EECE416: Microcomputer Fundamentals and Design

PIC - Introduction

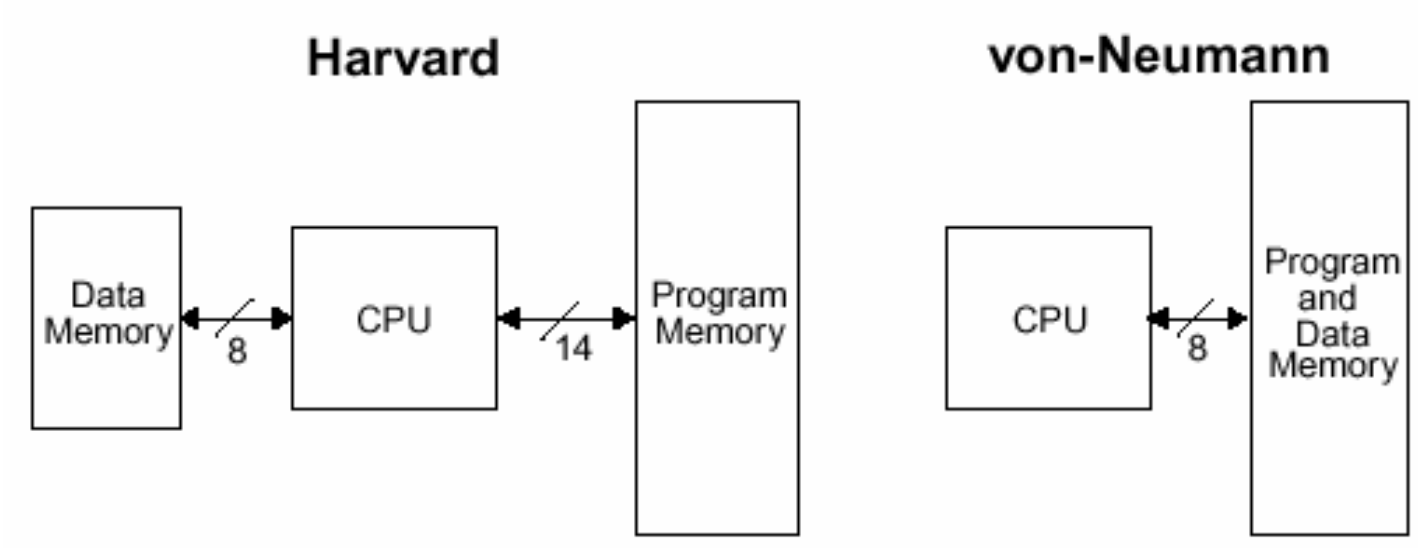
PIC16F877

Dr. Charles J. Kim

Howard University

Peripheral Interface Controller

- PIC: Peripheral Interface Controller
- Microchip Technology (www.microchip.com)
- Harvard Architecture

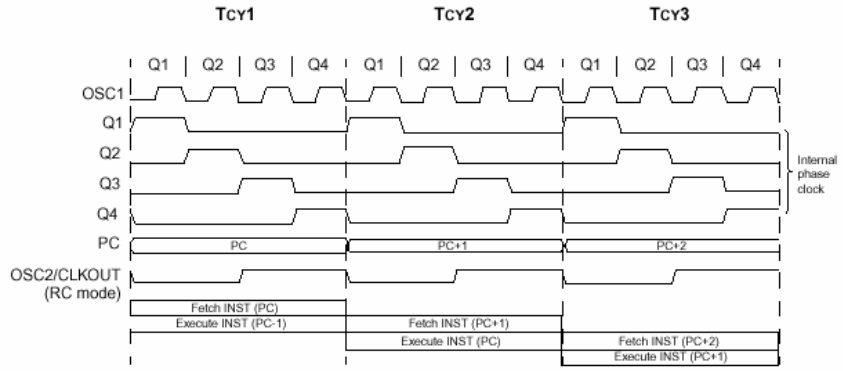


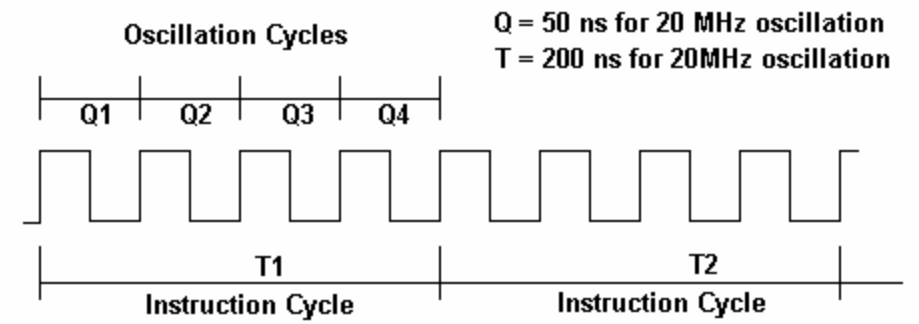
PIC- continued

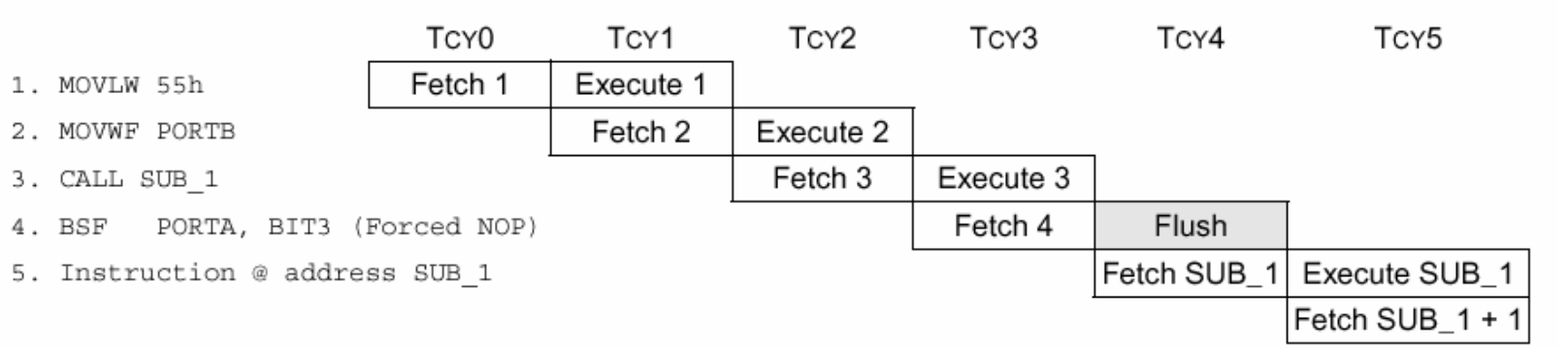
- Origin: Harvard Architecture for DAPRA Project
  - Beaten by Princeton (Single memory)
  - Picked by Signetics 8x300
  - PIC for General Instruments
  - Compensation for poor I/O
  - GI spun off into Arizona Microchip Technology (1985)-→Microchip Technology

PIC -continued

- A Large Register Set: 368 Bytes + W
- RISC Architecture--pipelining
- 35 fixed length (14-bit) single-cycle instructions

