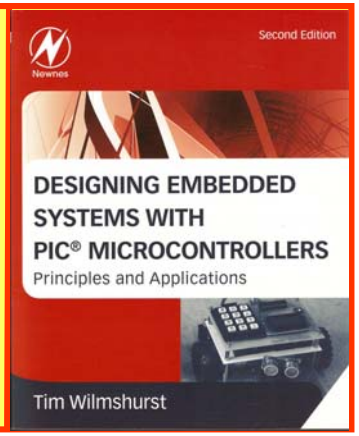
Designing Embedded Systems with PIC Microcontrollers:
Principles and Applications

2nd Edition. Tim Wilmshurst



Chapter 7
Larger Systems and the PIC 16F873A

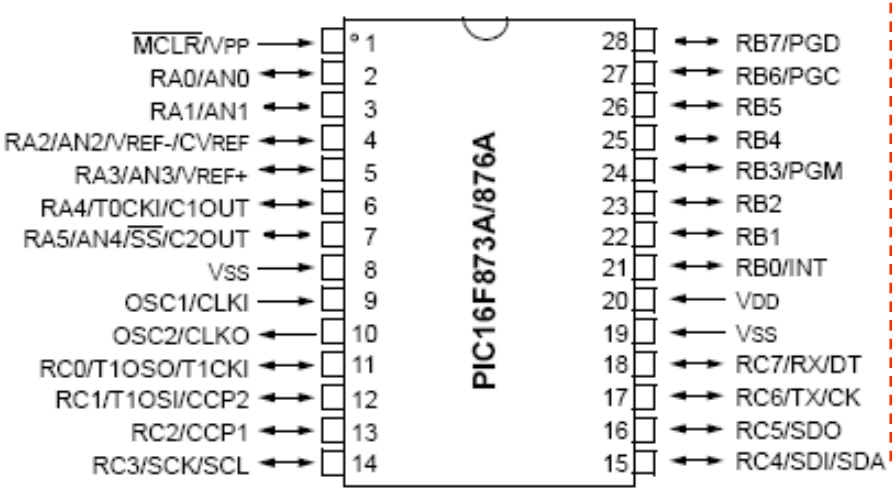
The aims of this chapter are to introduce:

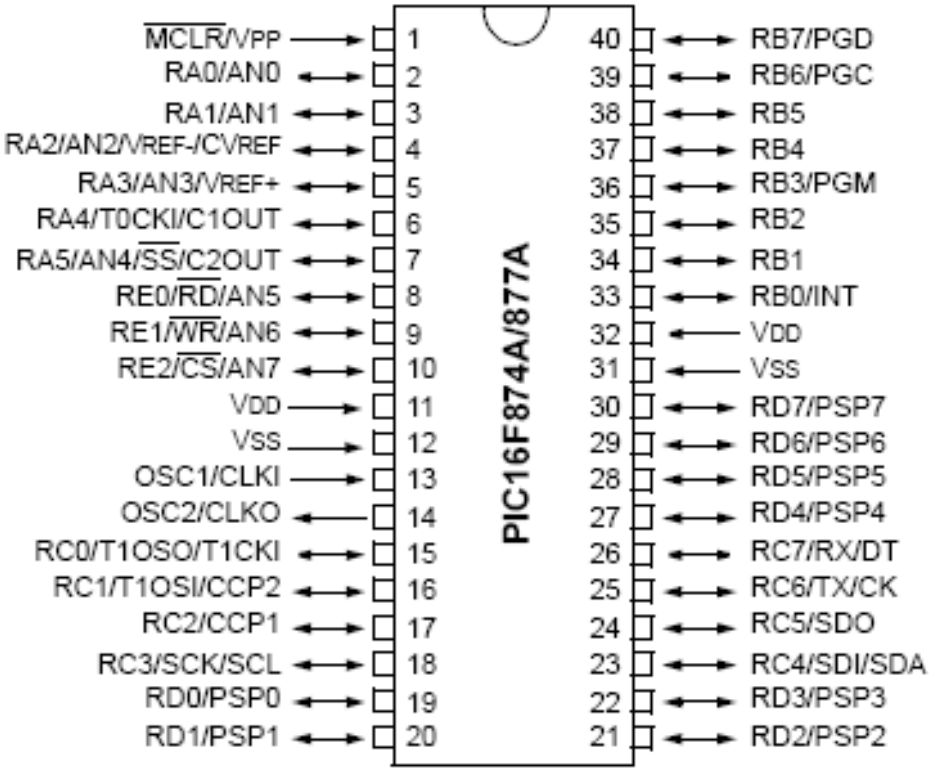
- *The architecture of the 16F873A microcontroller;*
- *The 16F873A memory map;*
- *The 16F873A interrupt structure;*
- *The use of the Microchip in-circuit debugger.*

The 16F87XA Microcontroller - Some Overview Information

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

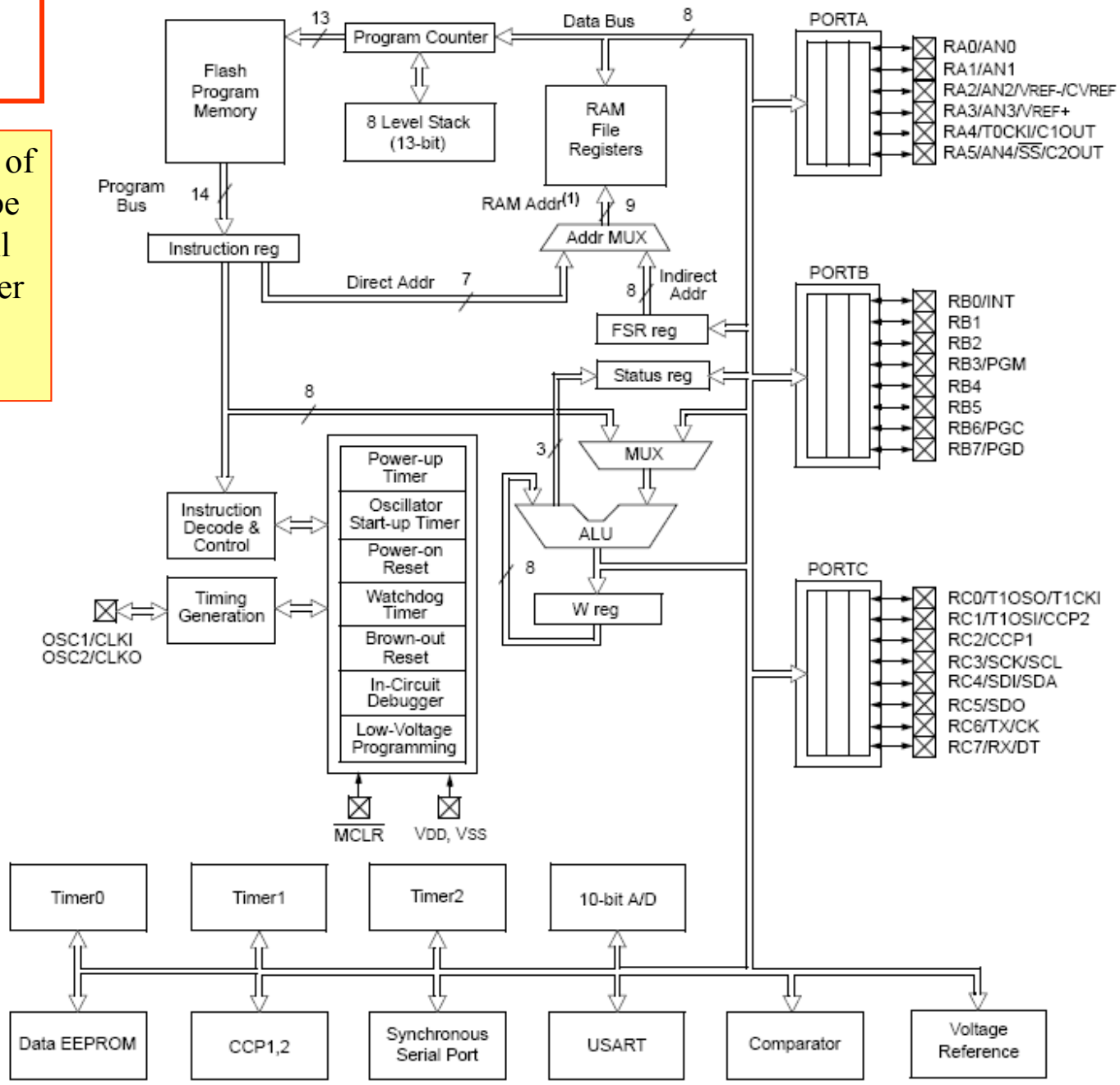
The 16F873A is one of a group of 4 very similar 16 Series PIC microcontrollers. It comes in the smaller of two package sizes, and has the smallest memory.



