Architecture & Pipelining (PIC 16F877A)

What is PIC?

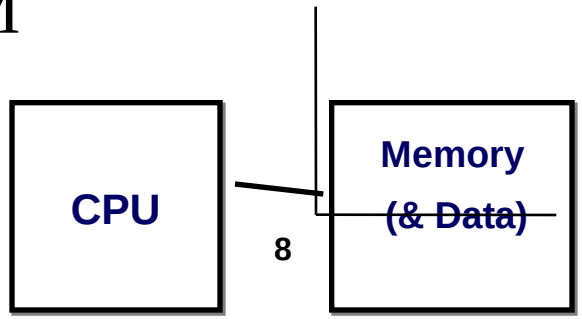
- The term PIC or Peripheral Interface Controller, has been coined by Microchip Technology Inc.

- Low-end range, mid-range and high end range of controllers.

PIC Architecture: Background

We're used to the Von-Neuman Architecture

- Used in: 80X86 (PCs), 8051, 68HC11, etc.)
- Only one bus between CPU and memory
- RAM and program memory share the same bus and the same memory, and so must have the same bit width
- Bottleneck: Getting instructions interferes with accessing RAM

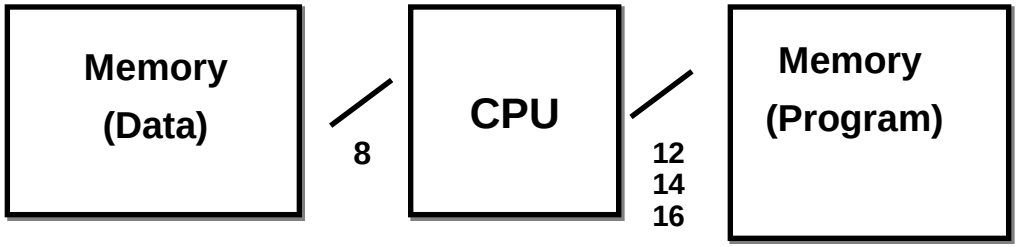


PIC Architecture: Background

PICs use the Harvard Architecture

Used mostly in RISC CPUs (we'll get there)

- Separate program bus and data bus: can be different widths!
- For example, PICs use:
  - Data memory (RAM): a small number of 8bit registers
  - Program memory (ROM): 12bit, 14bit or 16bit wide (in EPROM, FLASH, or ROM)

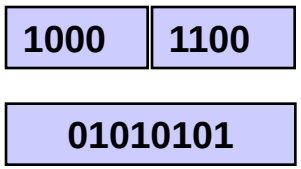


CISC

Traditionally, CPUs are “CISC”

- Complex Instruction Set Computer (CISC)
- Used in: 80X86, 8051, 68HC11, etc.
- Many instructions (usually > 100)
- Many, many addressing modes
- Usually takes more than 1 internal clock cycle (T cycle) to execute
- Example:

MC68HC05:



RISC

PICs and most Harvard chips are “RISC”

- Reduced Instruction Set Computer (RISC)
- Used in: SPARC, ALPHA, Atmel AVR, etc.
- Few instructions (usually < 50)
- Only a few addressing modes
- Executes 1 instruction in 1 internal clock cycle (Tcyc)
- Example:

PIC16CXXX:

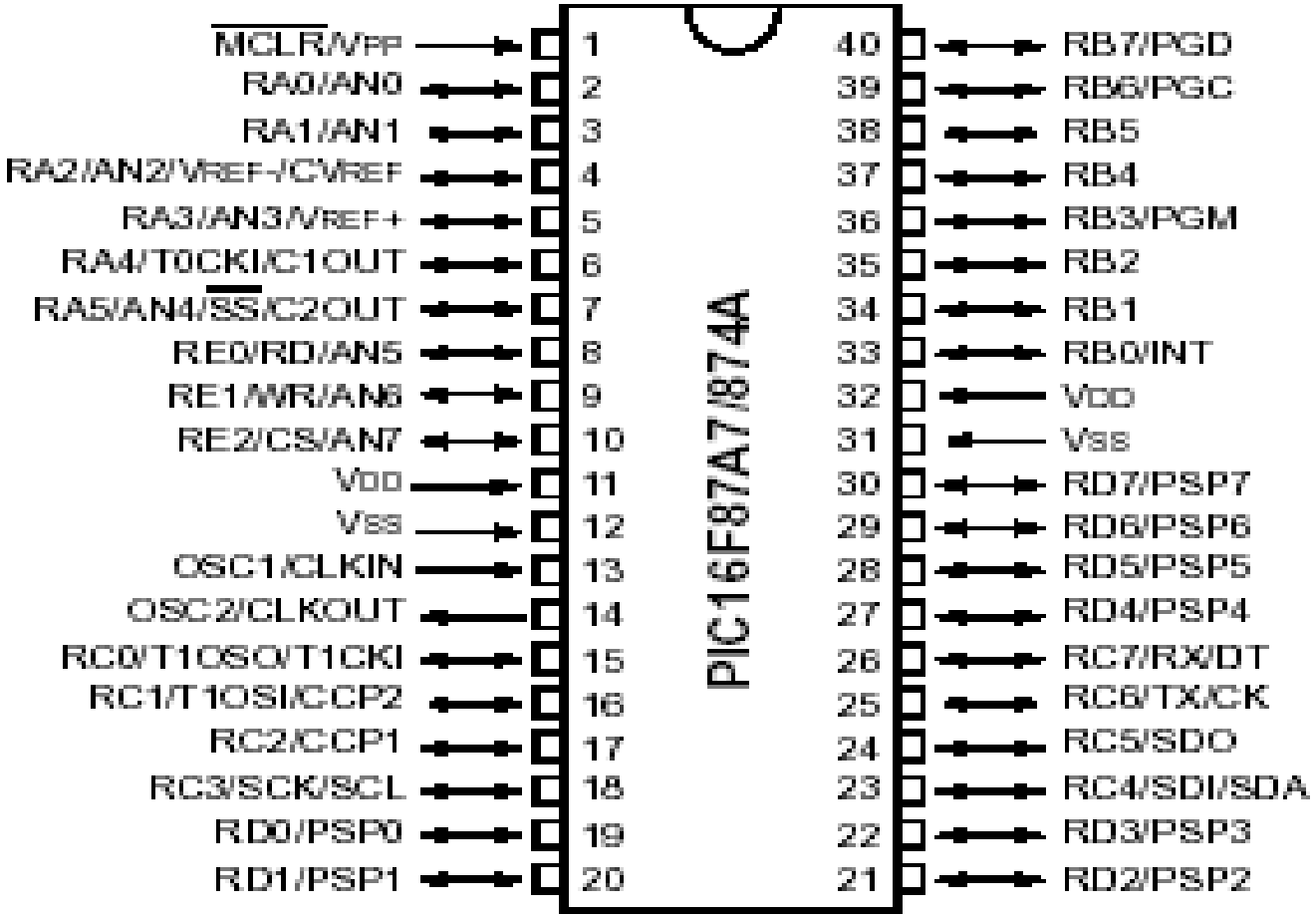
1100XX	01010101

What is New in PIC?

- High performance RISC CPU with 35 no. of instruction set only
- Harvard Architecture
- Pipelined Instructions
- And a large number of Peripherals In-built

Pin Diagram

PDIP (40 pin)



Features

- 8kB of flash program memory
- 368bytes of Data memory
- 256-EEPROM data memory
- 15 Interrupts
- In-circuit programming
- 3 internal hardware timers
- Capture/Compare/PWM modules
- Up to 8 channels of 10-Bit A/D
- Built-in USART for serial communication
- 5 digital I/O ports (Up to 22 lines)

I/O Ports

- PIC 16F877A has FIVE I/O Ports
- A total of 33 pins are used for I/O operations.

PORT A

- Port A is 6 bit wide and bi-directional.
- Its corresponding data direction register is TRISA.
- If TRISA port pin is set to 1,corresponding port A pin will act as an input pin and vice versa.
- Port A is used for analog inputs.

Port B

- Port B is 8 bit wide and bi-directional.
- Its corresponding data direction register is TRISB.
- If TRISB port pin is set to 1,corresponding port B pin will act as an input pin and vice versa.
- Port B is used for Data Transmission.

Port C

- Port C is 8 bit wide and bi-directional.
- Its corresponding data direction register is TRISC.
- If TRISC port pin is set to 1,corresponding port C pin will act as an input pin and vice versa.
- Port C is used for control registers(serial communication, I2C functions,serial data transfer).