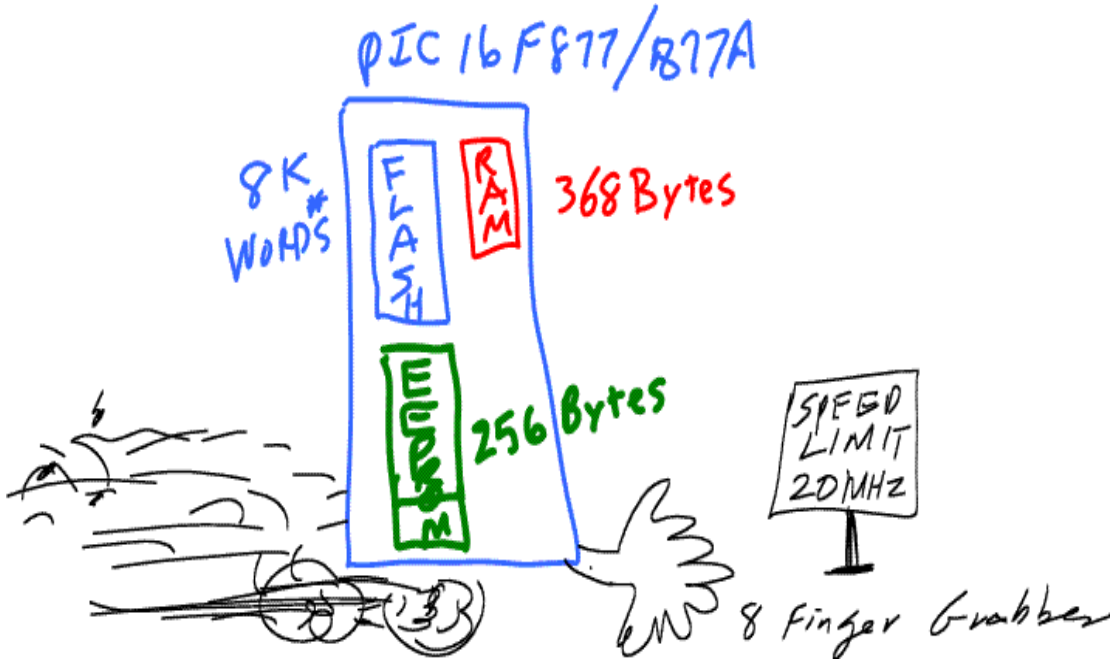






PIC-continued

- 8-bit machine
- Operating Speed: Up to 20 MHz
- 8K 14-bit Words FLASH Memory (for Program)
- 368 Bytes RAM (for Data)
- 256 Bytes EEPROM (for Data)
- Power Saving Mode



PIC -continued

- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 mA typical @ 3V, 32 kHz
  - < 1 mA typical standby current
- Wide Operating Voltage: 2.0 – 5.0 V
- Timers
  - Timer0: 8-bit timer/counter with 8-bit prescaler
  - Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
  - Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler

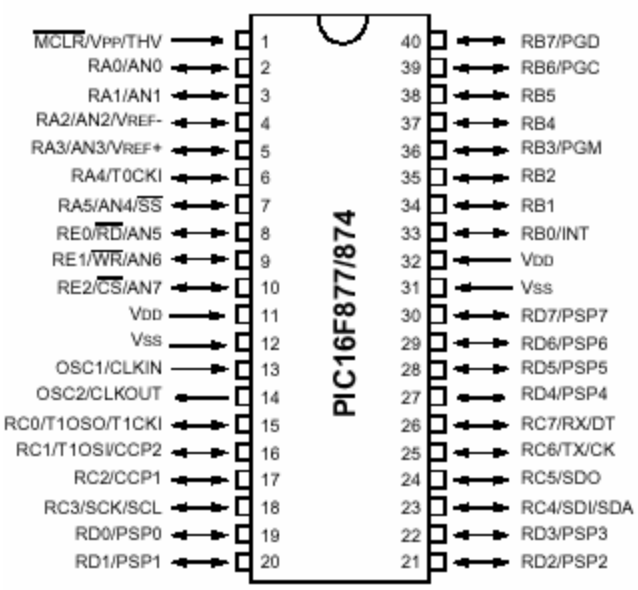
PIC -continued

- Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel A-to-D Converter
- I^2C (Inter IC)Bus
- USART for Serial Communication
- 5 I/O Ports: A, B, C, D, and E

16F87x Family

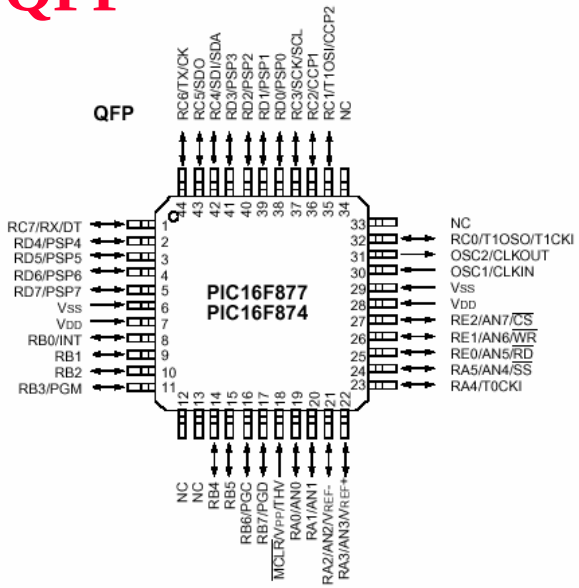
Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