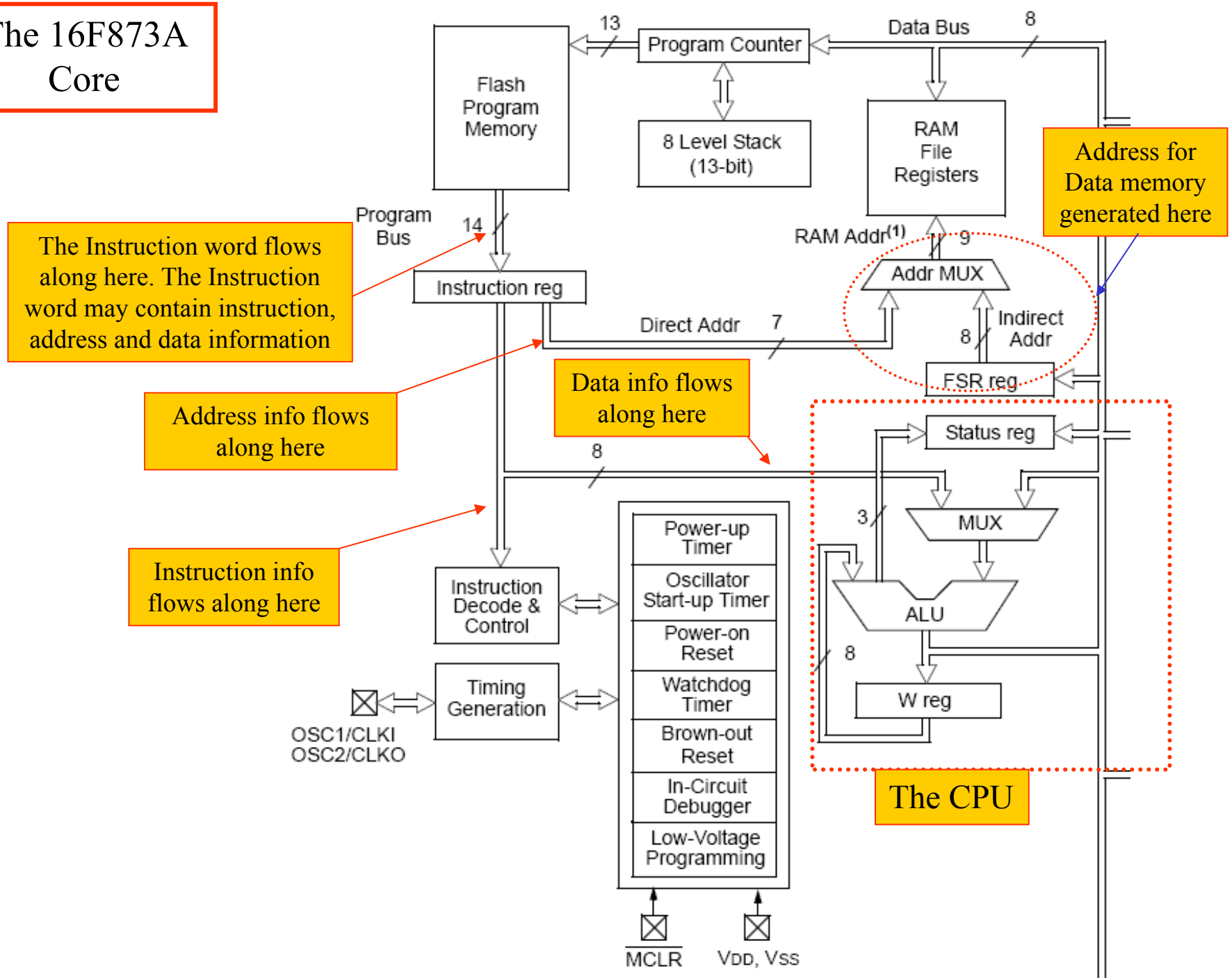


The PIC 16F873A Block Diagram

As a direct "big brother" of the 16F84A, it should be possible to recognise all the features of the smaller device in the larger 16F873A.



The 16F873A Core



The Status Register

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h-1FFh)
10 = Bank 2 (100h-17Fh)
01 = Bank 1 (80h-FFh)
00 = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

bit 4 **$\overline{TO}$**: Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

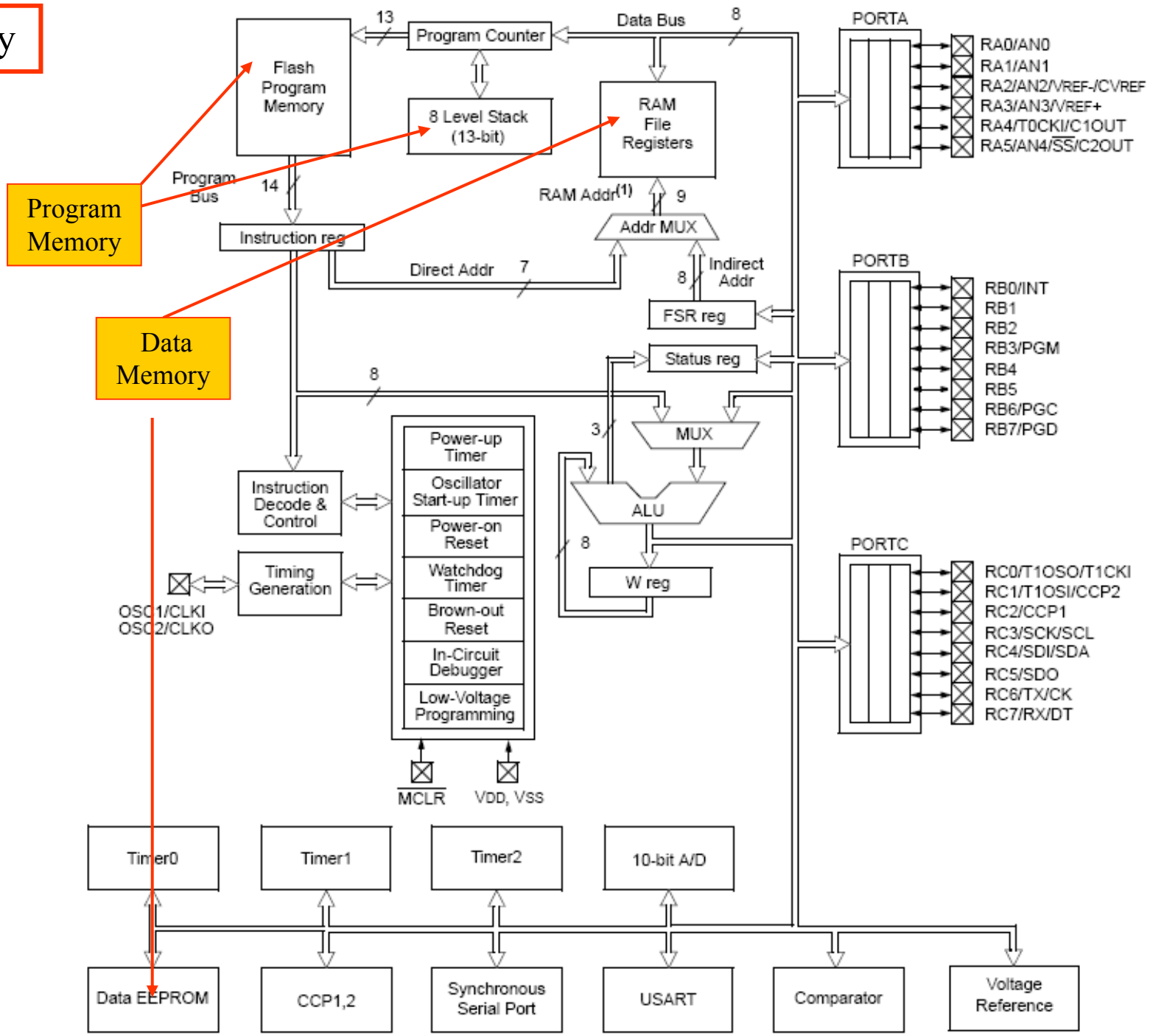
bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
(for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

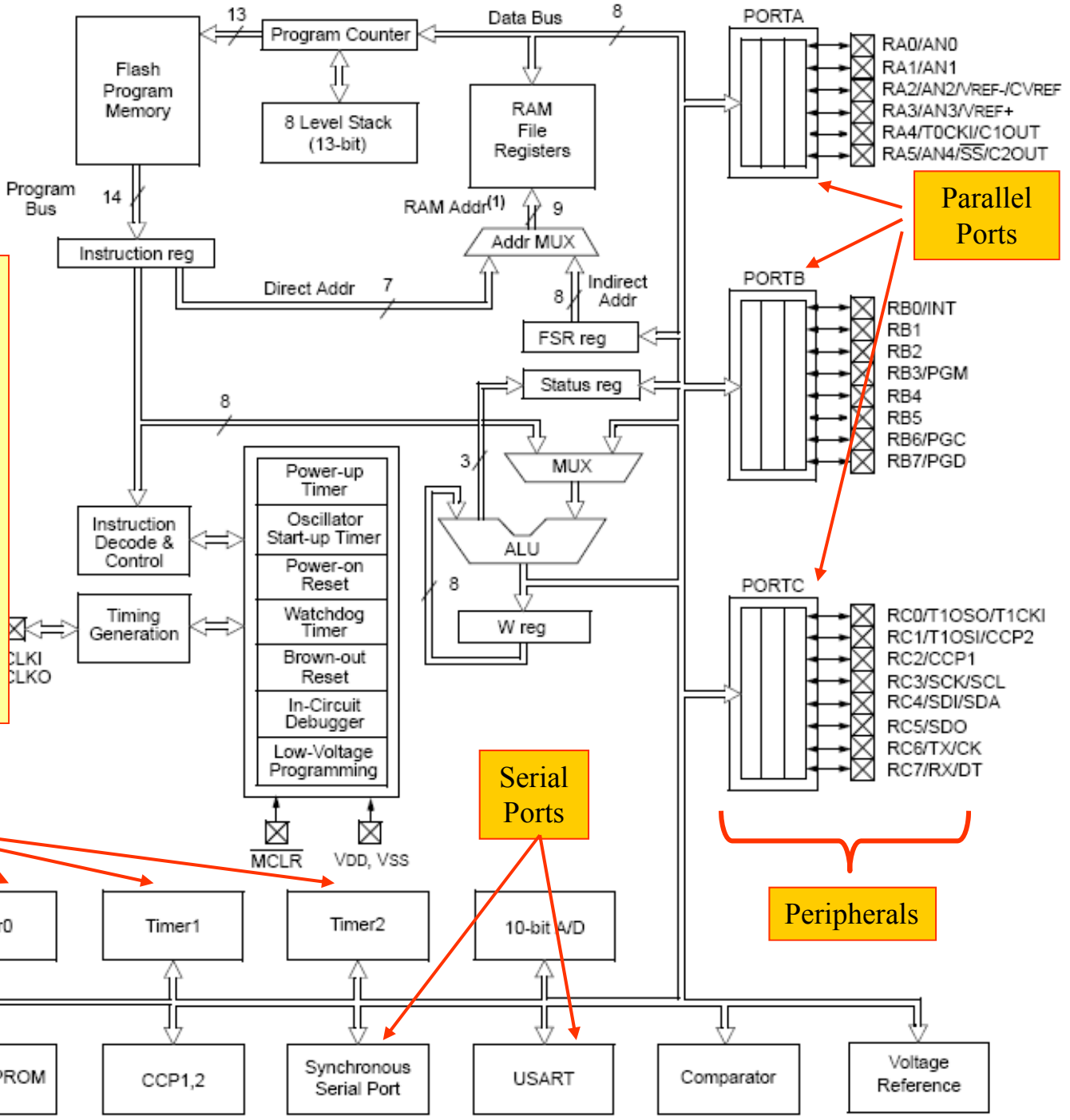
Note: For $\overline{borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.