Port D

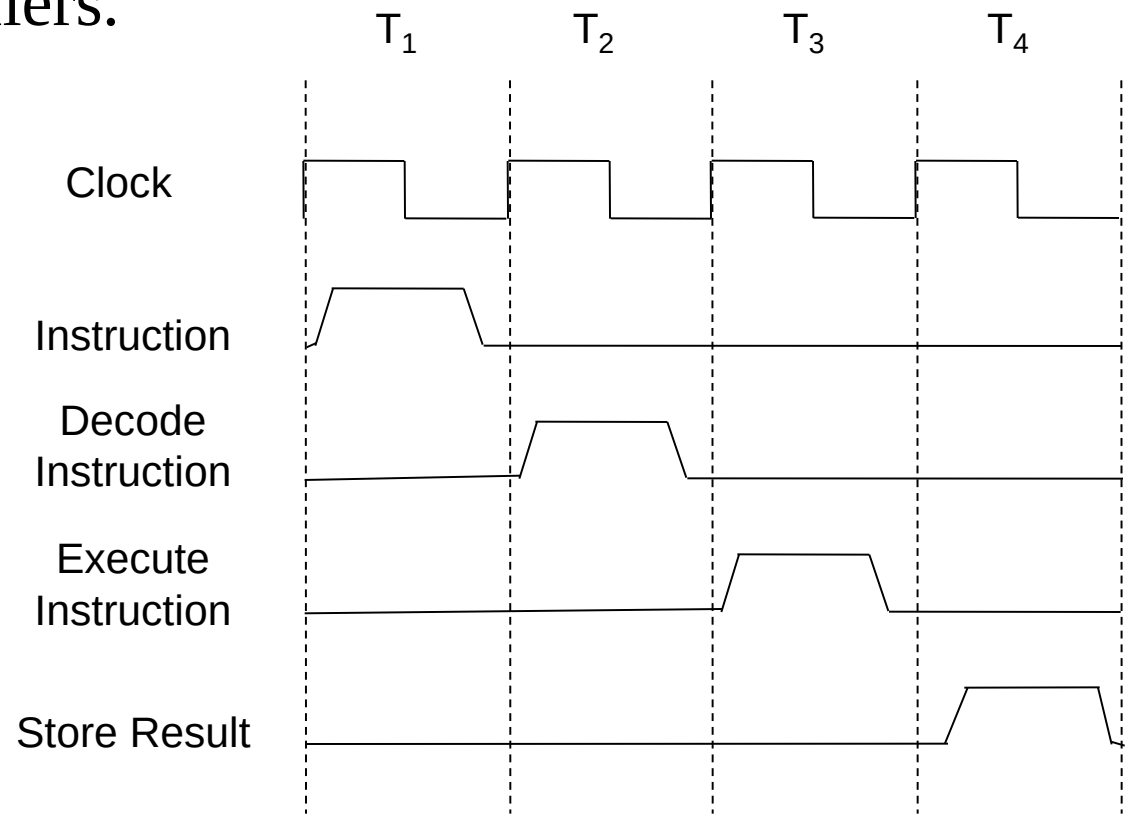
- Port D is 8 bit wide and bi-directional.
- Its corresponding data direction register is TRISD.
- If TRISD port pin is set to 1,corresponding port D pin will act as an input pin and vice versa.
- Port D is used as Data port

Port E

- Port E is 3 bit wide . They are for read, write and chip select operation.
- Each pin is individually configurable as inputs and outputs.
- Port E is generally used for controlling purposes.

Machine Cycle

4 cycles per4 cycles per instruction on the PIC16F87x micro controllers. instruction on the PIC16F87x micro controllers.



Oscillator circuits generate a clock signal.

Calculations

- A Machine cycle is the time taken for a data transfer from or to memory/ I/O Ports.
- Machine cycle is calculated using the formula:

  Clock Frequency=6.144MHz

  Machine cycle frequency= 6.144 MHz /4

  Hence 1 Machine cycle(Time taken for a data transfer)= 1/T

  = 4/ 6.144 MHz

  = 0.651 μs

Instruction Cycle

- An instruction cycle is the time taken to complete an instruction.
- All instructions in 16F877A are single cycle instructions except for Branching instruction. They take two machine cycles to complete an instruction.

PIC overview

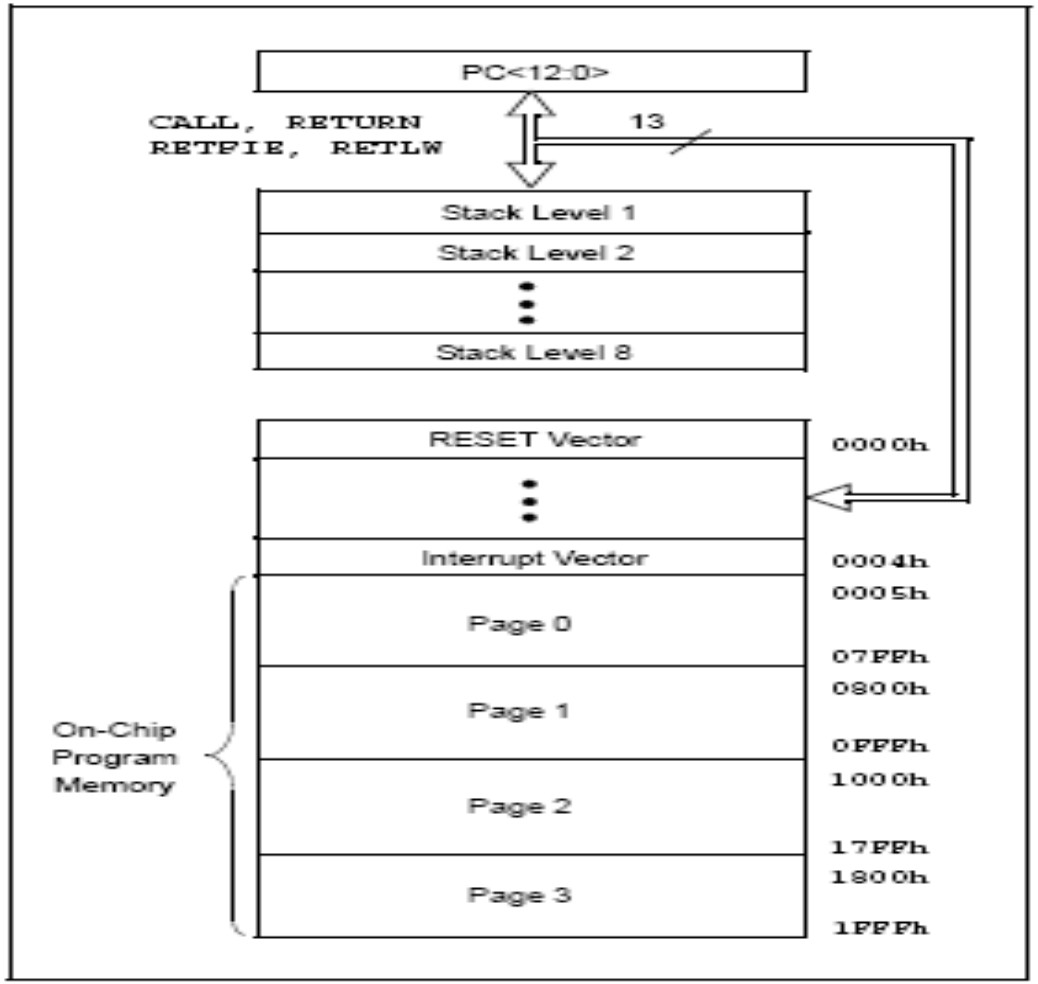
Different PICs have different on-board peripherals some common peripherals are:

- 3 Timers (0 & 2- 8bits, 1-16 bits)
- 2 Compare/Capture/PWM Modules
- Analog to Digital Converters (ADC) (8, 10 and 12bit, 50ksps)
- Serial communications: UART (RS-232C), SPI, I^2C, CAN
- Pulse Width Modulation (PWM) (10bit)
- Voltage Comparators
- Voltage Reference Modules
- MSSP – Master Synchronous Serial Port
- I2C (Master and Slave)
- SPI (Master and Slave)
- Watchdog timers, Brown out detect, LCD drivers

Overview

- Instructions
- Memory Organisation

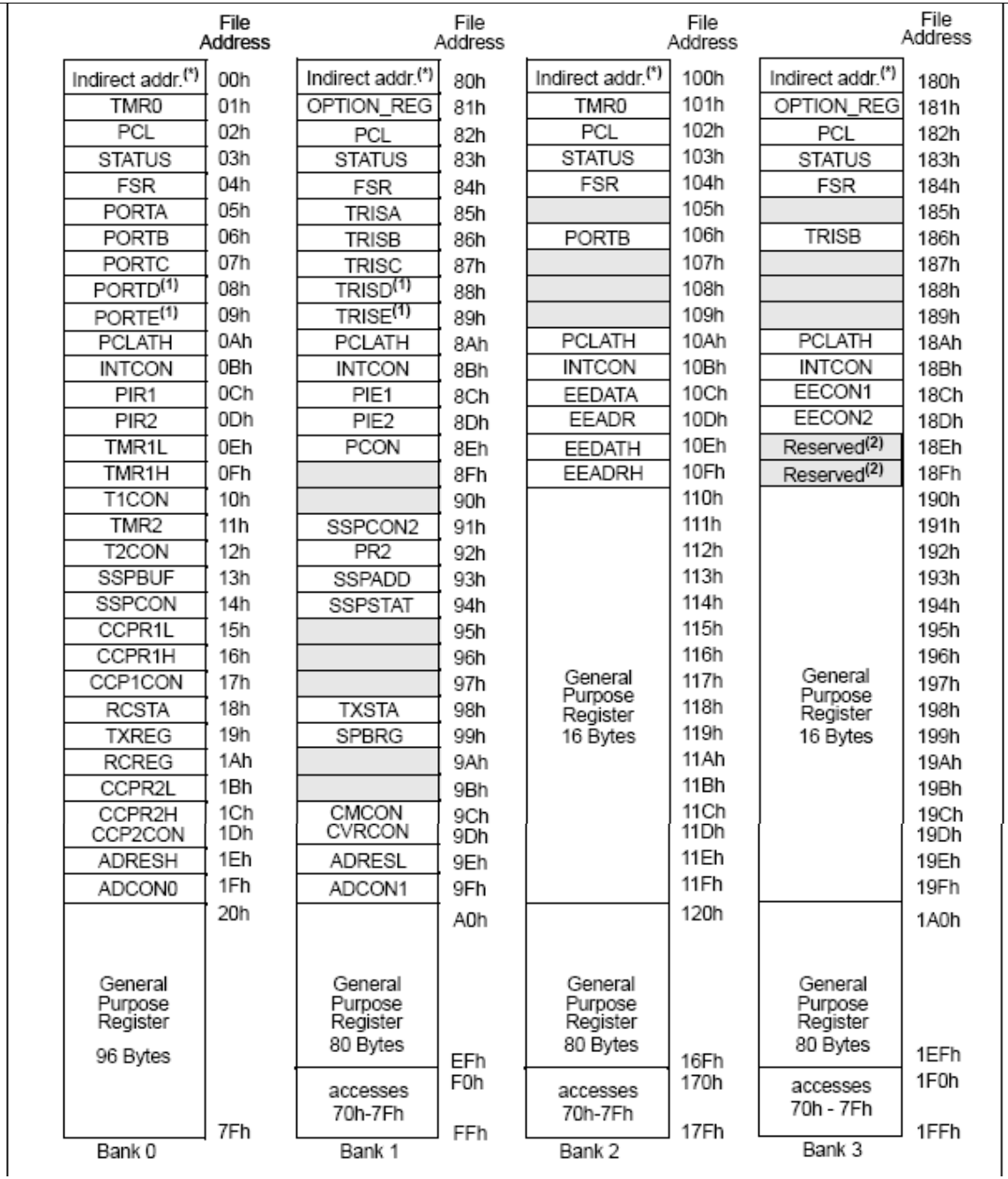
FIGURE 2-1: PIC16F876A/877A PROGRAM MEMORY MAP AND STACK