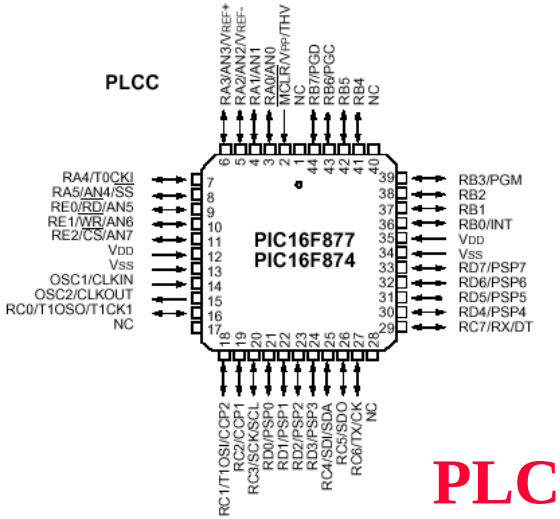
PIN and PACKAGE



DIP

QFP





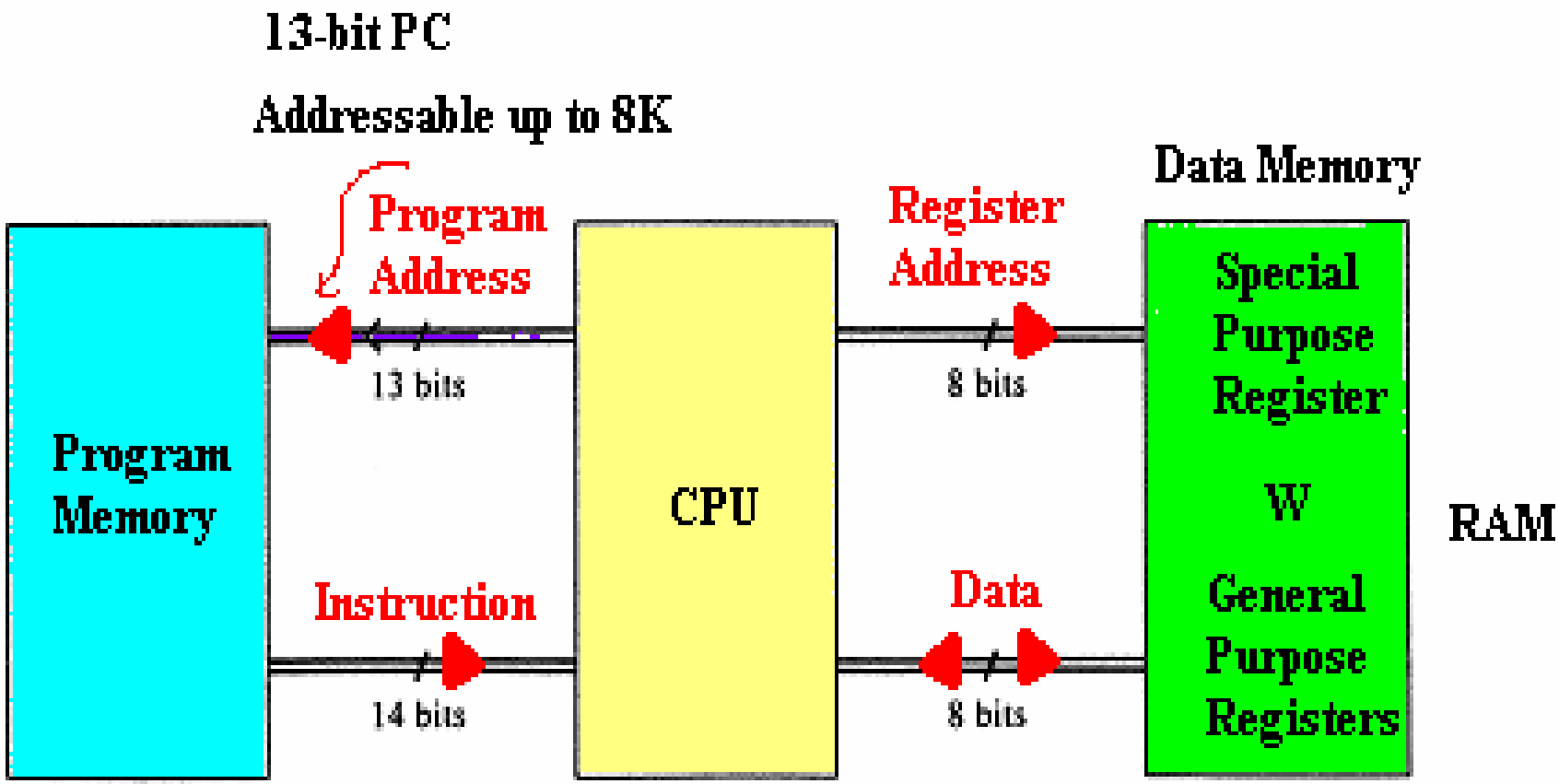
PLCC

PIN

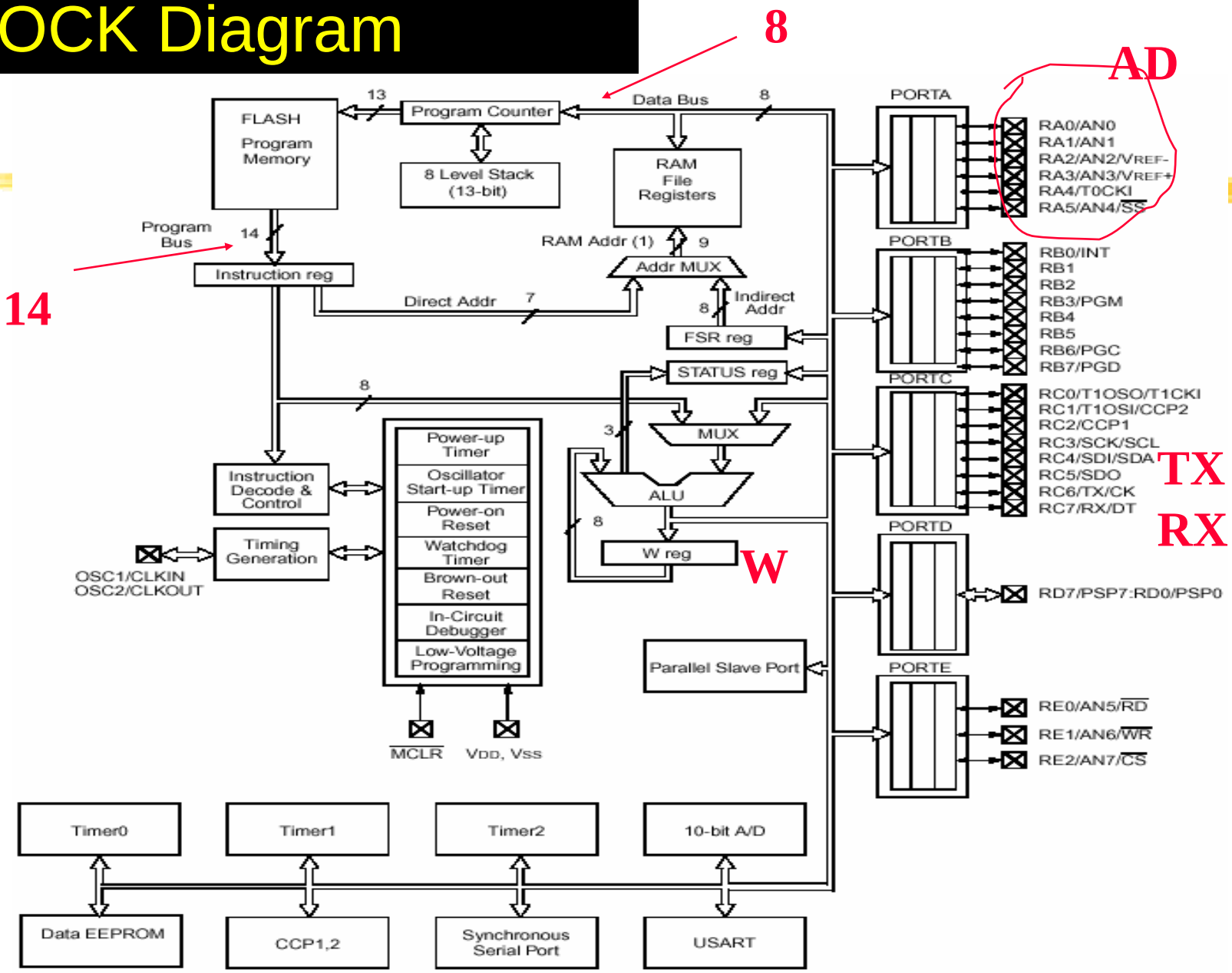
Pin Name	DIP Pin#	I/O/P Type
OSC1/CLKIN	13	I
OSC2/CLKOUT	14	O
$\overline{MCLR}$/VPP	1	I/P
RA0/AN0	2	I/O
RA1/AN1	3	I/O
RA2/AN2/VREF-	4	I/O
RA3/AN3/VREF+	5	I/O
RA4/T0CKI	6	I/O
RA5/$\overline{SS}$/AN4	7	I/O
RB0/INT	33	I/O
RB1	34	I/O
RB2	35	I/O
RB3/PGM	36	I/O
RB4	37	I/O
RB5	38	I/O
RB6/PGC	39	I/O
RB7/PGD	40	I/O

Pin Name	DIP Pin#	I/O/P Type
RC0/T1OSO/T1CKI	15	I/O
RC1/T1OSI/CCP2	16	I/O
RC2/CCP1	17	I/O
RC3/SCK/SCL	18	I/O
RC4/SDI/SDA	23	I/O
RC5/SDO	24	I/O
RC6/TX/CK	25	I/O
RC7/RX/DT	26	I/O
RD0/PSP0	19	I/O
RD1/PSP1	20	I/O
RD2/PSP2	21	I/O
RD3/PSP3	22	I/O
RD4/PSP4	27	I/O
RD5/PSP5	28	I/O
RD6/PSP6	29	I/O
RD7/PSP7	30	I/O
RE0/$\overline{RD}$/AN5	8	I/O
RE1/$\overline{WR}$/AN6	9	I/O
RE2/$\overline{CS}$/AN7	10	I/O
VSS	12,31	P
VDD	11,32	P