The Memory

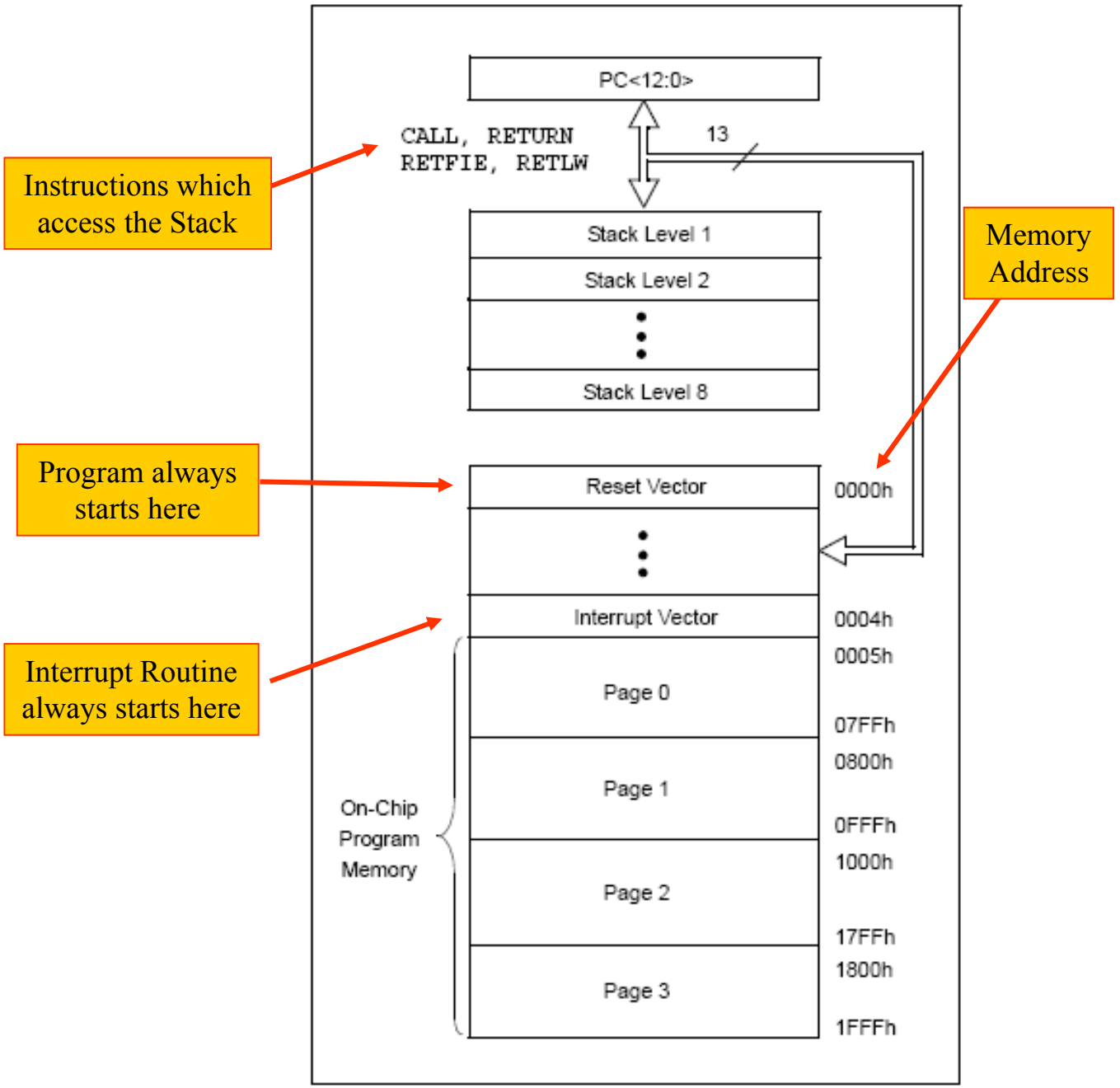

Program Memory
Data Memory
Flash Program Memory
13
Program Counter
Data Bus
8
8 Level Stack (13-bit)
RAM File Registers
Program Bus
14
RAM Addr(1)
9
Addr MUX
Instruction reg
Direct Addr
7
8
Indirect Addr
FSR reg
Status reg
8
3
MUX
Power-up Timer
Oscillator Start-up Timer
Power-on Reset
Watchdog Timer
Brown-out Reset
In-Circuit Debugger
Low-Voltage Programming
Instruction Decode & Control
ALU
8
W reg
Timing Generation
OSC1/CLKI
OSC2/CLKO
MCLR
VDD, VSS
PORTA
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
PORTB
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
PORTC
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
Timer0
Timer1
Timer2
10-bit A/D
Data EEPROM
CCP1,2
Synchronous Serial Port
USART
Comparator
Voltage Reference

The Peripherals

It is easy to see the peripherals, tho' each one will take some work to understand. Connection to the peripherals goes through the parallel port bits. Most of these bits therefore have more than one use, which begins to explain why the names against them are not simple.

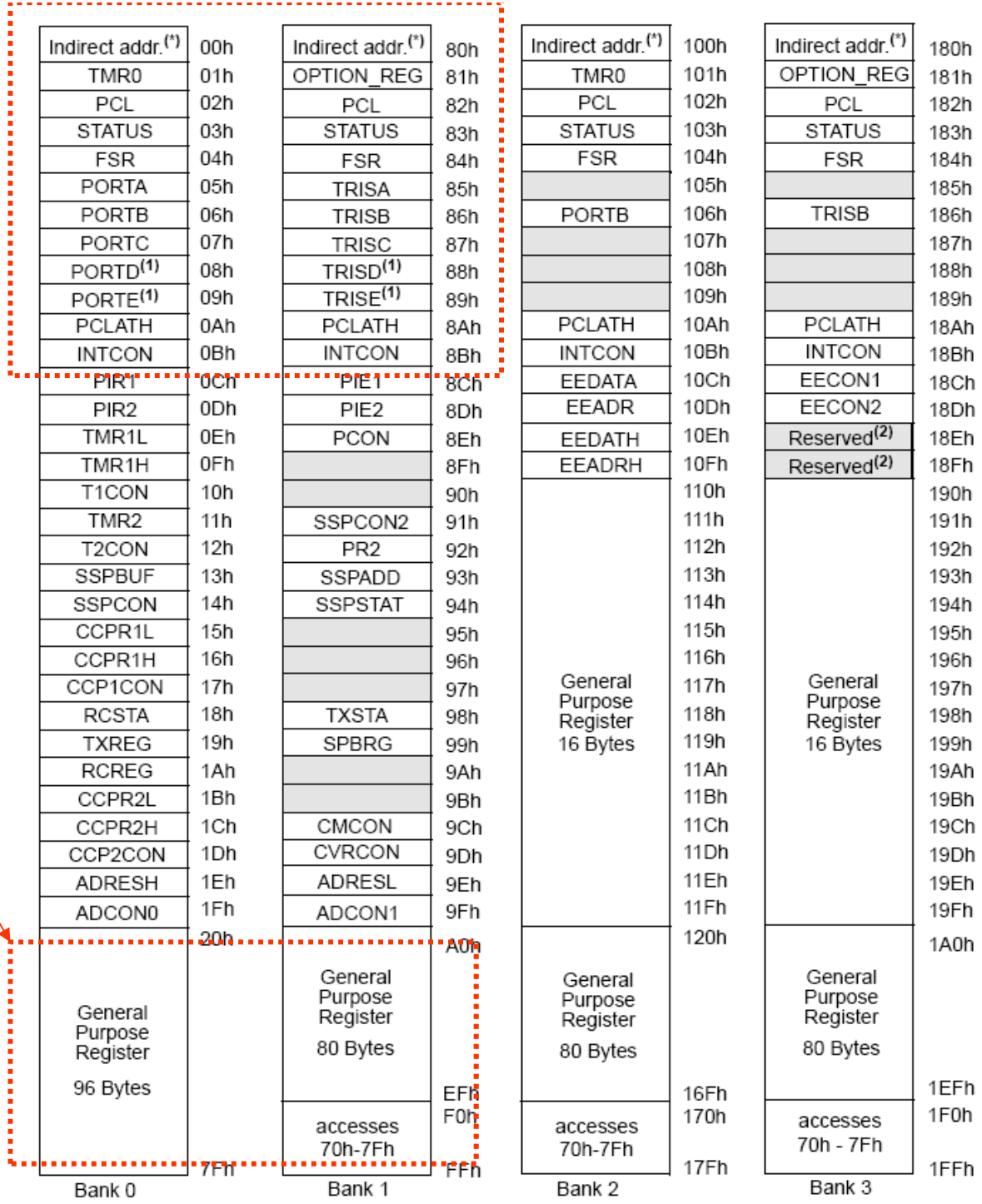


Program Memory



The Data Memory Map, including the Special Function Registers

Areas of immediate interest



Understanding the Banked Addressing

Bank 0		Bank 1		Bank 2		Bank 3	
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose	117h	General Purpose	197h