Data Memory Organisation

- The data memory partitioned into General Purpose registers and Special Function Registers.
- Bits RP1 and RP0 are the bank select bits.

RP1	RP0	Bank
0	0	0
0	1	1
1	0	2
1	1	3


File Address
Indirect addr.(*) 00h
TMR0 01h
PCL 02h
STATUS 03h
FSR 04h
PORTA 05h
PORTB 06h
PORTC 07h
PORTD(1) 08h
PORTE(1) 09h
PCLATH 0Ah
INTCON 0Bh
PIR1 0Ch
PIR2 0Dh
TMR1L 0Eh
TMR1H 0Fh
T1CON 10h
TMR2 11h
T2CON 12h
SSPBUF 13h
SSPCON 14h
CCPR1L 15h
CCPR1H 16h
CCP1CON 17h
RCSTA 18h
TXREG 19h
RCREG 1Ah
CCPR2L 1Bh
CCPR2H 1Ch
CCP2CON 1Dh
ADRESH 1Eh
ADCON0 1Fh
20h
General Purpose Register 96 Bytes
7Fh
Bank 0
File Address
Indirect addr.(*) 80h
OPTION_REG 81h
PCL 82h
STATUS 83h
FSR 84h
TRISA 85h
TRISB 86h
TRISC 87h
TRISD(1) 88h
TRISE(1) 89h
PCLATH 8Ah
INTCON 8Bh
PIE1 8Ch
PIE2 8Dh
PCON 8Eh
8Fh
90h
SSPCON2 91h
PR2 92h
SSPADD 93h
SSPSTAT 94h
95h
96h
97h
TXSTA 98h
SPBRG 99h
9Ah
9Bh
CMCON 9Ch
CVRCON 9Dh
ADRESL 9Eh
ADCON1 9Fh
A0h
General Purpose Register 80 Bytes
EFh
accesses 70h-7Fh
F0h
FFh
Bank 1
File Address
Indirect addr.(*) 100h
TMR0 101h
PCL 102h
STATUS 103h
FSR 104h
105h
PORTB 106h
107h
108h
109h
PCLATH 10Ah
INTCON 10Bh
EEDATA 10Ch
EEADR 10Dh
EEDATH 10Eh
EEADRH 10Fh
110h
111h
112h
113h
114h
115h
116h
General Purpose Register 16 Bytes
117h
118h
119h
11Ah
11Bh
11Ch
11Dh
11Eh
11Fh
120h
General Purpose Register 80 Bytes
16Fh
accesses 70h-7Fh
170h
17Fh
Bank 2
File Address
Indirect addr.(*) 180h
OPTION_REG 181h
PCL 182h
STATUS 183h
FSR 184h
185h
TRISB 186h
187h
188h
189h
PCLATH 18Ah
INTCON 18Bh
EECON1 18Ch
EECON2 18Dh
Reserved(2) 18Eh
Reserved(2) 18Fh
190h
191h
192h
193h
194h
195h
196h
General Purpose Register 16 Bytes
197h
198h
199h
19Ah
19Bh
19Ch
19Dh
19Eh
19Fh
1A0h
General Purpose Register 80 Bytes
1EFh
accesses 70h - 7Fh
1F0h
1FFh
Bank 3

Timers

PIC16F877A has 3 Timers

- Timer 0 - 8 bit

  can be used as a Timer/counter

- Timer 1 – 16 bit

  can be used as a Timer/counter

- Timer 2 – 8 bit Timer

  can be used as the PWM time-base for the PWM mode of the CCP module.

REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN**: Timer1 Oscillator Enable Control bit

1 = Oscillator is enabled
0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)

bit 2 **$\overline{\text{T1SYNC}}$**: Timer1 External Clock Input Synchronization Control bit

When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input

When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS**: Timer1 Clock Source Select bit

1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR1ON**: Timer1 On bit

1 = Enables Timer1
0 = Stops Timer1

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Timer 1

- The data registers of Timer 1 are TMR1H and TMR1L.
- The 16 bit value to be loaded in the timer is given to Data registers
- The Control register is T1CON.
- The timer ON and oscillator enable operations are controlled through the control register.
- To determine if the timer has completed its operation, the timer overflow bit TMR1IF of the PIR1 register is checked continuously.

USART

- Universal synchronous asynchronous receiver transmitter is also known as a serial communication interface.
- USART can be configured in the following modes:
  - Asynchronous - FULL DUPLEX
  - synchronous - HALF DUPLEX

Serial Communication

- For serial Communication, the SFR involved are TXSTA, TXREG, RCSTA,,RCREG and SPBRG.
- SPBRG is used to set the required baud rate.
- In TXSTA,TXEN is set high to enable transmission.
- The transmitted data is stored in a temporary buffer,TXREG.
- In RCSTA, SPEN and CREN are set high to enable reception.
- The received data is stored in a temporary buffer, RCREG.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC**: Clock Source Select bit

Asynchronous mode:
Don't care

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented**: Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D**: 9th bit of Transmit Data, can be Parity bit

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode - Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - Slave:
Don't care

bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2 **FERR**: Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR**: Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data (can be parity bit, but must be calculated by user firmware)

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate = Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	N/A

X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

FOR BAUD RATE:

For asynchronous mode,

Baud rate=Fosc / 64(x+1)

For synchronous mode,

Baud rate=Fosc / 4(x+1)

FOR RECEPTION:

Check RCIF flag till it is set. This indicates that the data is completely received in the RCREG. Move this value to working register for further manipulation.

FOR TRANSMISSION:

To transmit the calculated data, move the value to TXREG. Check the TXIF flag till it is set. This indicates that the data is completely transmitted out.

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF**: Parallel Slave Port Read/Write Interrupt Flag bit(1)
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

Note 1: PSPIF is reserved on PIC16F873A/876A devices; always maintain this bit clear.

bit 6 **ADIF**: A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4 **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI
  - A transmission/reception has taken place.
- I^2C Slave
  - A transmission/reception has taken place.
- I^2C Master
  - A transmission/reception has taken place.
  - The initiated START condition was completed by the SSP module.
  - The initiated STOP condition was completed by the SSP module.
  - The initiated Restart condition was completed by the SSP module.
  - The initiated Acknowledge condition was completed by the SSP module.
  - A START condition occurred while the SSP module was idle (Multi-Master system).
  - A STOP condition occurred while the SSP module was idle (Multi-Master system).