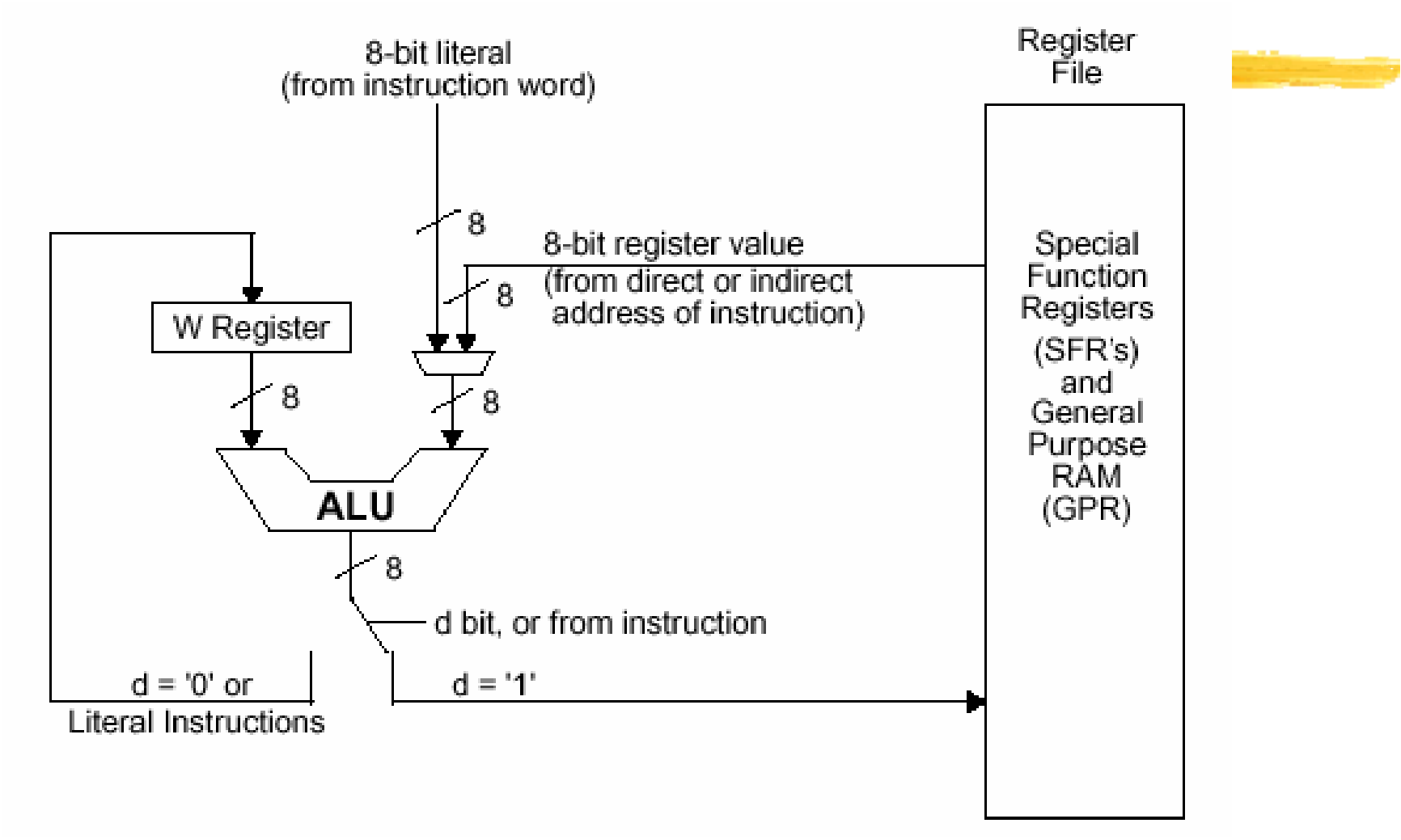
F877 Architecture



BLOCK Diagram

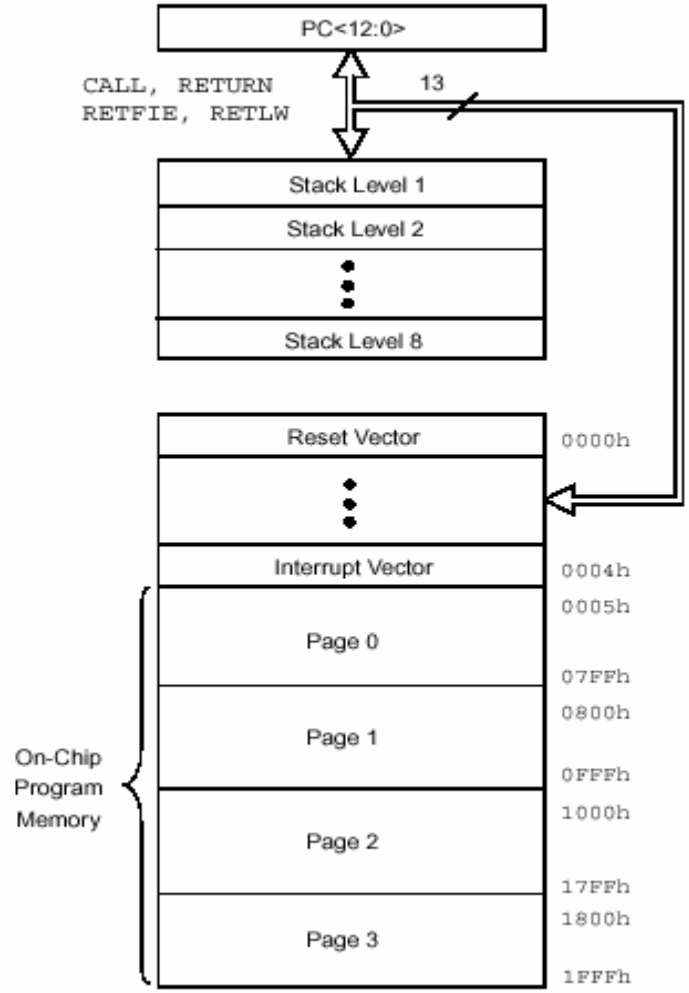


ALU and W register



PROGRAM MEMORY

- 13-bit PC
- Addressable up to 8Kx14-bit
- FLASH Memory
- Reset Vector: $0000
- Interrupt: $0004

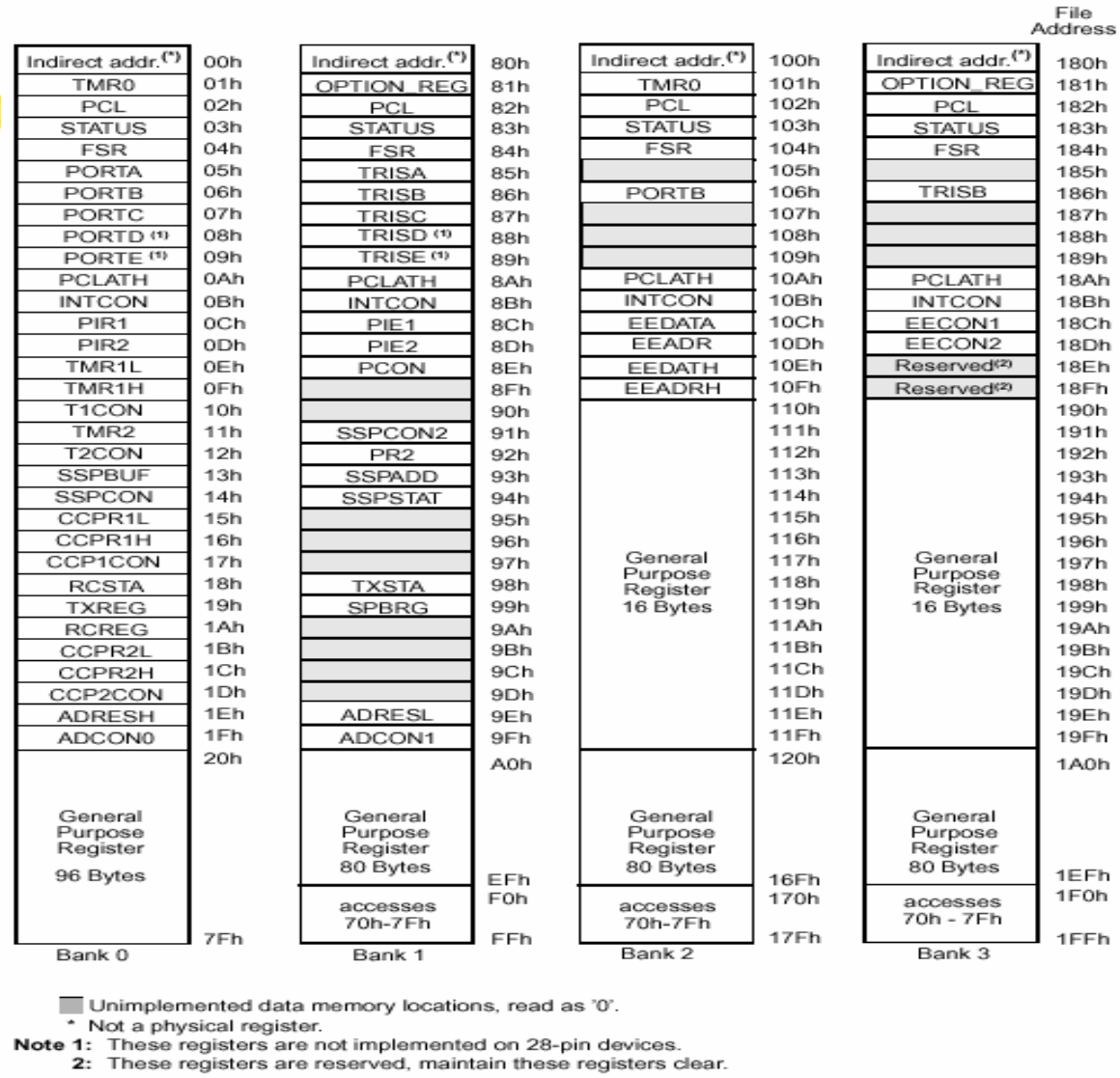


DATA memory ("Register File")

- Partitioned into 4 Banks (or Pages): 0-3
- Each Bank: 128 Bytes
- Upper Locations: GPR (General Purpose) Reg.-- RAM
- Lower Location: SFR (Special Function) Reg.
- BANK SELECTION:
  - Status Register
  - RP1: Status<6>
  - RP0: Status<5>

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Register File Map



Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on 28-pin devices.
2: These registers are reserved, maintain these registers clear.