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h-1FFh)
10 = Bank 2 (100h-17Fh)
01 = Bank 1 (80h-FFh)
00 = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

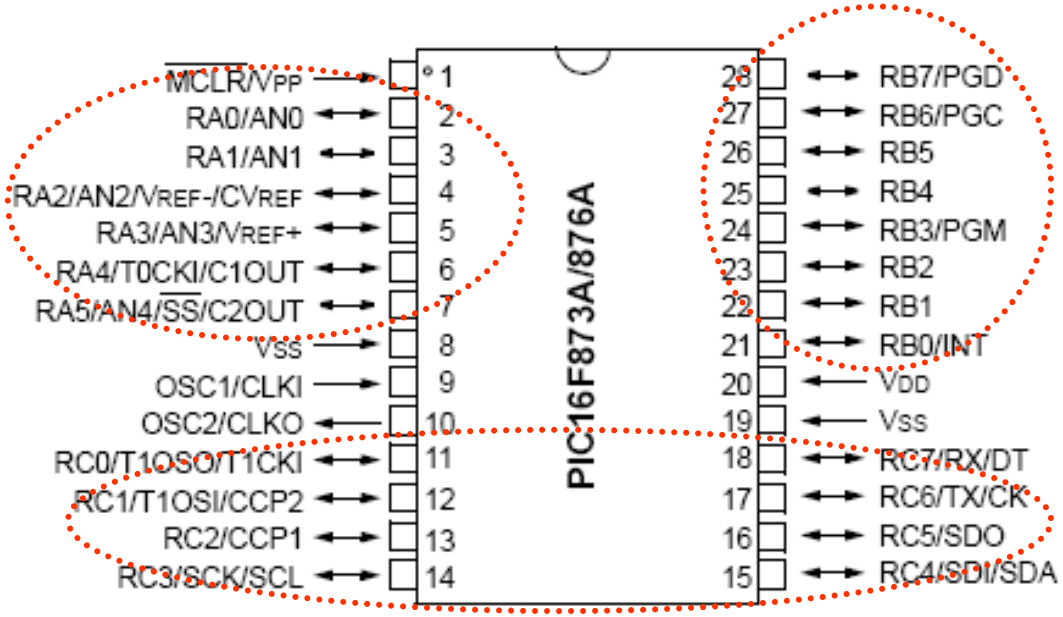
Parallel Input/Output Ports 4
SFRs for the 16F873A Parallel Ports

PORTA	05h	TRISA	85h
PORTB	06h	TRISB	86h
PORTC	07h	TRISC	87h
PORTD(1)	08h	TRISD(1)	88h
PORTE(1)	09h	TRISE(1)	89h

The SFR named PORTX holds the input/output data for the port, ie it holds all the "Data Latch" bits for that port.

The SFR named TRISX holds all the "TRIS Latch" bits for that port. The bits can be set independently, so one can be input while another output. They cannot be both at the same time.

Port A, B and C pin connections can easily be found on the pin diagram. Note that many pins have several functions, as indicated on the diagram.



The 16F873A Configuration Word

R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
CP	—	DEBUG	WRT1	WRT0	CPD	LVP	BOREN	—	—	$\overline{\text{PWRTEN}}$	WDTEN	Fosc1	Fosc0
bit 13													bit0

bit 13 **CP:** Flash Program Memory Code Protection bit
1 = Code protection off
0 = All program memory code-protected

bit 12 **Unimplemented:** Read as '1'

bit 11 **DEBUG:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger

bit 10-9 **WRT1:WRT0** Flash Program Memory Write Enable bits
For PIC16F876A/877A:
11 = Write protection off; all program memory may be written to by EECON control
10 = 0000h to 00FFh write-protected; 0100h to 1FFFh may be written to by EECON control
01 = 0000h to 07FFh write-protected; 0800h to 1FFFh may be written to by EECON control
00 = 0000h to 0FFFh write-protected; 1000h to 1FFFh may be written to by EECON control
For PIC16F873A/874A:
11 = Write protection off; all program memory may be written to by EECON control
10 = 0000h to 00FFh write-protected; 0100h to 0FFFh may be written to by EECON control
01 = 0000h to 03FFh write-protected; 0400h to 0FFFh may be written to by EECON control
00 = 0000h to 07FFh write-protected; 0800h to 0FFFh may be written to by EECON control

bit 8 **CPD:** Data EEPROM Memory Code Protection bit
1 = Data EEPROM code protection off
0 = Data EEPROM code-protected

bit 7 **LVP**: Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit
1 = RB3/PGM pin has PGM function; low-voltage programming enabled
0 = RB3 is digital I/O, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 6 **BOREN**: Brown-out Reset Enable bit
1 = BOR enabled
0 = BOR disabled

bit 5-4 **Unimplemented:** Read as '1'

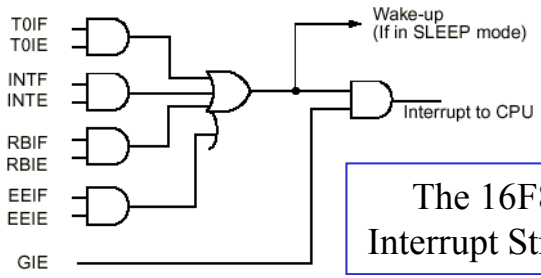
bit 3 **$\overline{\text{PWRTEN}}$**: Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 2 **WDTEN**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

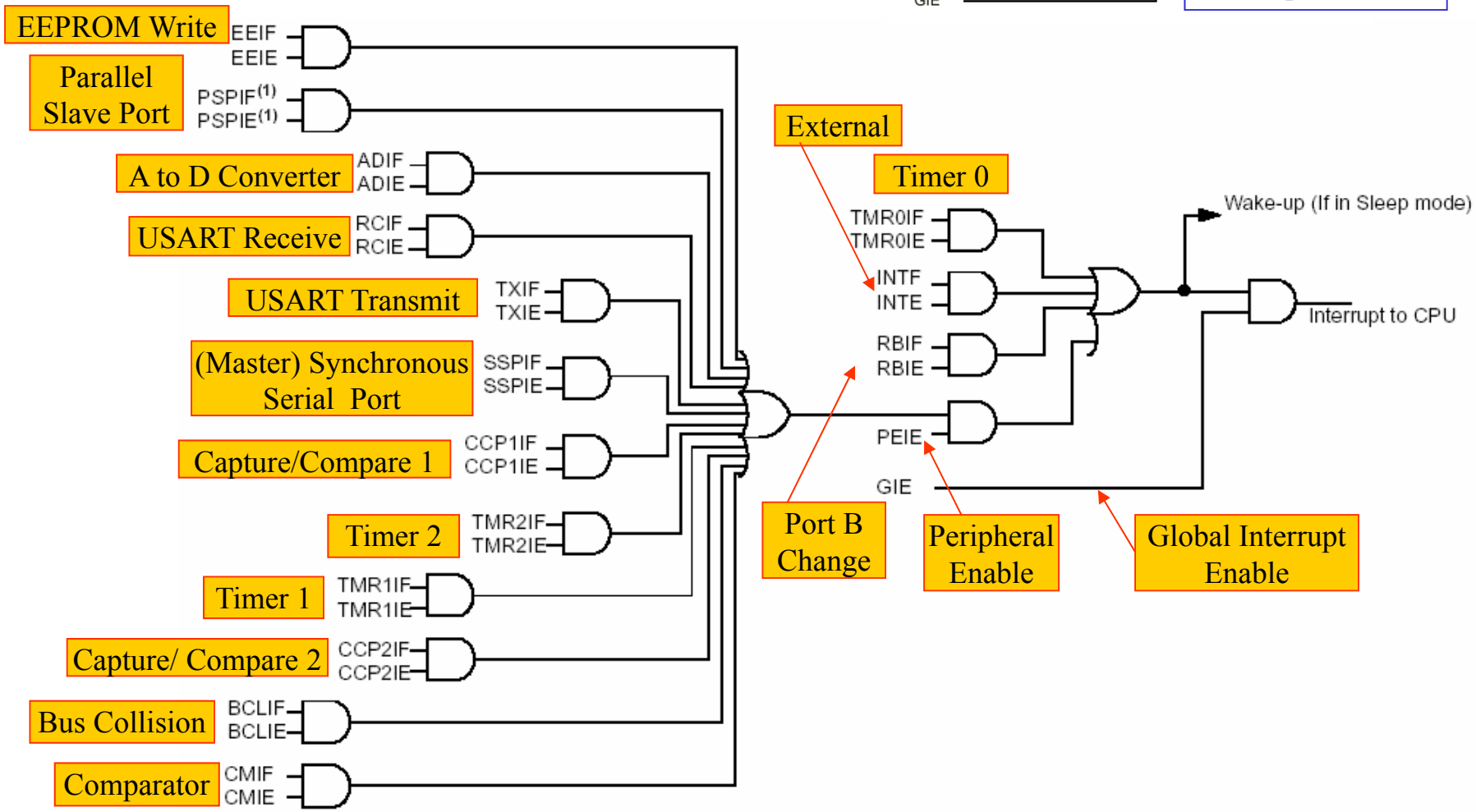
bit 1-0 **Fosc1:Fosc0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

16F873A Interrupt Structure

Here the minimalist structure of the 16F84A is seen extended to make something much larger. All interrupts are routed ultimately through to the single interrupt vector, seen in the program memory map.