0 = No SSP interrupt condition has occurred

bit 2 **CCP1IF**: CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode

bit 1 **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Interrupts

- PIC16F877A has 15 sources of interrupts.
- Only one vector location is allocated for the interrupts.
- The Global interrupt enable bit is set, so that any interrupt can be acknowledged.
- Regardless of the Global enable bit, the user should ensure,the appropriate interrupt flag bits that needs to be executed, is also set.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE**: Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **TMR0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

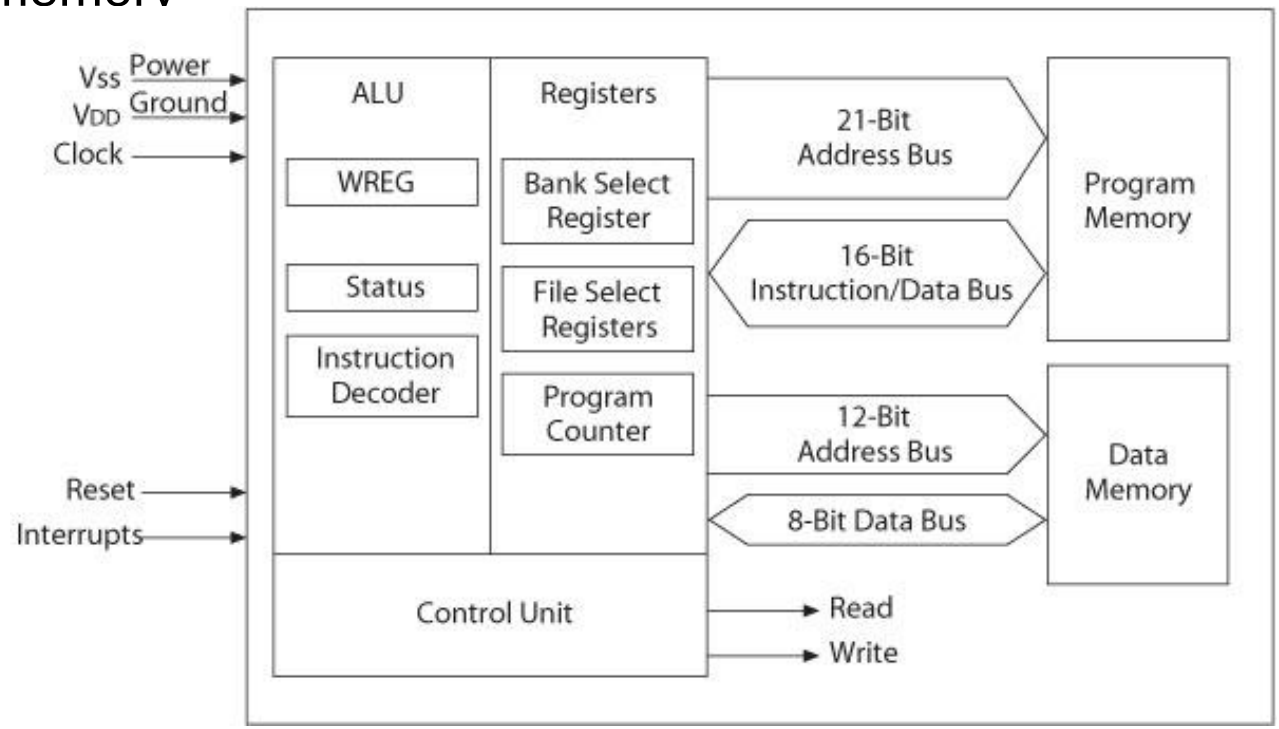
bit 1 **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
0 = None of the RB7:RB4 pins have changed state

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

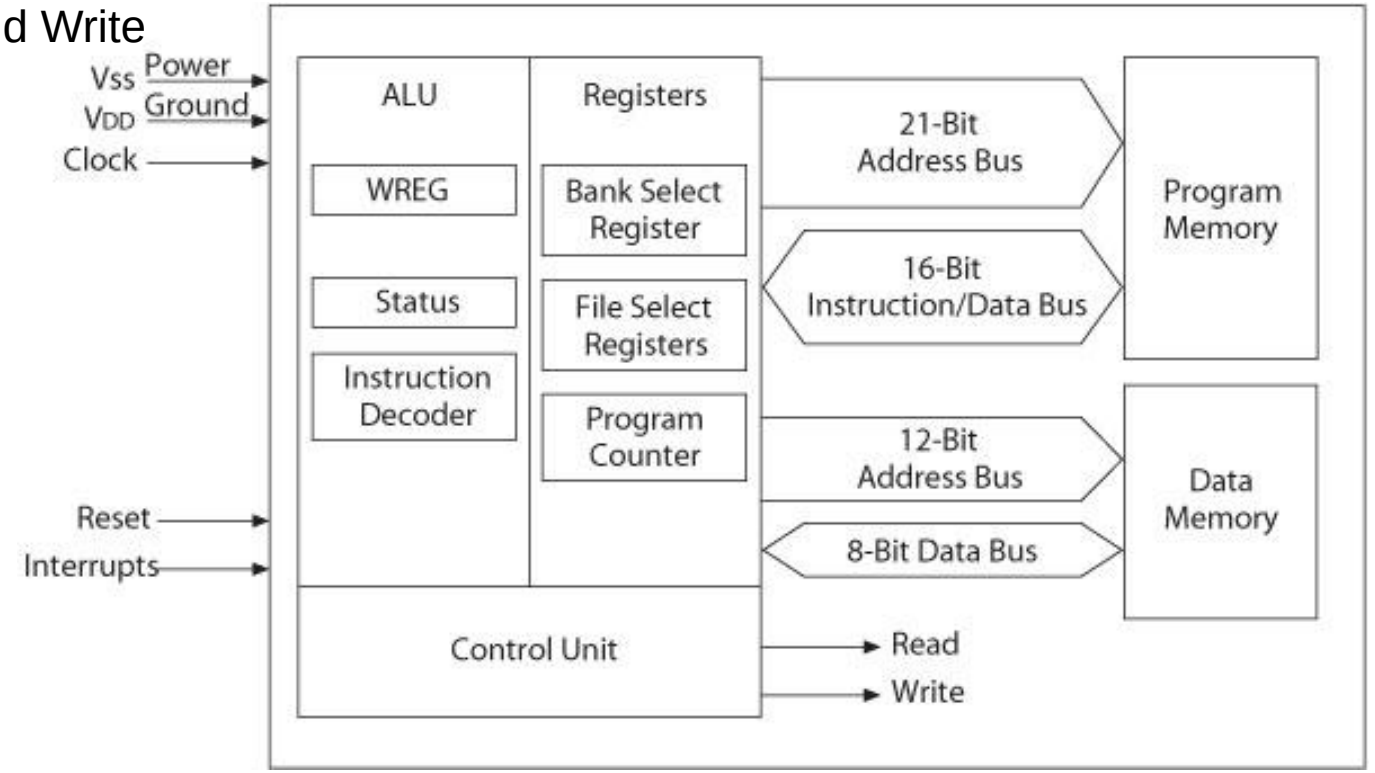
PIC18F - Address Buses

- Address bus
  - 21-bit address bus for program memory addressing capacity: 2 MB of memory
  - 12-bit address bus for data memory addressing capacity: 4 KB of memory



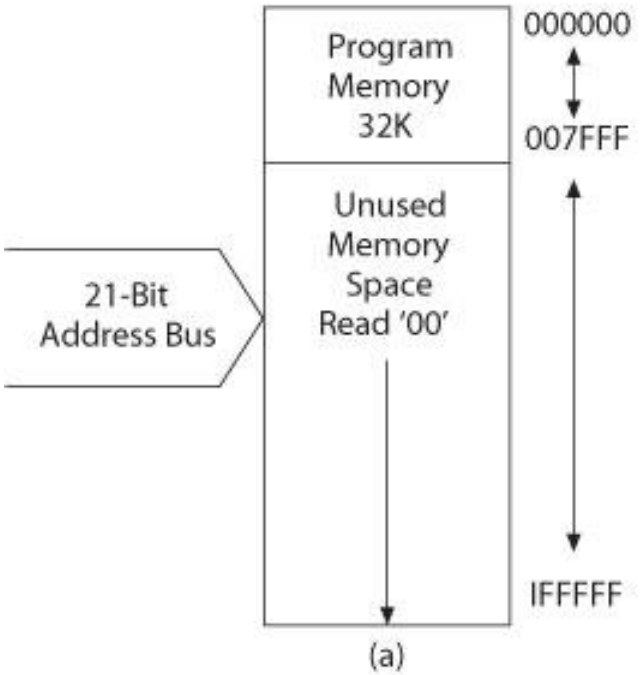
Data Bus and Control Signals

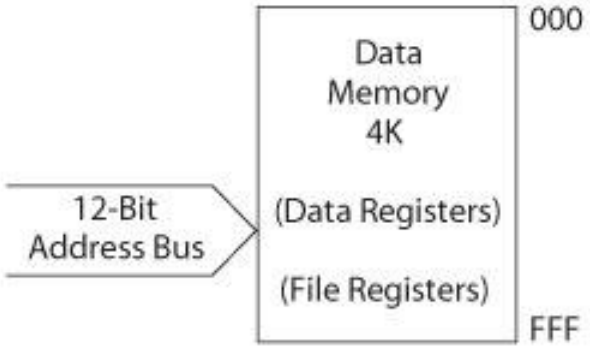
- Data bus
  - 16-bit instruction/data bus for program memory
  - 8-bit data bus for data memory
- Control signals
  - Read and Write