Special Function Registers (bank 0)

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 (2)
Bank 0											
00h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h(4)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h(5)	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h(5)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(3)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	(6)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	-r-0 0--0
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Special Function Registers (bank 1)

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 (2)
Bank 1											
80h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(4)	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
84h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h(5)	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h(5)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
8Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	PSPIE(3)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	(6)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	BOR	---- --qq	---- --uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	0000 0000
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Special Function Registers(bank 2 & 3)

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 (2)
Bank 2											
100h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h(4)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high byte						xxxx xxxx	uuuu uuuu
10Fh	EEADRH	—	—	—	EEPROM address register high byte					xxxx xxxx	uuuu uuuu
Bank 3											
180h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h(4)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

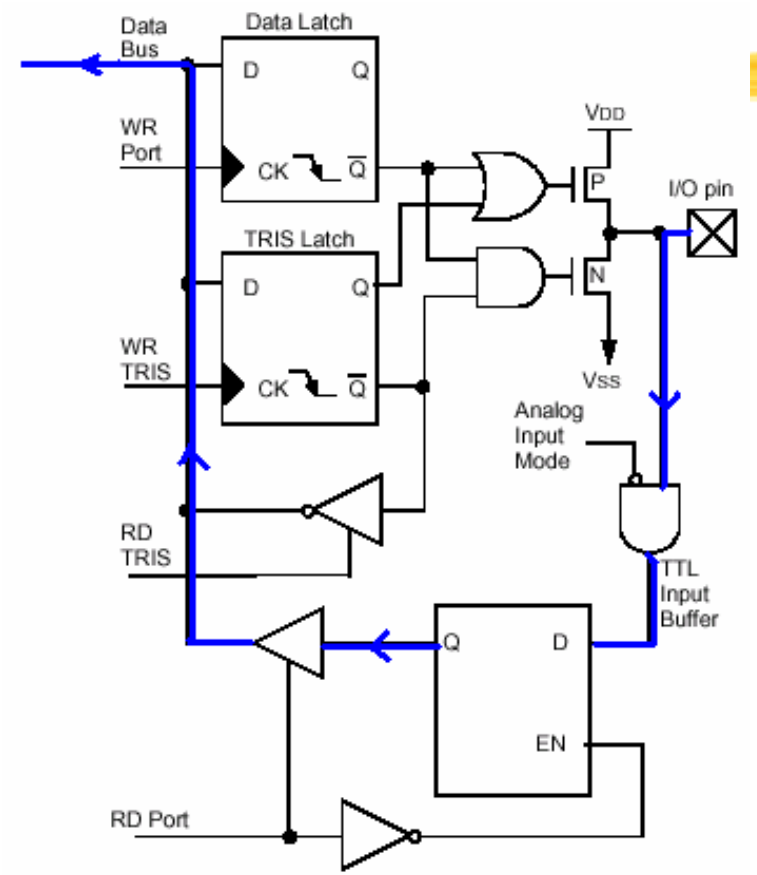
3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

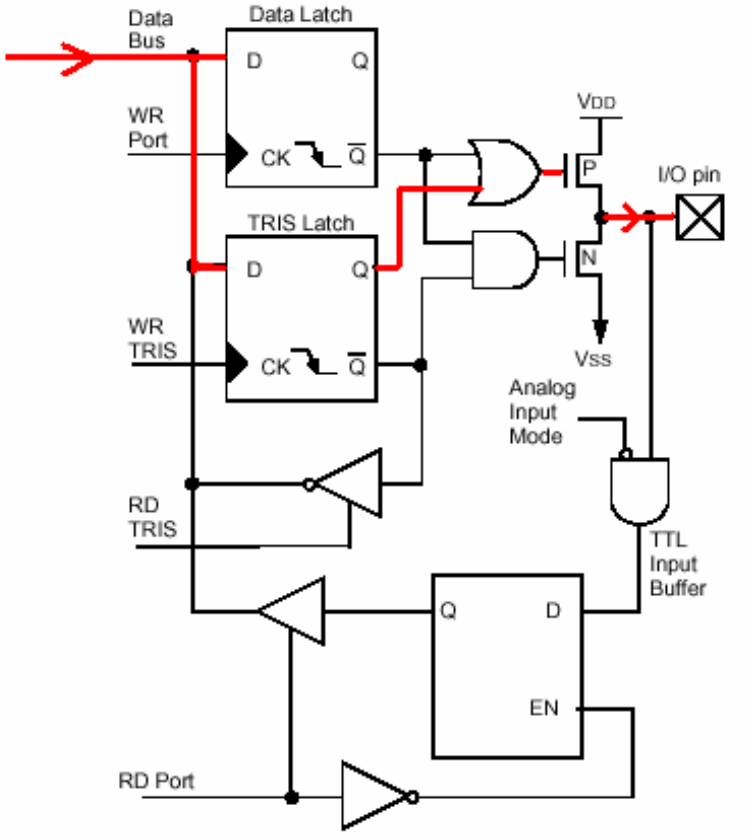
4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

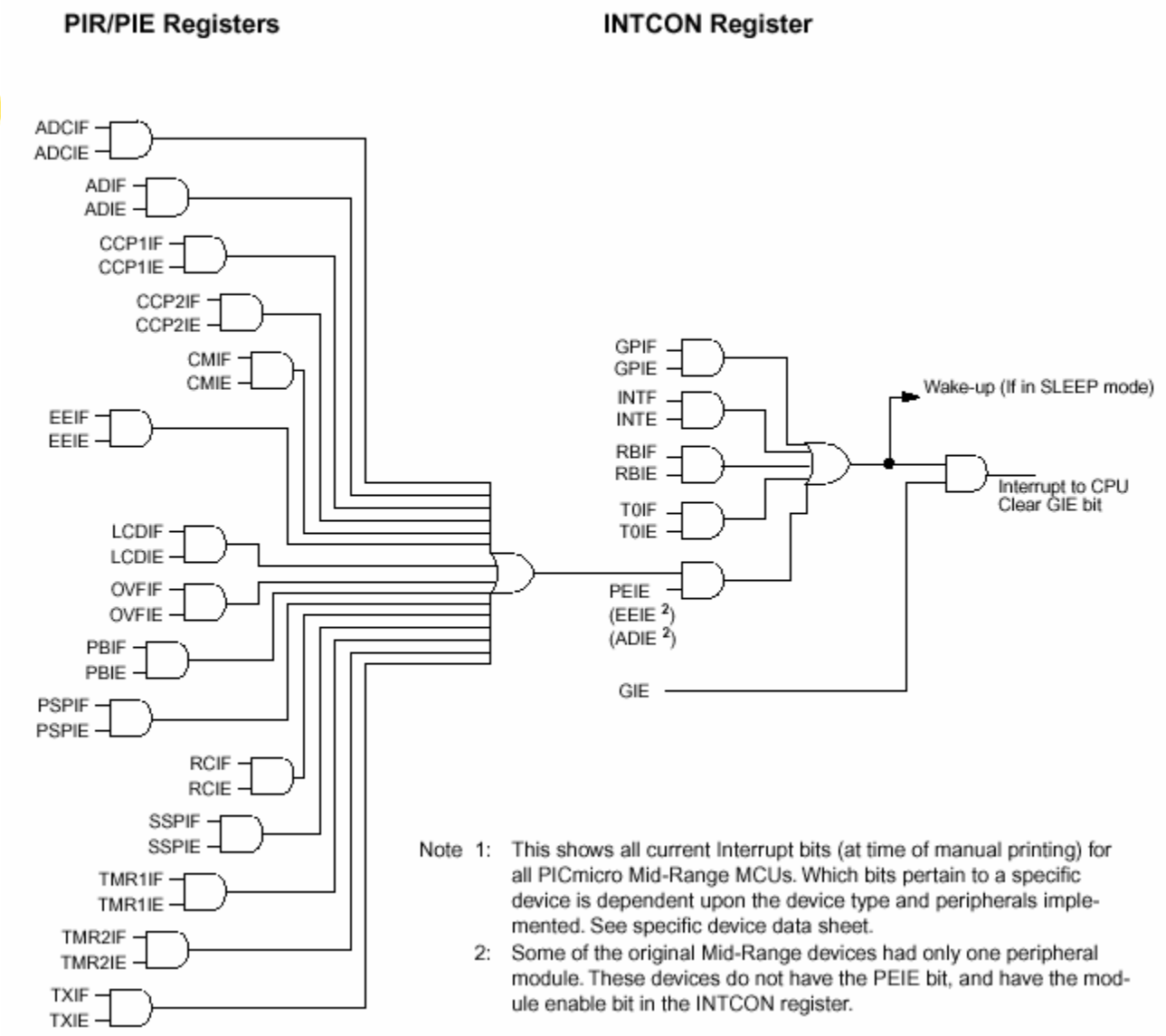
6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Input/Output Ports