The 16F84A Interrupt Structure



Note 1: PSP interrupt is implemented only on PIC16F874A/877A devices.

16F873A INTCON Register

With 15 interrupt sources, the INTCON register can only hold a small proportion of all flags and enable bits. PEIE is the *Peripheral Interrupt Enable Bit*, acting as a subsidiary Global Enable, to all those interrupt sources upstream of the AND gate it drives. It must be set to 1 if *any* of the "upstream" interrupts are to be used.

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

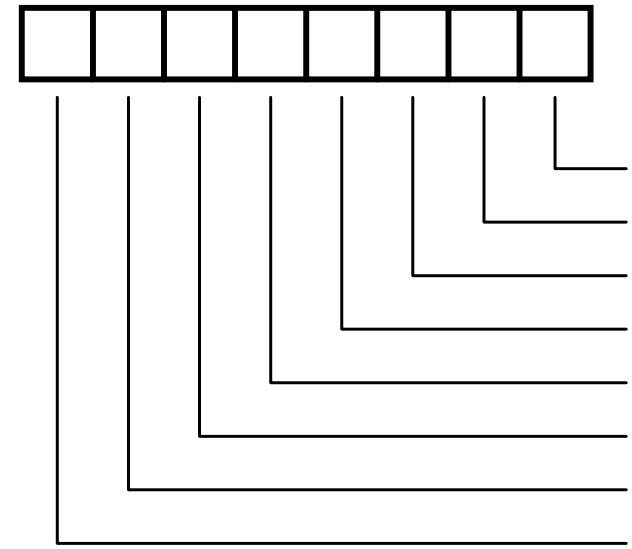
bit 1 **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

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

16F873A PIE1/PIR1 (Peripheral Interrupt Enable/ Peripheral Interrupt Request) Registers

INTCON	0Bh		INTCON	8Bh
PIR1	0Ch		PIE1	8Ch
PIR2	0Dh		PIE2	8Dh

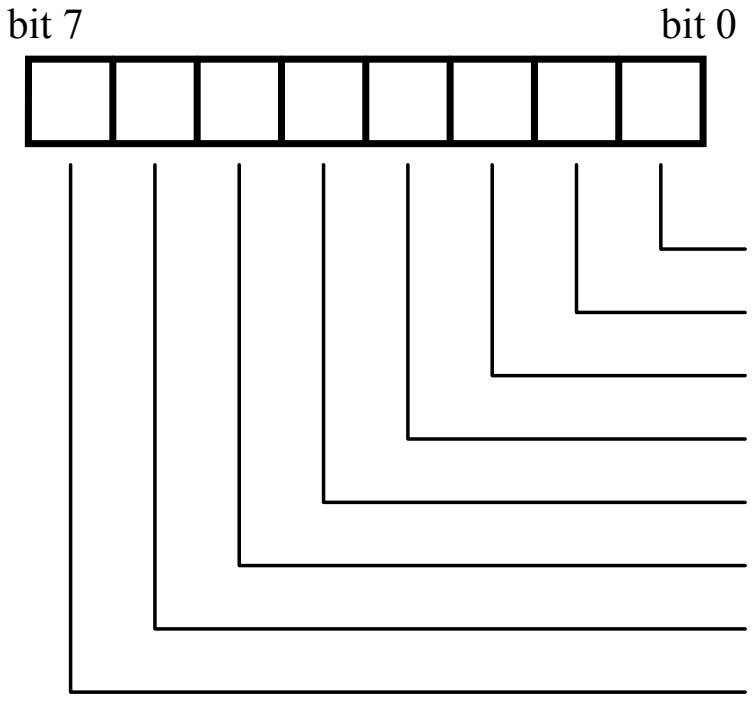
To augment the INTCON register, *four* SFRs are added - PIE1 and PIE2 for enable bits and located in memory bank 1; and PIR1 and PIR2 in memory bank 0, holding the flag bits. PIE1 and PIR1 each follow the same pattern, as do PIE2 and PIR2. All active bits are reset to 0 on any form of power up.



	PIE1 (Enable Bits)	PIR1 (Flag Bits)
Timer 1 Overflow	**TMR1IE**	**TMR1IF***
Timer 2 Overflow	**TMR2IE**	**TMR2IF***
Capture Compare 1	**CCP1IE**	**CCP1IF***
Synchronous Serial Port	**SSPIE**	**SSPIF***
USART Transmit	**TXIE**	**TXIF**
USART Receive	**RCIE**	**RCIF**
Analog to Digital Converter	**ADIE**	**ADIF**
Parallel Slave Port Read/Write**	**PSPIE**	**PSPIF***

* Must be cleared in software
** 16F874/7 only. Reserved in 16F873/6

16F873A PIE2/PIR2 Registers



	PIE2 (Enable Bits)	PIR2 (Flag Bits)
CCP2	**CCP2IE**	**CCP2IF**
Unimplemented	read as 0	read as 0
Unimplemented	read as 0	read as 0
Bus Collision	**BCLIE**	**BCLIF**
EEPROM Write	**EEIE**	**EEIF***
Unimplemented	read as 0	read as 0
Comparator	**CMIE**	**CMIF***
Unimplemented	read as 0	read as 0

* Must be cleared in software

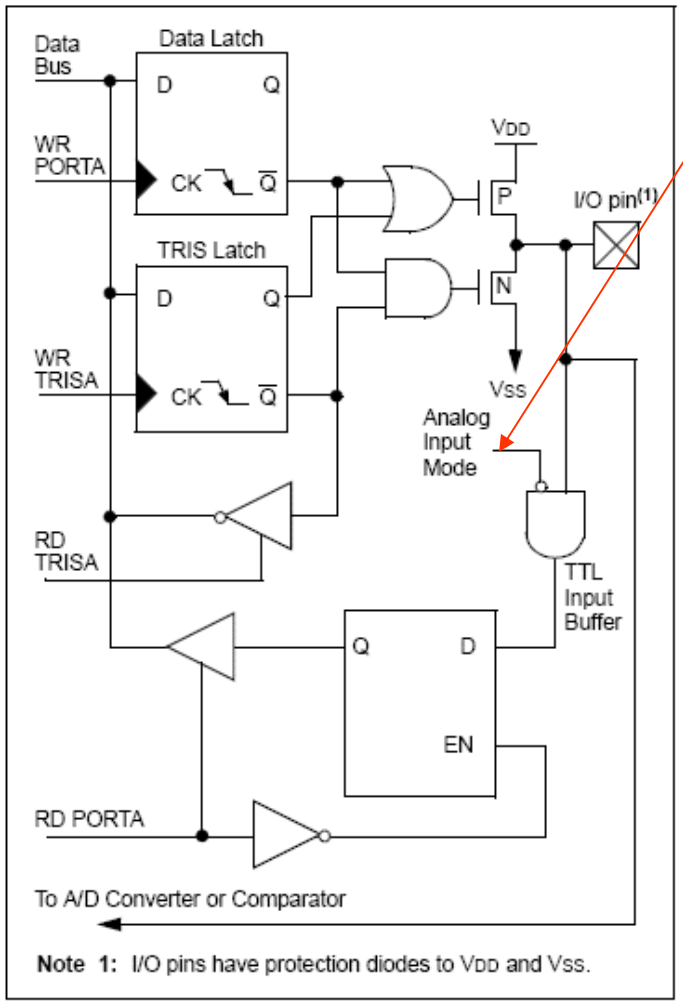
An Interrupt Example

Program Example 10.3 uses 2 interrupts. Work out from this program fragment how they are applied.

```
          bsf                 pie1,sspie          ;enable I2C interrupt
...
          bcf                 intcon,rbif         ;clear pending interrupts
          bcf                 pir1,sspif
          bsf                 intcon,rbie
          bsf                 intcon,peie
          bsf                 intcon,gie
...
loop      goto      loop                ;await keypad and I2C interrupts
...
;*************************************************************
;This is ISR, caused by keypad or I2C address match.
;Does not context save, as all action is in ISRs.
;*************************************************************
Interrupt_SR  btfsc intcon, rbif        ;is it keypad interrupt?
          goto      kpad_ISR
;Here if interrupt is I2C, either address match (Ack sent automatically)
;OR further received byte has been detected.
;check whether this byte was address or data
          bsf       status,rp0
          btfsc     sspstat,d_a
          goto      ISR1                ;go if word was data
...
          bcf       pir1,sspif          ;clear interrupt bit, and end ISR
          retfie
;Here if data byte has been detected, word is hence already in buffer.
ISR1      call      dig_pntr_set ;sort display pointer
...
          bcf       pir1,sspif          ;clear interrupt bit
          retfie
;here if sending I2C word. Send byte held in kpad_char
Send_I2C bcf        status,rp0
...
          bcf       pir1,sspif          ;clear interrupt bit
          retfie
```