PIC18F452/4520 Memory

- Program memory with addresses (Flash)
- Data memory with addresses



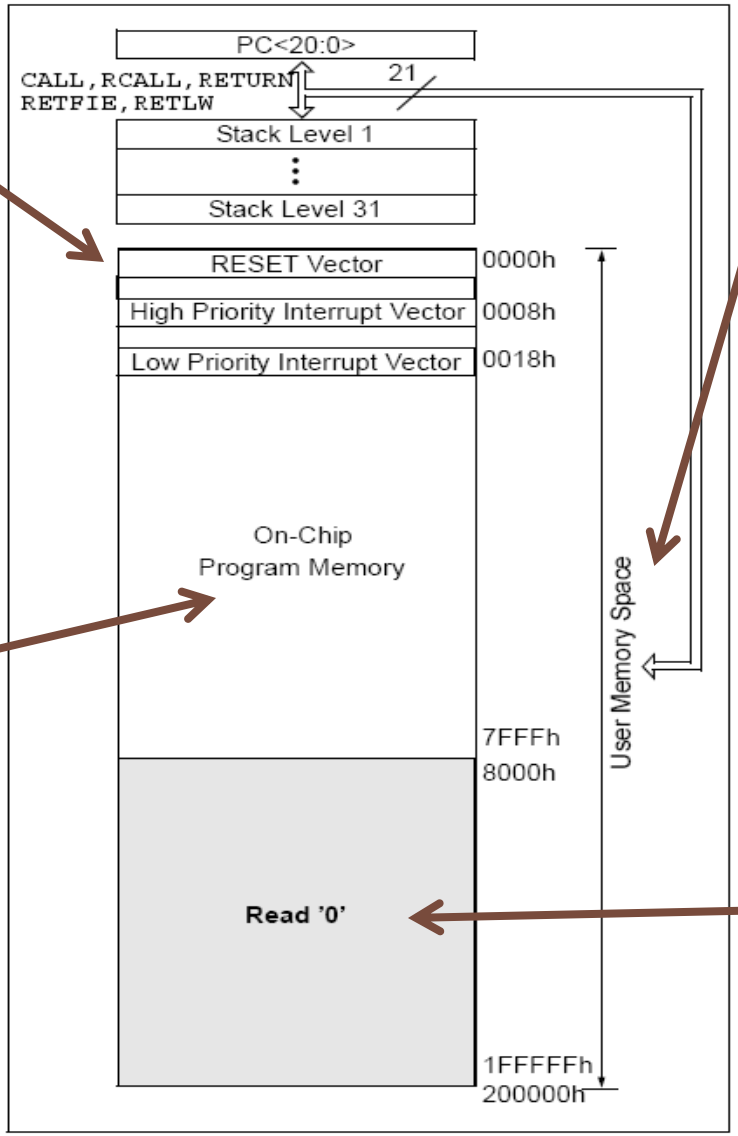


$FFF=2^{12}=16x256=4096=4K$

Program Memory

The RESET vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

PIC18F452 each have 32 Kbytes of FLASH memory.
This means that it can store up to 16K of single word instructions

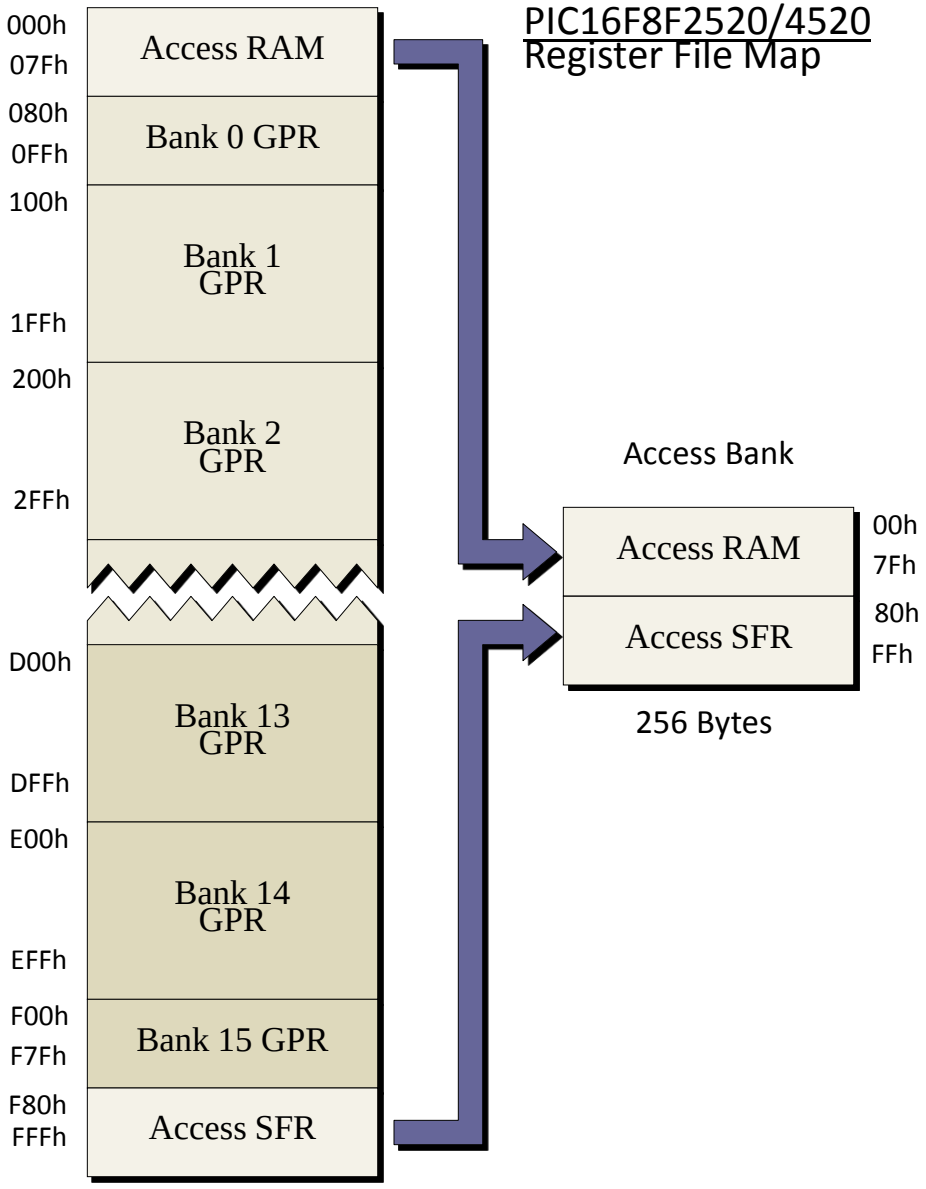


A 21-bit program counter is capable of addressing the 2-Mbyte program memory space.

Accessing a location between the physically implemented memory and the 2-Mbyte address will cause a read of all ’0’s (a NOP instruction).

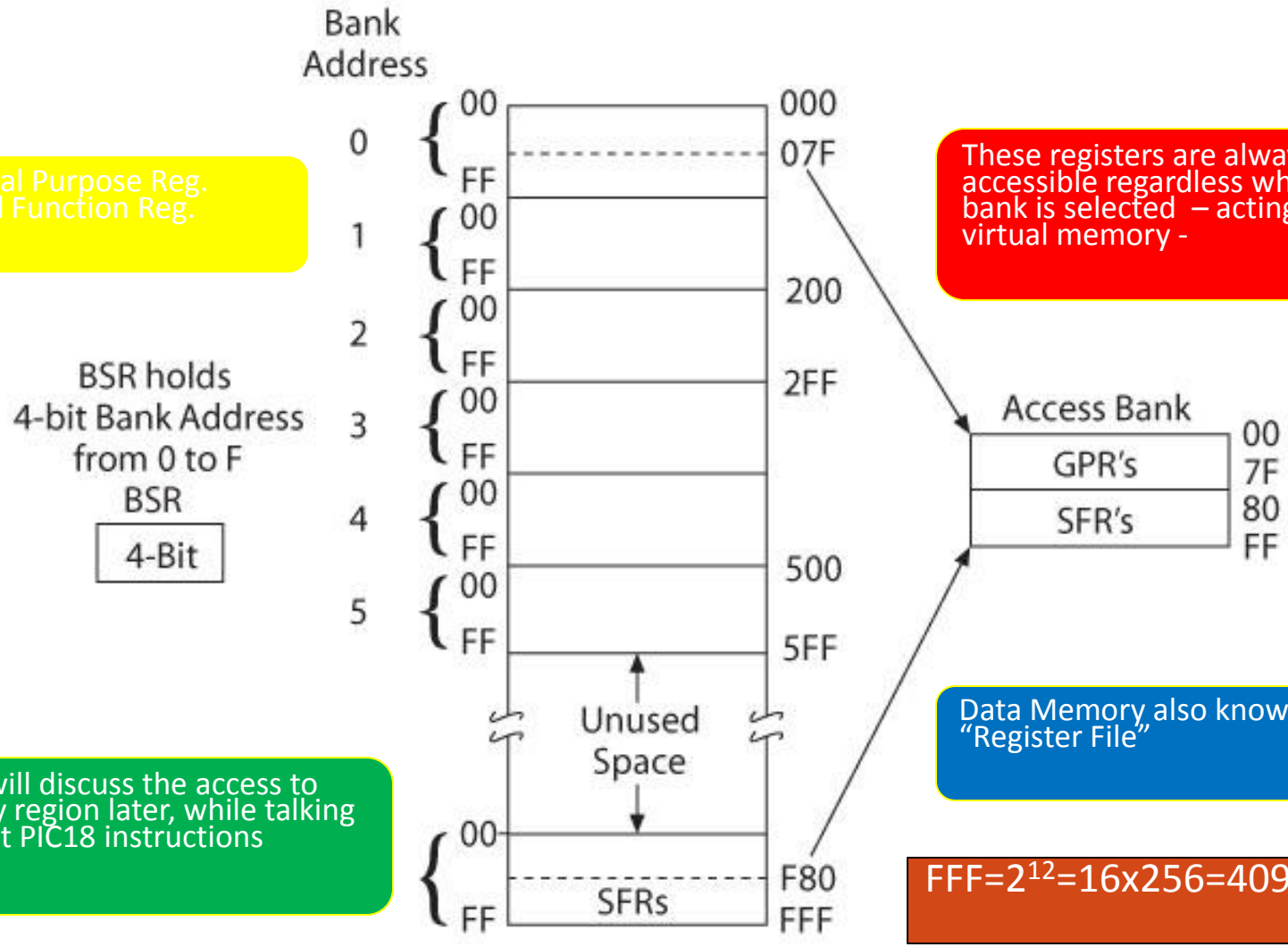
Data Memory Organization

- Data Memory up to 4k bytes
- Divided into 256 byte banks
- Half of bank 0 and half of bank 15 form a virtual bank that is accessible no matter which bank is selected



Data Memory with Access Banks

GPR=General Purpose Reg.
SFR=Special Function Reg.