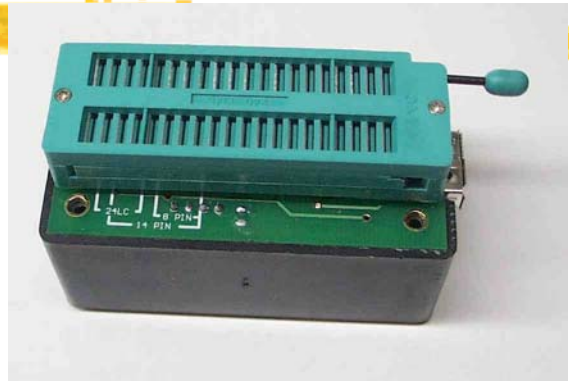


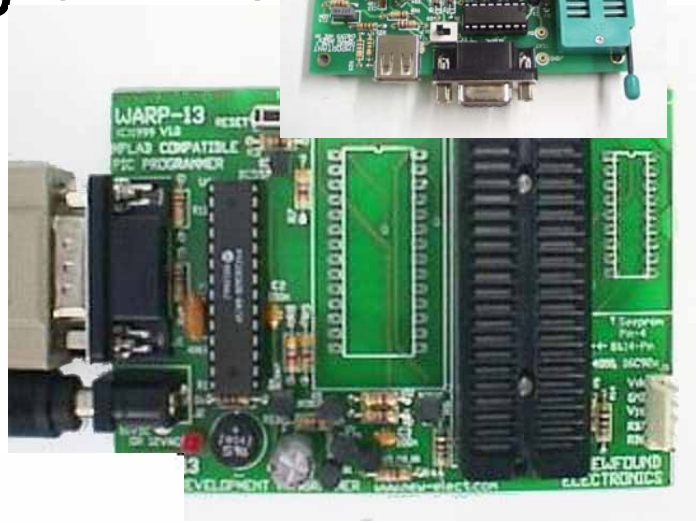
Interrupts

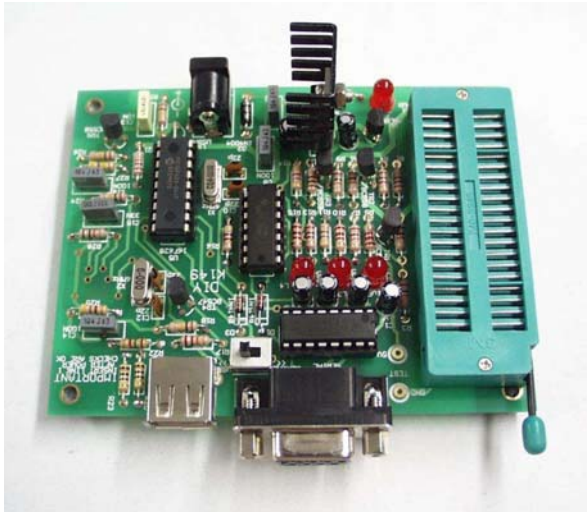

PIR/PIE Registers
INTCON Register
ADCIF
ADCIE
ADIF
ADIE
CCP1IF
CCP1IE
CCP2IF
CCP2IE
CMIF
CMIE
EEIF
EEIE
LCDIF
LCDIE
OVFIF
OVFIE
PBIF
PBIE
PSPIF
PSPIE
RCIF
RCIE
SSPIF
SSPIE
TMR1IF
TMR1IE
TMR2IF
TMR2IE
TXIF
TXIE
GPIF
GPIE
INTF
INTE
RBIF
RBIE
T0IF
T0IE
PEIE
(EEIE 2)
(ADIE 2)
GIE
Wake-up (If in SLEEP mode)
Interrupt to CPU
Clear GIE bit
Note 1: This shows all current Interrupt bits (at time of manual printing) for all PICmicro Mid-Range MCUs. Which bits pertain to a specific device is dependent upon the device type and peripherals implemented. See specific device data sheet.
2: Some of the original Mid-Range devices had only one peripheral module. These devices do not have the PEIE bit, and have the module enable bit in the INTCON register.

16F877 Hex Code Download

ogramme







Boot Loader Option

Bootstrap loader

Also known as **bootstrapping** or **boot loader**, a **bootstrap** loader is a program that resides in the computers EPROM, ROM, or other non-volatile memory that automatically executed by the processor when the computer is turned on. The bootstrap loader reads the hard disk drives boot sector to continue the process of loading the computers Operating System.

boot loader

A small program that loads the operating system into the computer's memory when the system is booted and also starts the operating system.

bootstrap

• **noun 1** a loop at the back of a boot, used to pull it on. **2** Computing the action of loading a program into a computer by means of a few initial instructions which enable the introduction of the rest of the program from an input device.

boot·strap

noun (*plural* boot·straps)

Definition:

