-bit address.
- Example:

```
Bsf STATUS, RP0      ;Bankl
movlw 0xFF           ;w=0xFF
movwf TRISA          ;address of
TRISA register is taken from
instruction movwf
```


5th and 6th bits of STATUS register
Seven bits from instructions
RP1
RP0
+
Selected bank
Selected location
00
01
00
0B
0C
4F
7F
Bank0
Bank1

- **Indirect Addressing:**
- Indirect unlike direct addressing does not take an address from an instruction but derives it from IRP bit of STATUS and FSR registers.
- Addressed location is accessed via INDF register which in fact holds the address indicated by a FSR.



Indirect addressing