These registers are always accessible regardless which bank is selected – acting as a virtual memory -

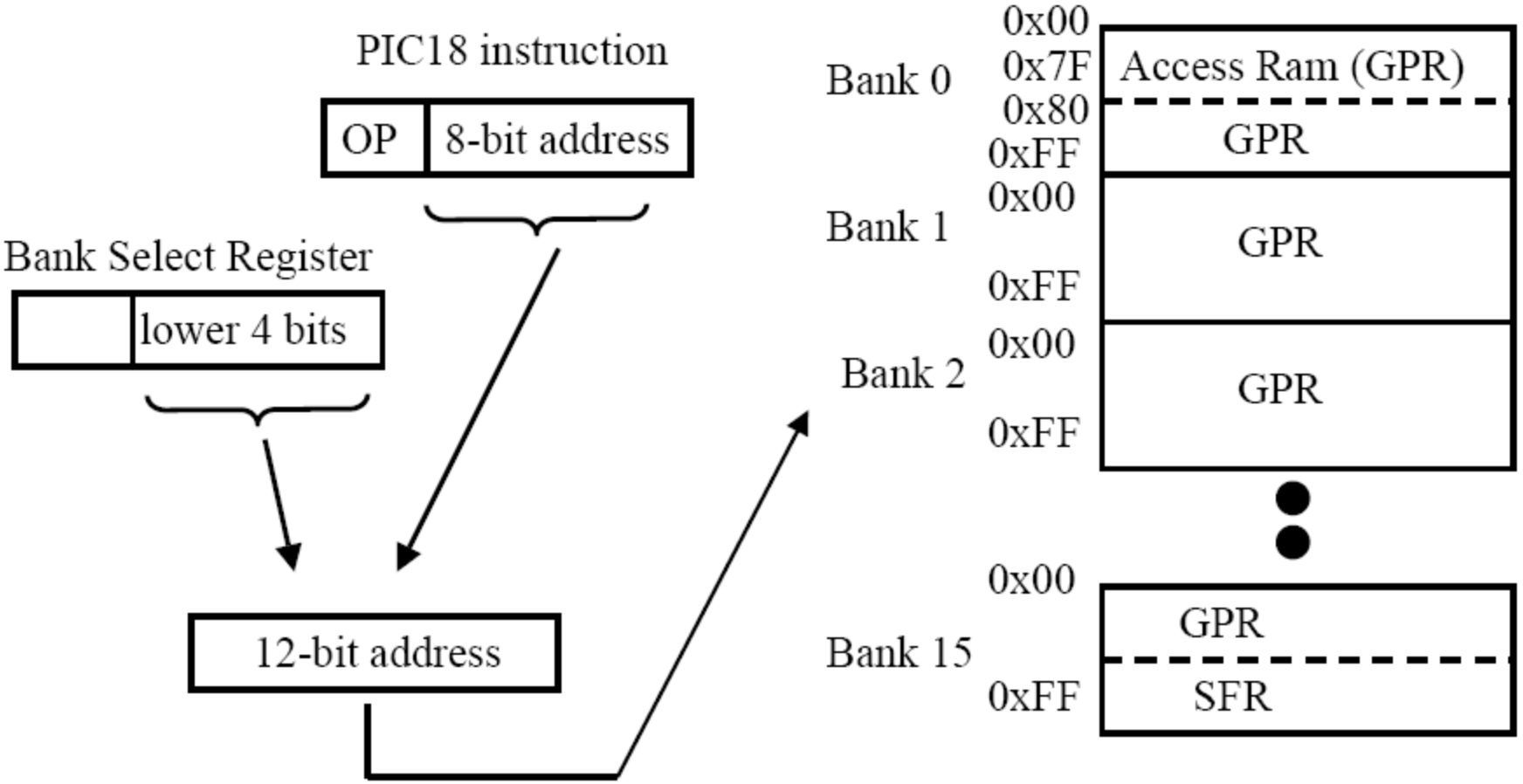
Data Memory also known as "Register File"

We will discuss the access to every region later, while talking about PIC18 instructions

FFF=2^{12}=16x256=4096=4K

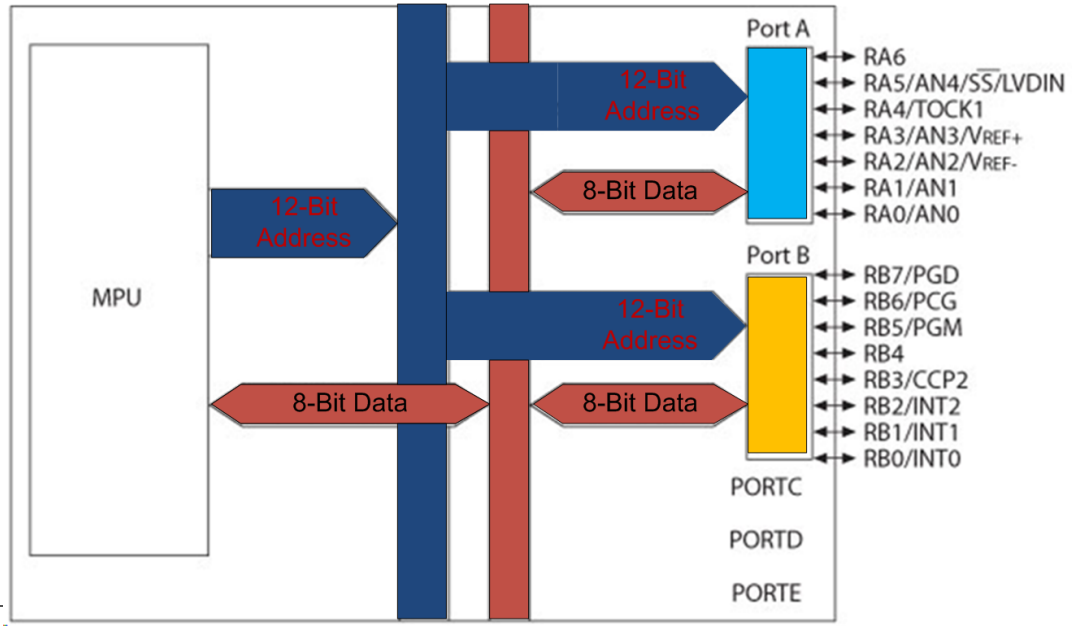
Accessing Data Memory

- The machine code for a PIC18 instruction has only **8** bits for a data memory address which needs 12 bits. The Bank Select Register (BSR) supplies the other 4