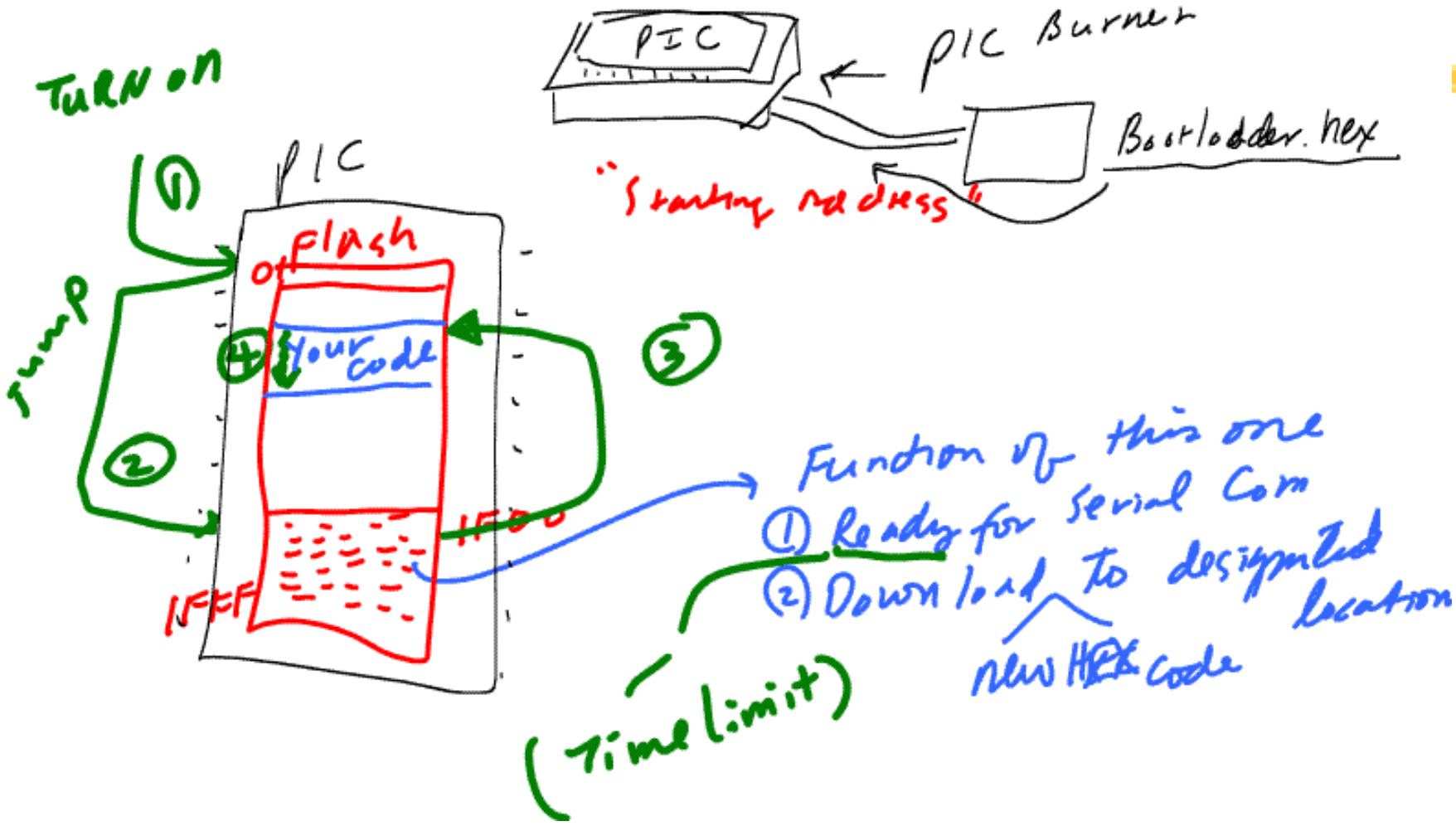
loop attached to boot: a leather or fabric loop on the back or side of a boot to help pull it on

adjective

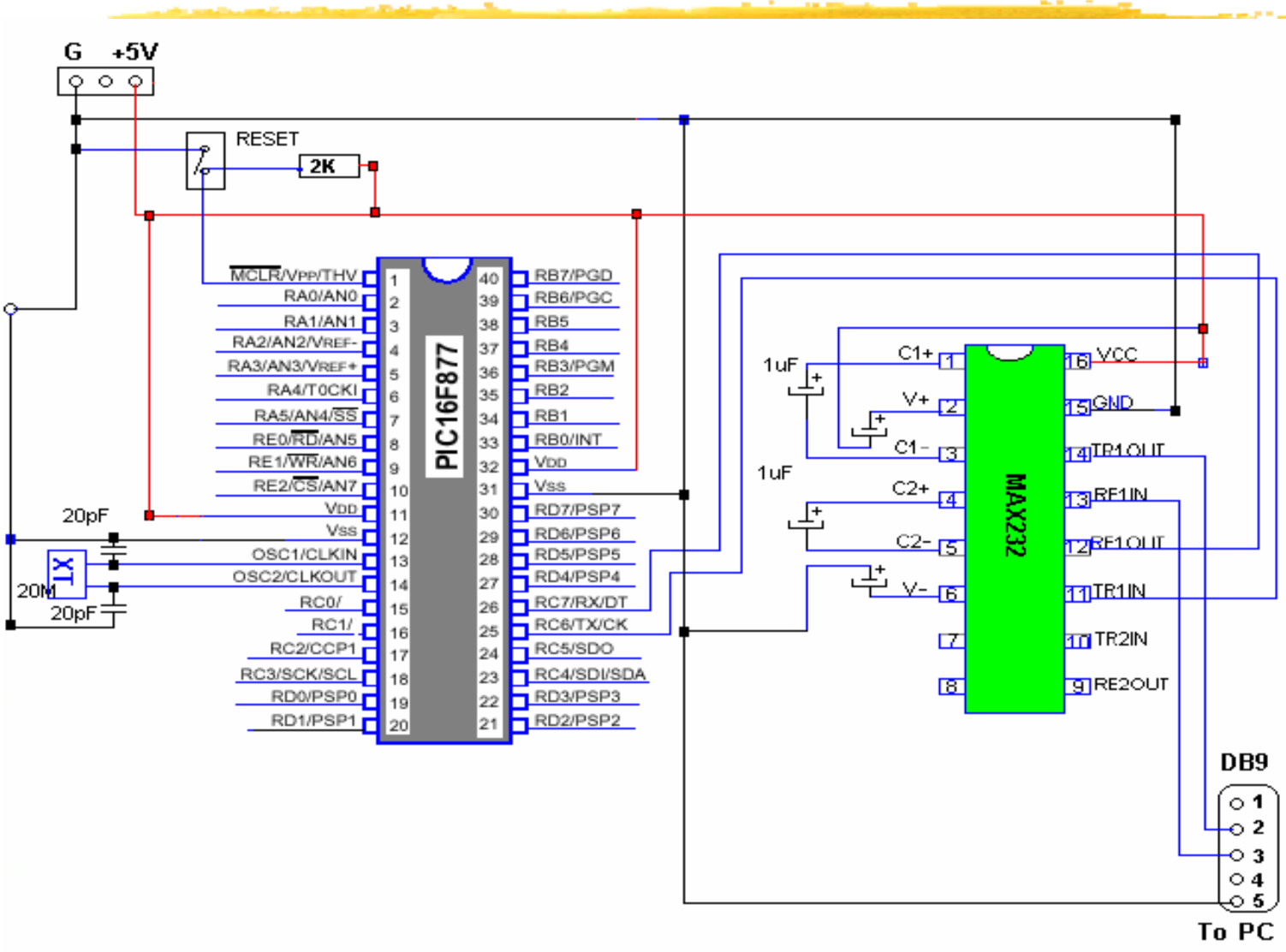
Definition:

self-reliant and self-sustaining: relying solely on somebody's own efforts and resources