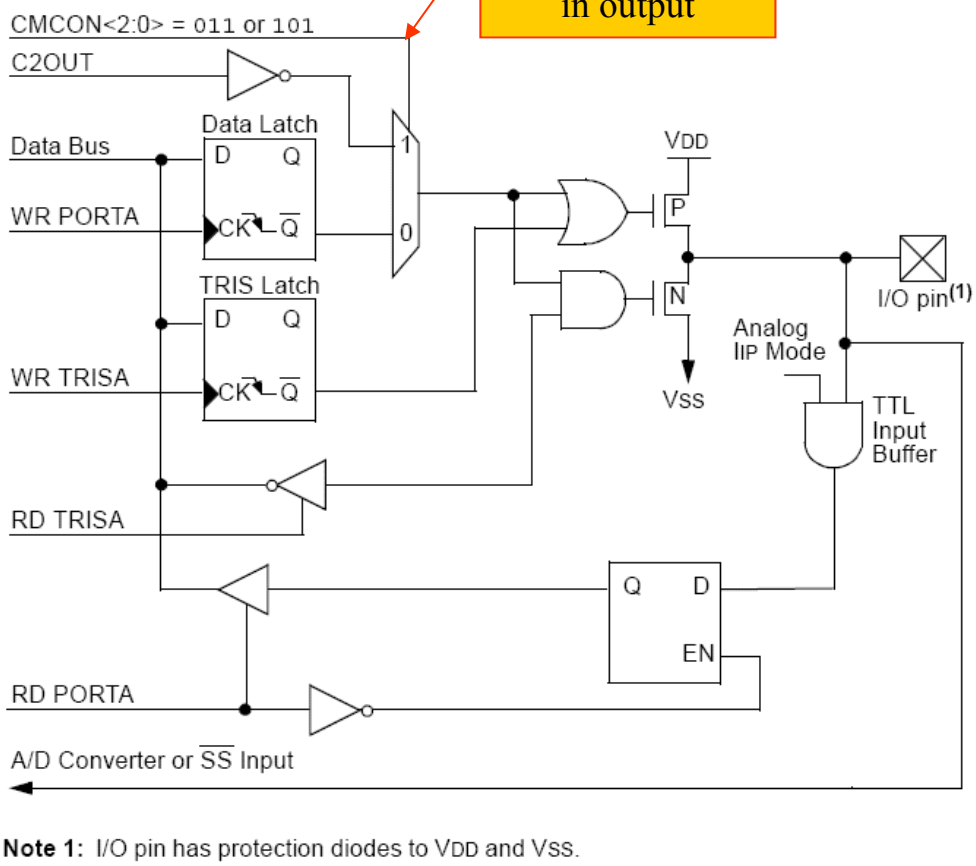
PIC 16F873A Port A Pin Driver Circuits

The port is shared between general-purpose bidirectional digital data, the Analog to Digital Converter (ADC) module, and comparators. On power-up the port bits are set as analog inputs. To use the port for digital purposes the ADCON0 register, must be set appropriately. Timer 0 input is on bit 4.



Peripheral can disable port input

multiplexer selects data source, when in output

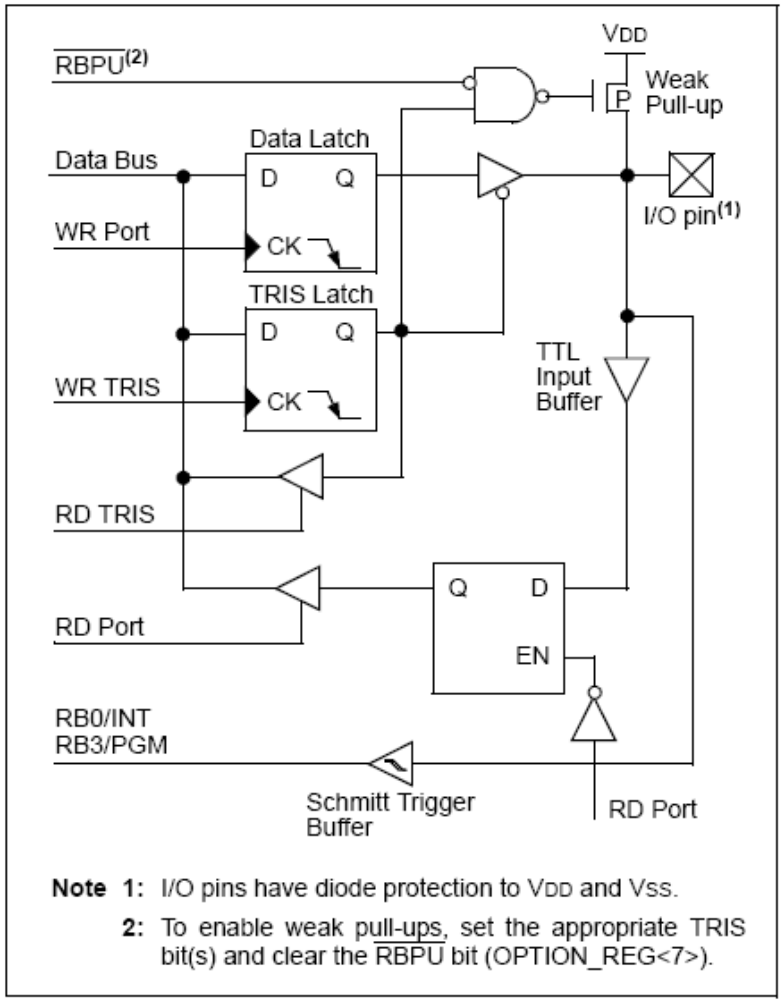


Pins RA0, 1, 2, and 3

Pin RA5

Port B Pin Driver Circuits

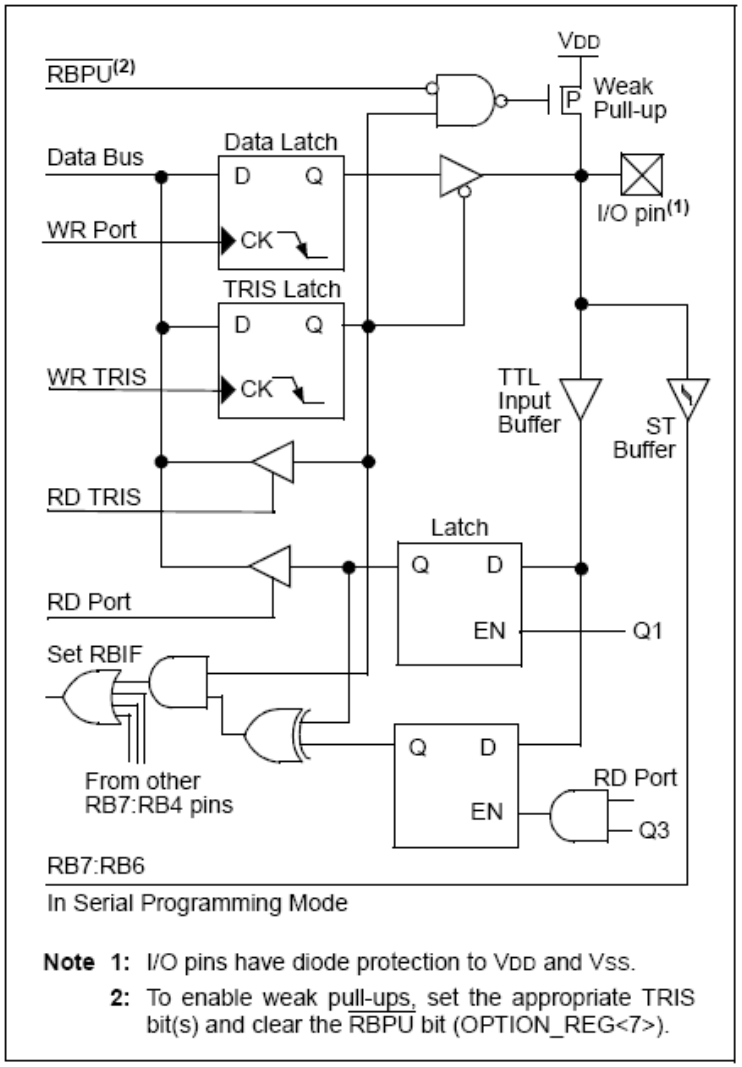
A straightforward port, except that bits 3, 6 and 7 are used for In Circuit Serial Programming (ICSP). If ICSP is used, the designer must either not use these pins for any other purpose, or use them in such a way that they are still available ICSP.



Note 1: I/O pins have diode protection to VDD and Vss.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

RB3 to RB0



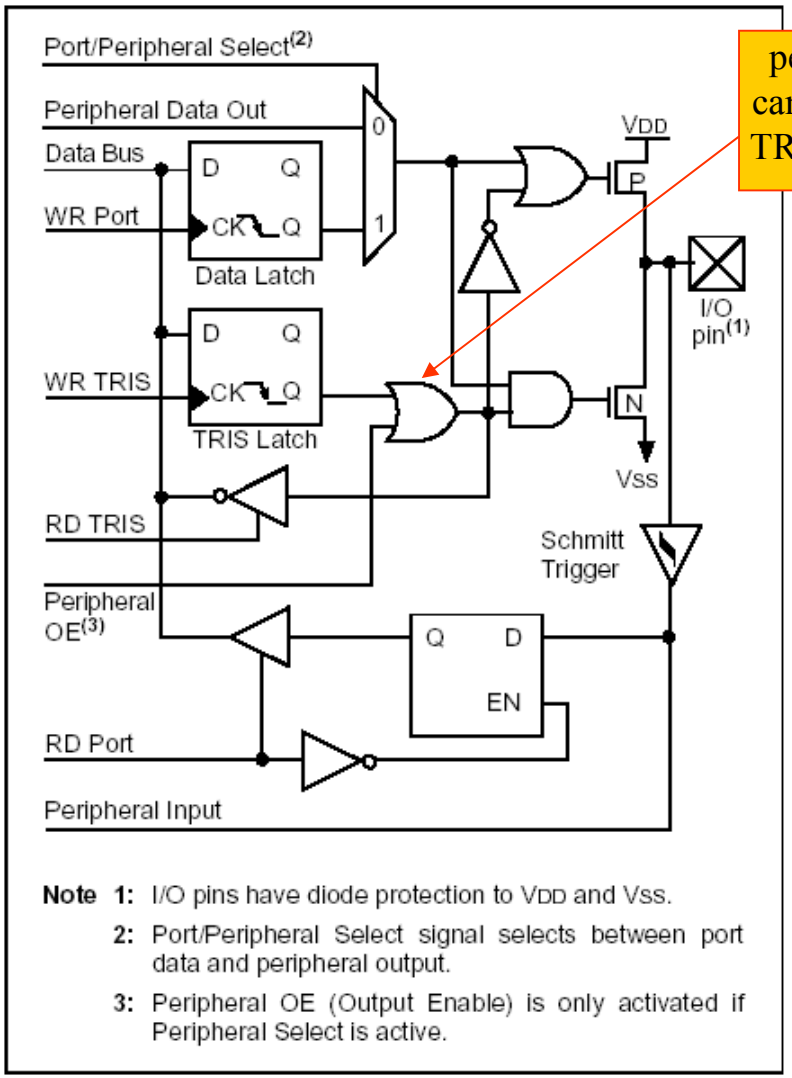
Note 1: I/O pins have diode protection to VDD and Vss.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