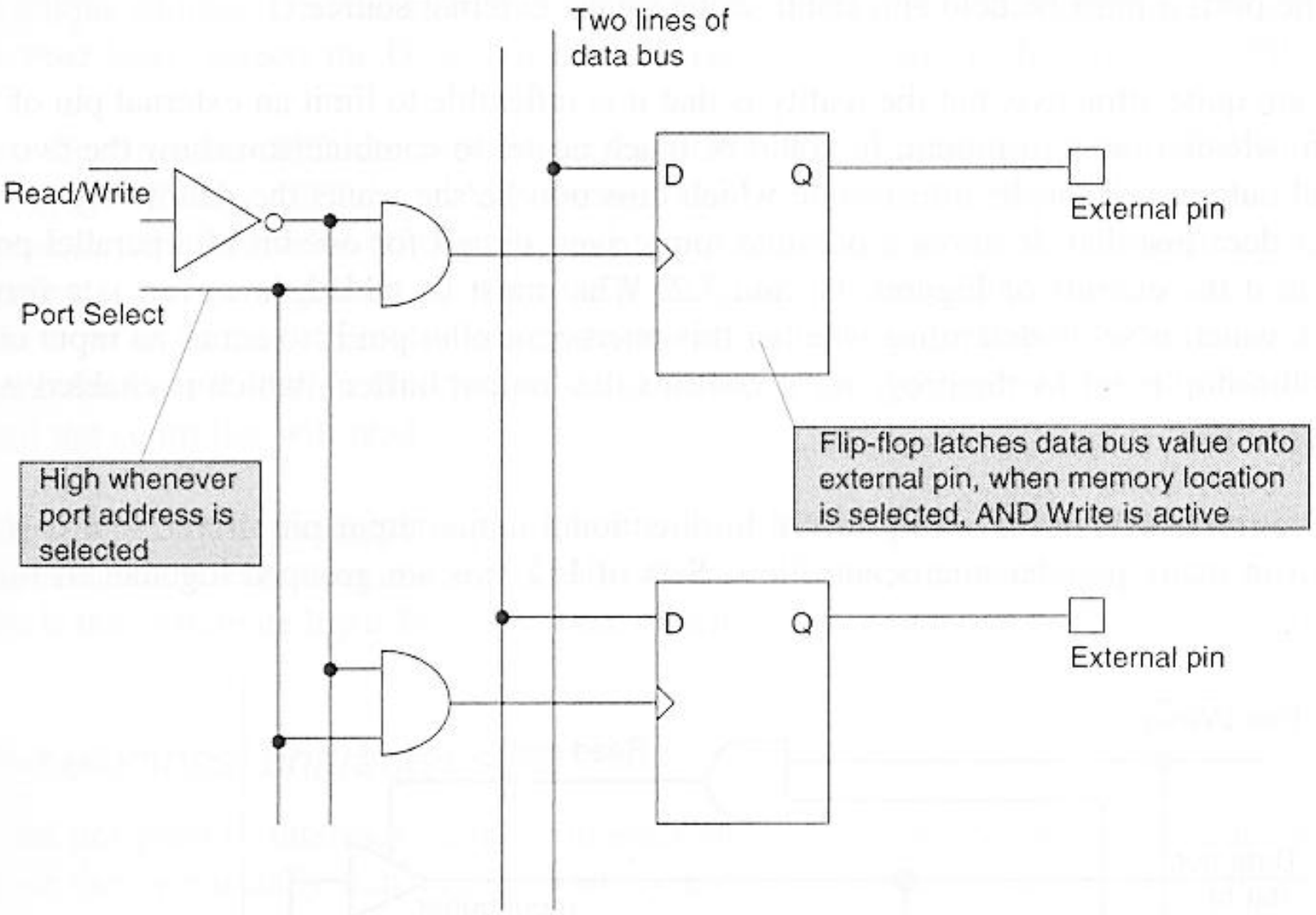


PIC18F452 I/O Ports

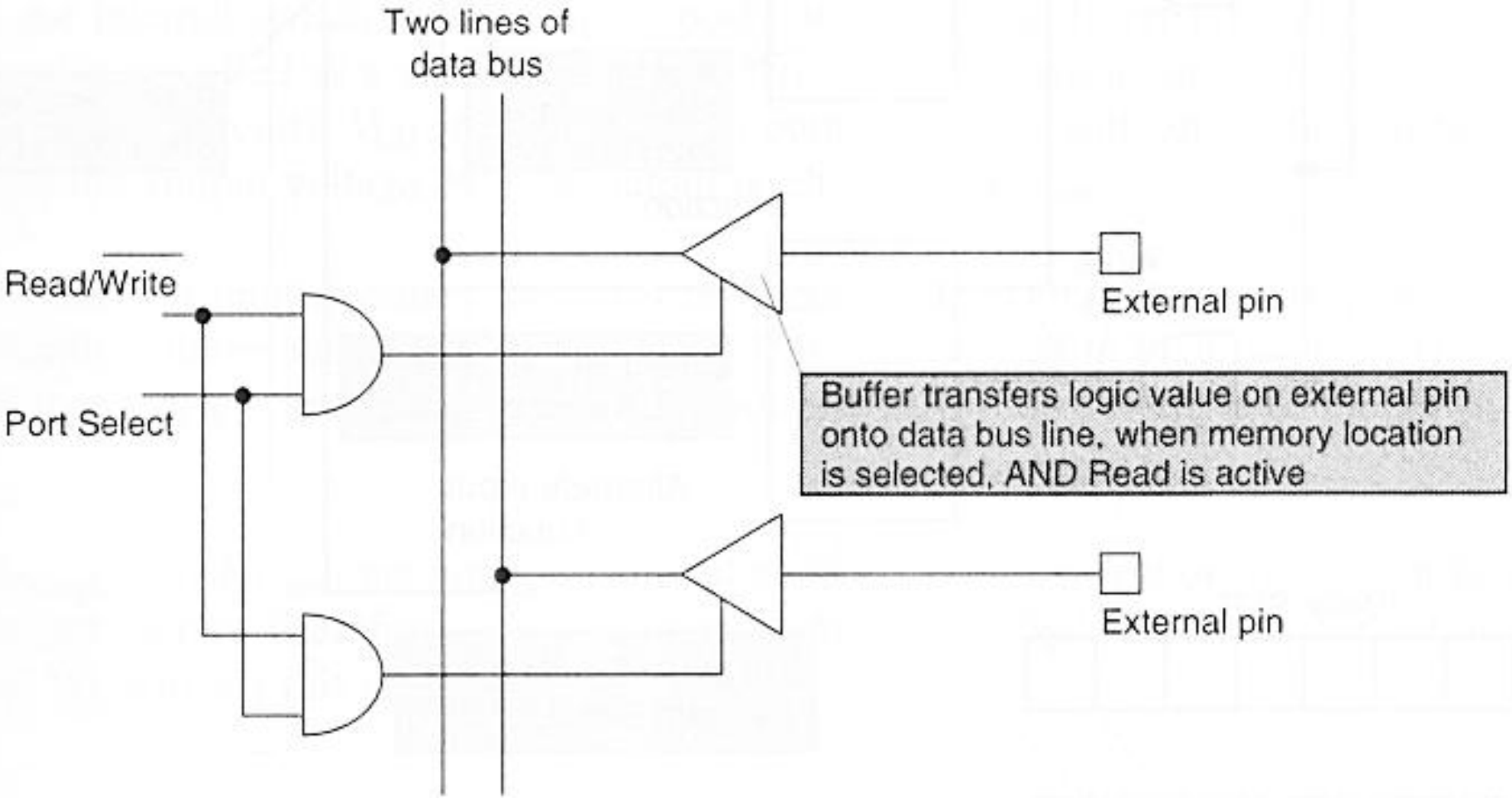
- Five I/O ports
  - PORT A through PORT E
  - Most I/O pins are multiplexed
  - Generally have eight I/O pins with a few exceptions
  - Addresses already assigned to these ports in the design stage
  - Each port is identified by its assigned SFR



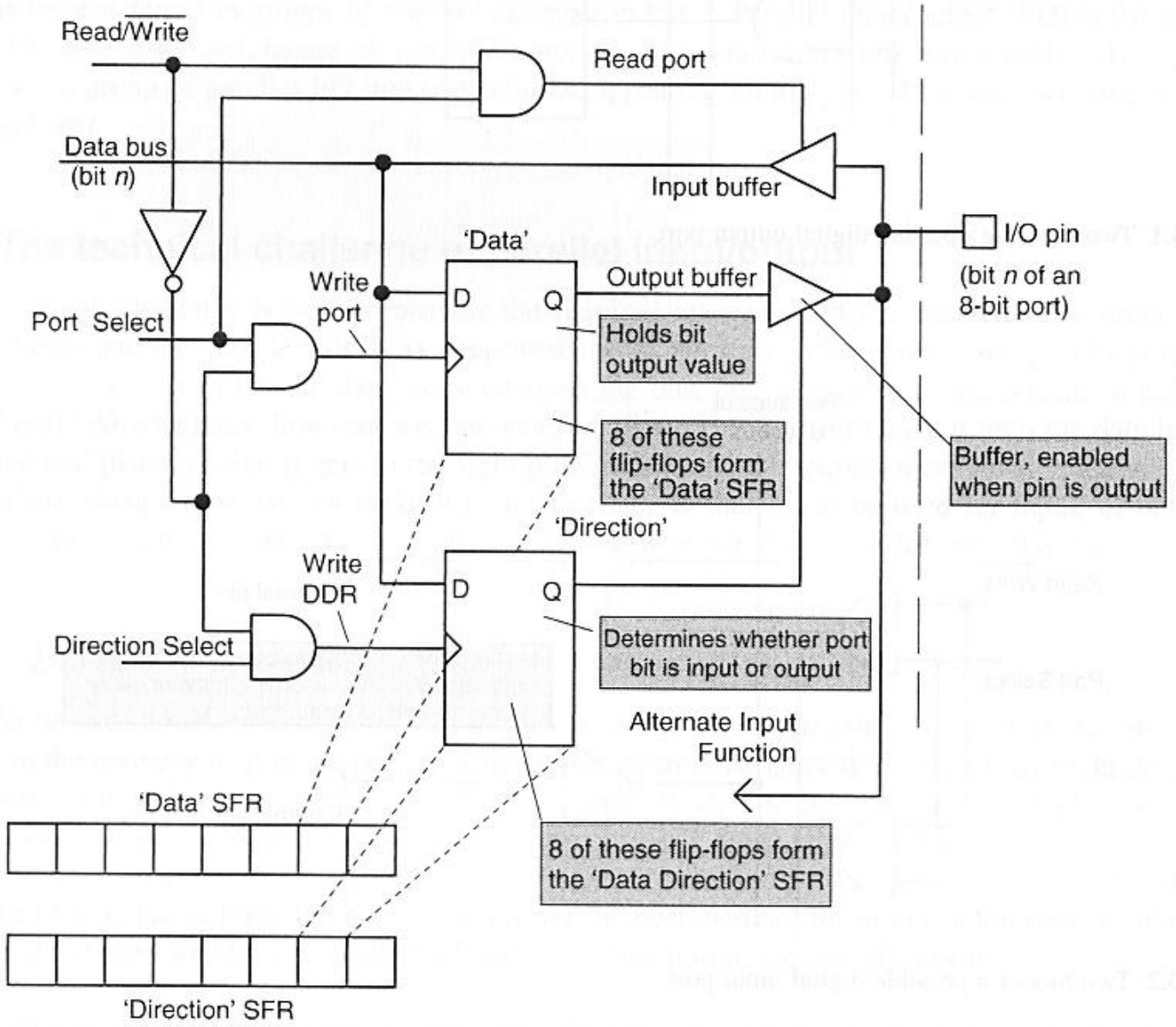
Parallel I/O Output Structure



Parallel I/O Input Structure



Parallel I/O Combined I/O Structure



Parallel I/O ports Main Features

- Simple memory mapped access
- Can be configured through software as either input or output
- Ability to set or reset individual bits
- Can have internal pull-ups
- Can drive small loads like LEDs
- Can be multifunction
- Different capability for pins (i.e. larger current)