PIC16F877 Bootloader



Minimum Hardware for Boot Loaded Platform

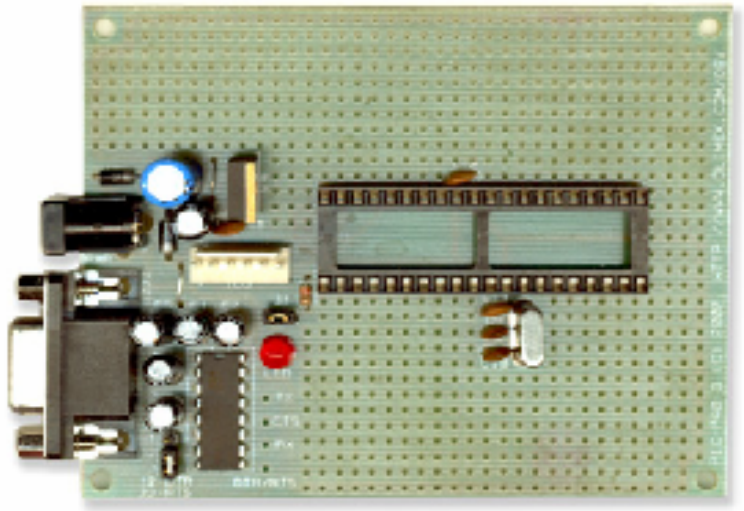


Commercially Available PIC16F877 Board

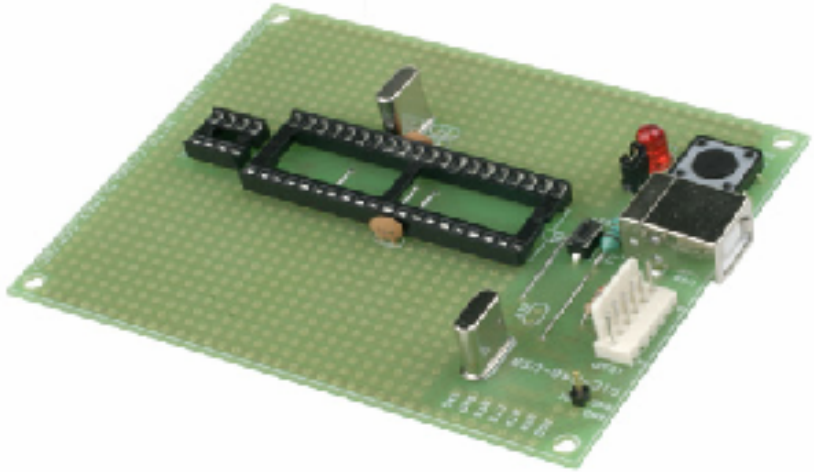
OLIMEX

www.olimex.com >>>Proto Board

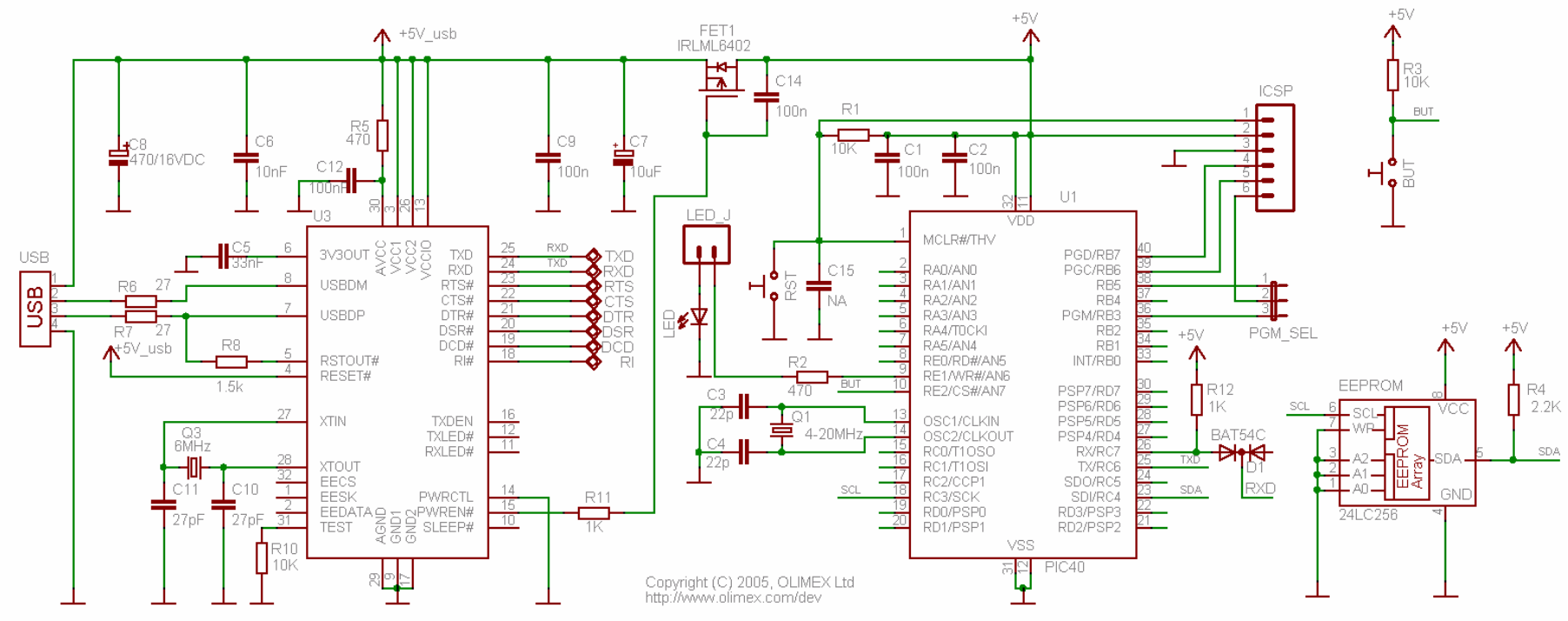
PIC-40B



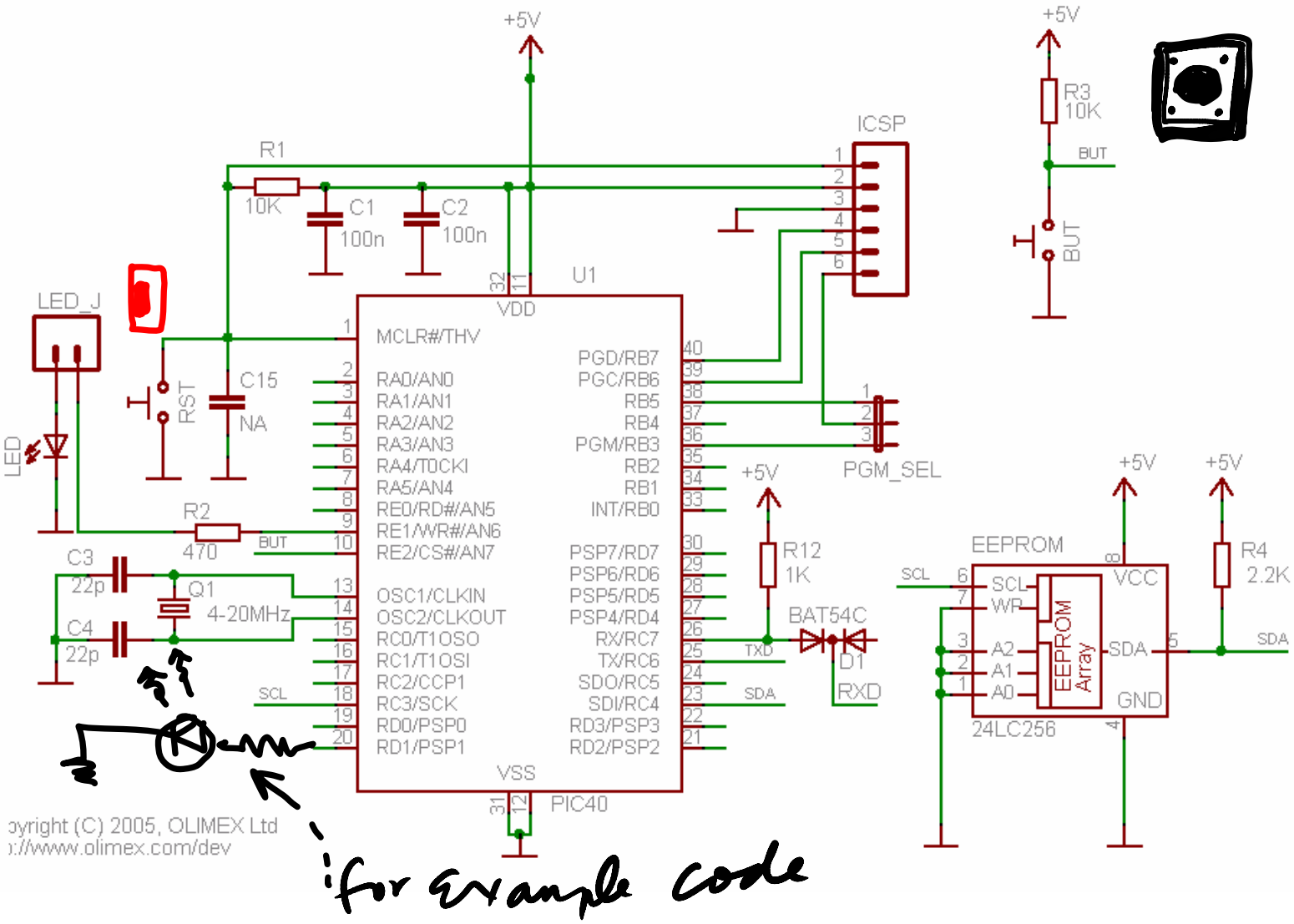
PIC-40B-USB



PIC-40B-USB Schematic



PIC-40B-USB Schematic (PIC area only)



Other PIC Board (Not fully evaluated yet)

- DLP-245PB-G-USB
- Not evaluated
- Problem in Bootloader downloading
- Problem in USB driver with Windows