RB7 to RB4

Port C Pin Driver Circuits

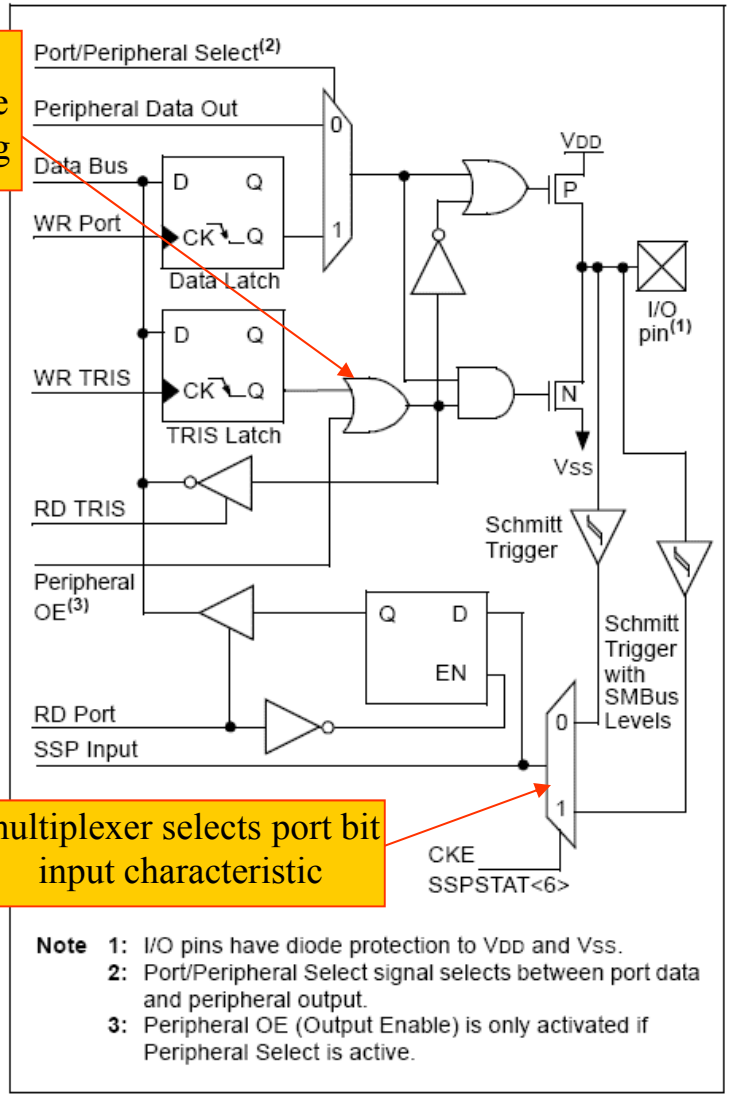
The most complex of the 16F873A ports, all inputs have Schmitt trigger inputs. Aside from general-purpose i/o, Port C pins are shared with some of the more complex microcontroller peripherals, including serial communication.



Note 1: I/O pins have diode protection to VDD and Vss.
2: Port/Peripheral Select signal selects between port data and peripheral output.
3: Peripheral OE (Output Enable) is only activated if Peripheral Select is active.