Parallel I/O ports

- For most ports, the I/O pin's direction (input or output) is controlled by the data direction register **TRISx** (x=A,B,C,D,E): a '1' in the TRIS bit corresponds to that pin being an input, while a '0' corresponds to that pin being an output
- The **PORTx** register is the latch for the data to be output. Reading PORTx register read the status of the pins, whereas writing to it will write to the port latch.
- **Example: Initializing PORTB** (PORTB is an 8-bit port. Each pin is individually configurable as an input or output).

```
bcf     STATUS, RP0          ; select bank0
bcf     STATUS, RP1
clrf    PORTB                ; clear PORTB output data latches
bsf     STATUS, RP0          ; select bank1
movlw   0xCF                 ; value used to initialize data direction
movwf   TRISB          ; PORTB<7:6>=inputs, PORTB<5:4>=outputs,
                       ; PORTB<3:0>=inputs
```

Relationship between TRIS and PORT Registers

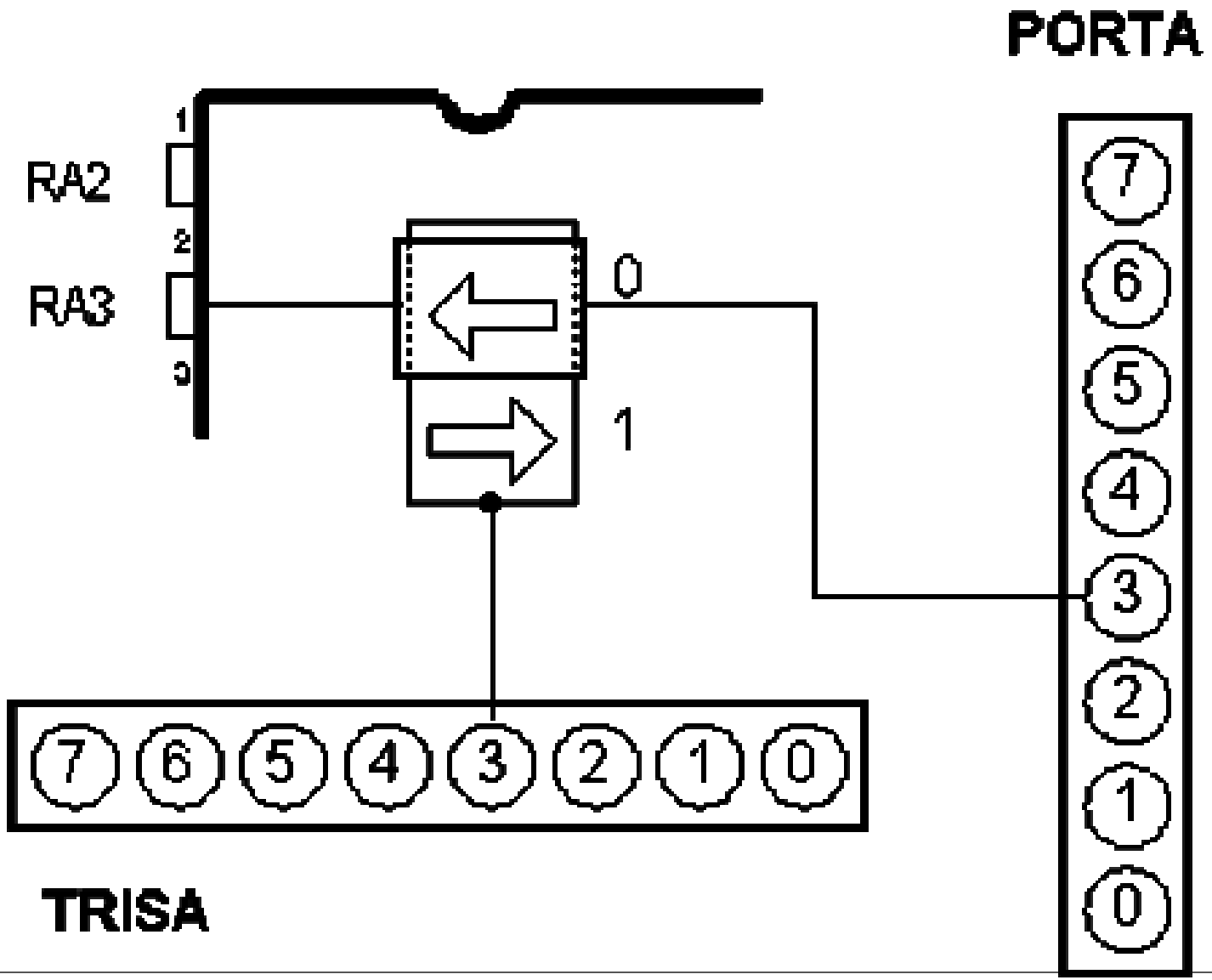


Illustration: Displaying a Byte at an I/O Port (1 of 5)

- Problem statement:
  - Write instructions to light up alternate LEDs at PORTC.
- Hardware:
  - PORTC
    - bidirectional (input or output) port; should be setup as output port for display
  - Logic 1 will turn on an LED in Figure 2.10.

Illustration (2 of 5)

- Interfacing LEDs to PORTC
- Port C is F82H
- Note that PORT C is set to be an output!
- Hence, TRISC (address 94H) has to be set to 0

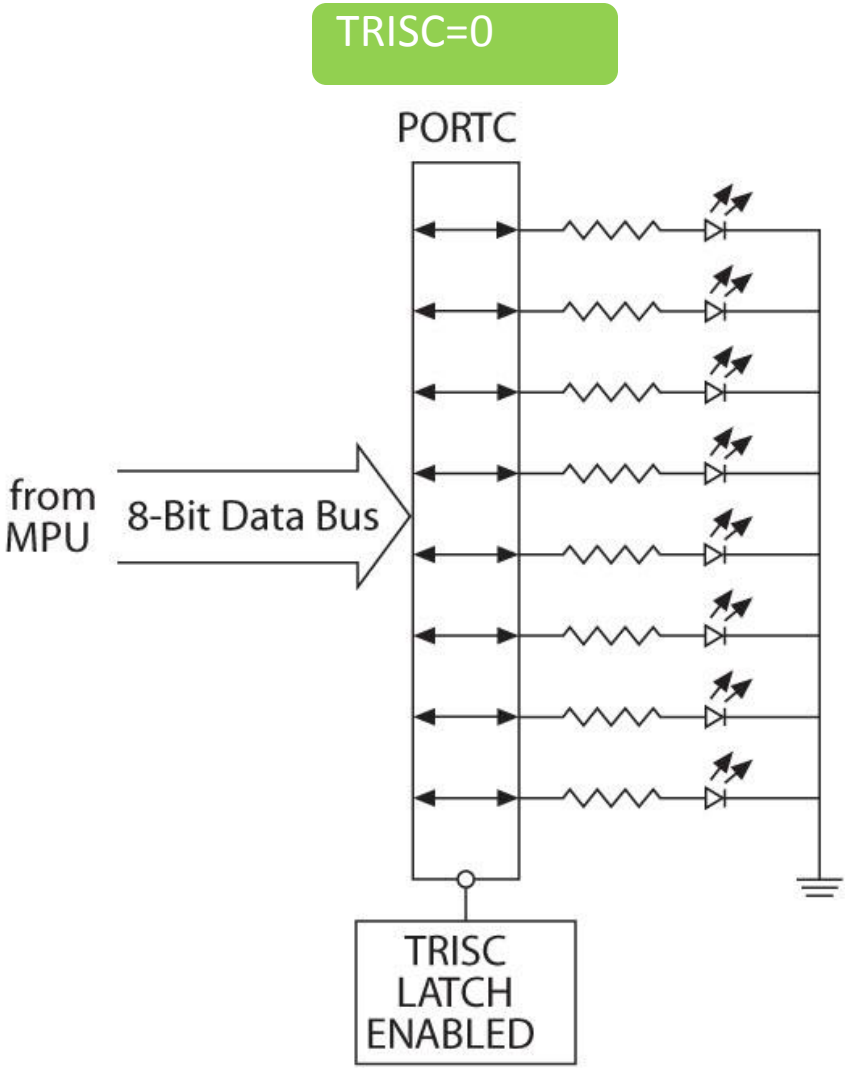


Illustration (3 of 5)

- Program (software)
  - Logic 0 to TRISC sets up PORTC as an output port
  - Byte 55H turns on alternate LEDs
    - MOVLW 00 ;Load W register with 0
    - MOVWF TRISC, 0 ;Set up PORTC as output
    - MOVLW 0x55 ;Byte 55H to turn on LEDS
    - MOVWF PORTC,0 ;Turn on LEDs
    - SLEEP ;Power down