RC4, 3 Pins

peripheral can override TRIS setting

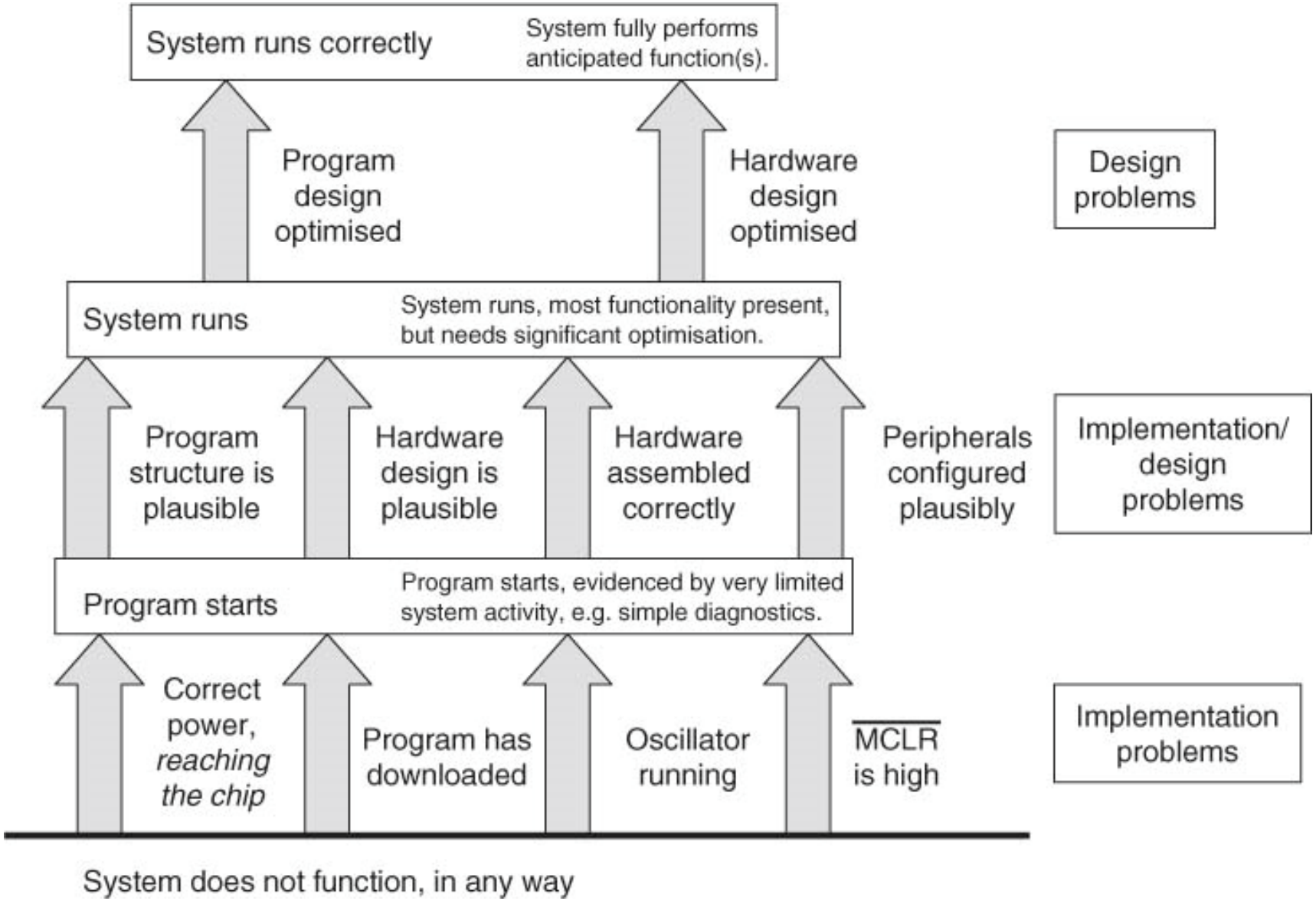


multiplexer selects port bit input characteristic

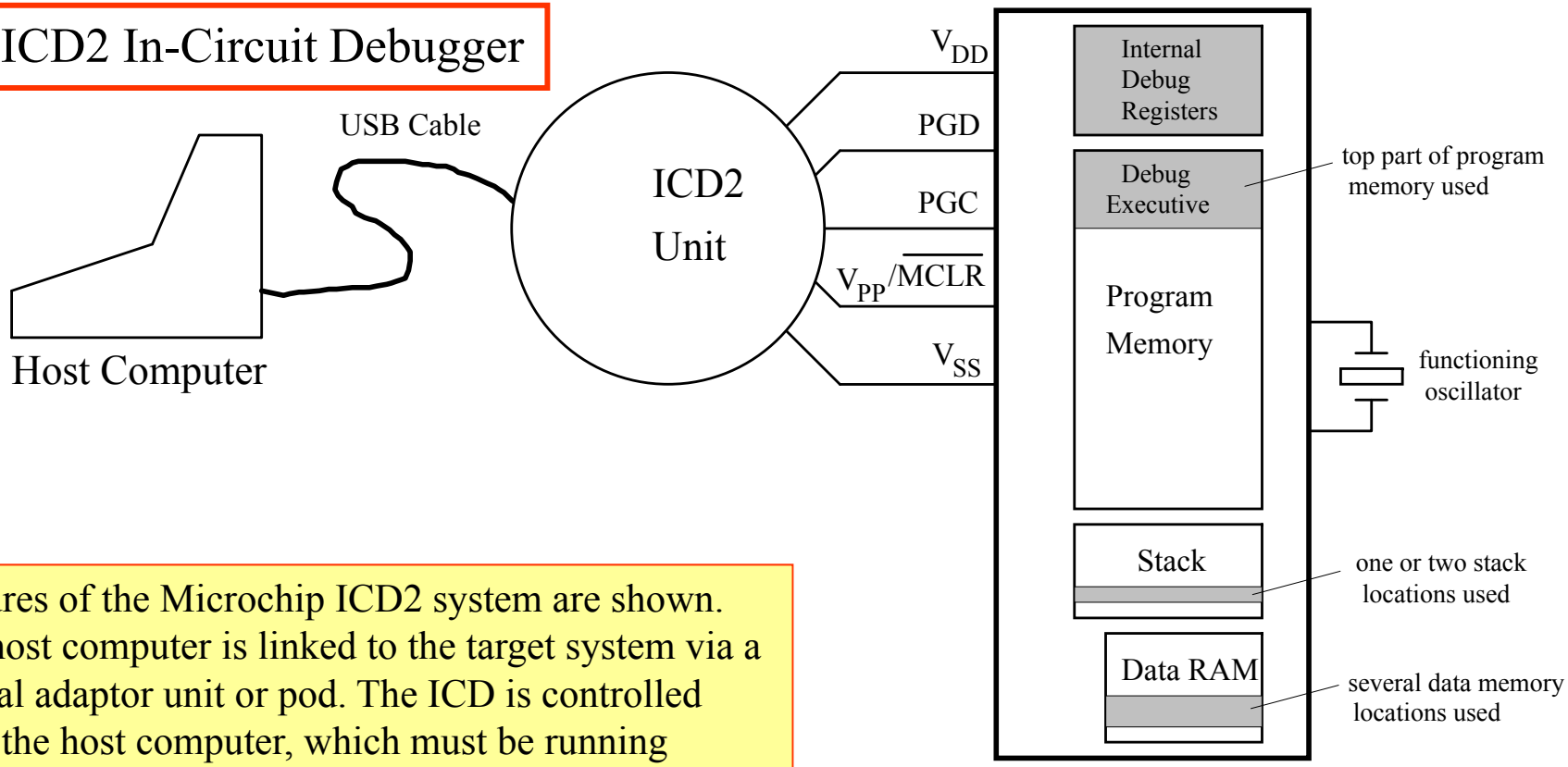
Note 1: I/O pins have diode protection to VDD and Vss.
2: Port/Peripheral Select signal selects between port data and peripheral output.
3: Peripheral OE (Output Enable) is only activated if Peripheral Select is active.

RC7 to RC5, RC2 to RC0 Pins

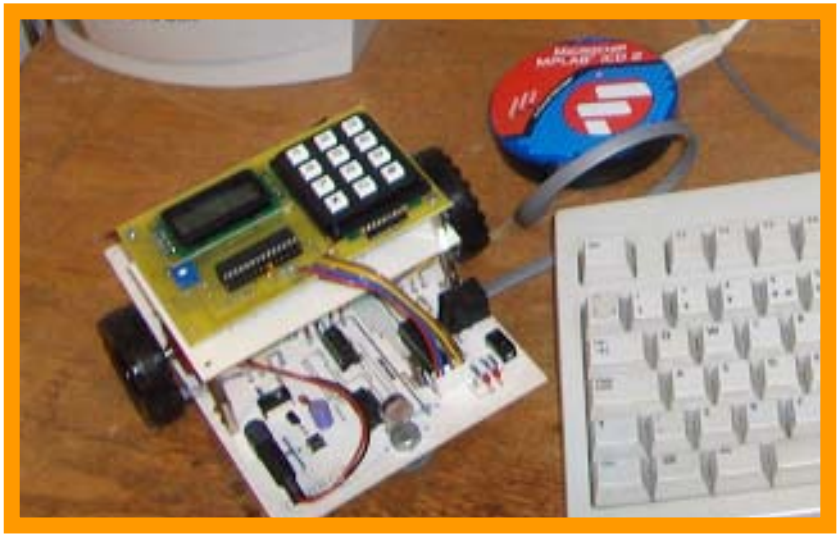
Test and Commission: Layers of Dependence in an Embedded System



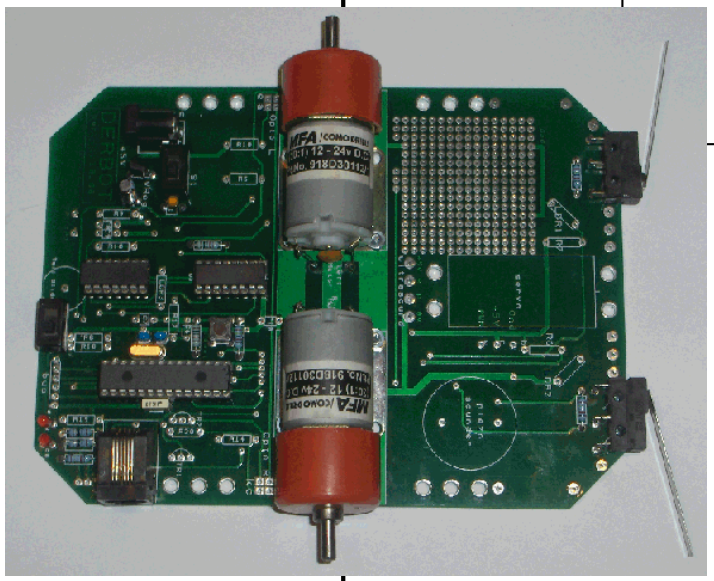
The ICD2 In-Circuit Debugger

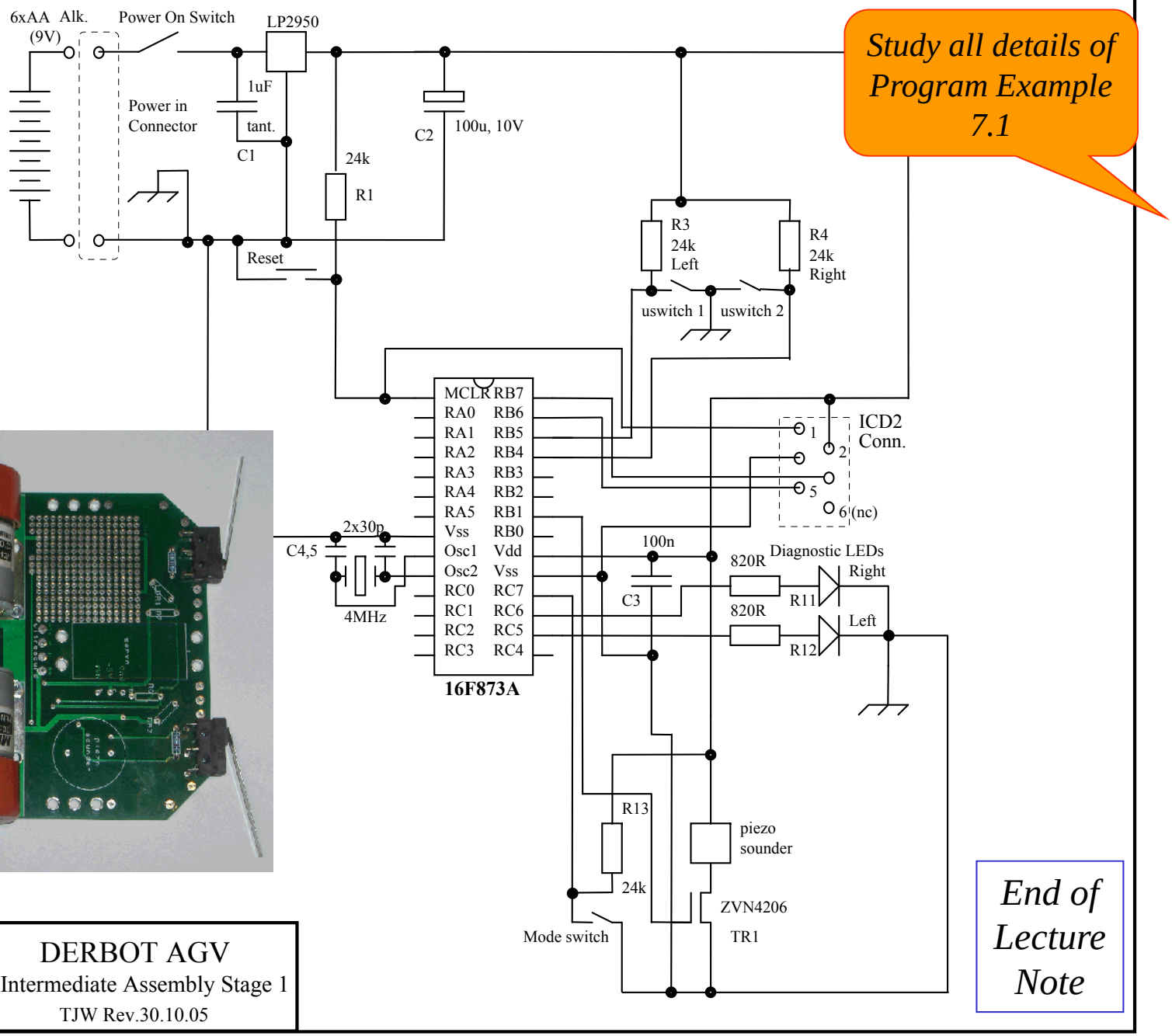


Features of the Microchip ICD2 system are shown. The host computer is linked to the target system via a special adaptor unit or pod. The ICD is controlled from the host computer, which must be running MPLAB. The ICD2 pod links to the host computer via a USB cable, and connects to the target system via another cable, having five interconnections. These are shown in the diagram, and are the same as those for ICSP, except that the low voltage programming pin, bit 3 of Port B is not used. The internal microcontroller resources needed by the ICD are also shown, and include elements of program memory, data memory, and the stack. The ICD2 can be used in two distinct modes – debugger and programmer.



Example design – the Derbot AGV





Study all details of Program Example 7.1

DERBOT AGV
Intermediate Assembly Stage 1
TJW Rev.30.10.05

End of Lecture Note