CENG 336

Architecture

Areas of use

- You are used to chips like the Pentium and the Athlon, but in terms of installed machines these are a small portion of total computer use.
- Think how many computers you have at home
- Digital cameras, video cameras, TVs, mobile phones, calculators, micro-wave ovens etc all contain processors.

Microcontrollers

- In order for a microprocessor to be used, other components such as memory, or components for receiving and sending data must be added to it.
- •On the other hand, microcontroller is designed to be all of that in one. No other external components are needed for its application because all necessary peripherals are already built into it. Thus, we save the time and space needed to construct devices.
- •The PICmicrowas originally designed around 1980 by General Instrument as a small, fast, inexpensive embedded microcontroller with strong I/O capabilities.PIC stands for "**P**eripheral Interface **C**ontroller".
- •General Instrument recognized the potential for the PIC and eventually spun off Microchip, headquartered in Chandler, AZ to fabricate and market the PICmicro.

Numbers of machines

- You might have 1 or 2 Pentium class chips at home. (Each PC will also contain several embedded processors)
- You will have perhaps 10 to 20 other embeded computers in other devices.
- If you think of childrens toys, the numbers grow even higher.

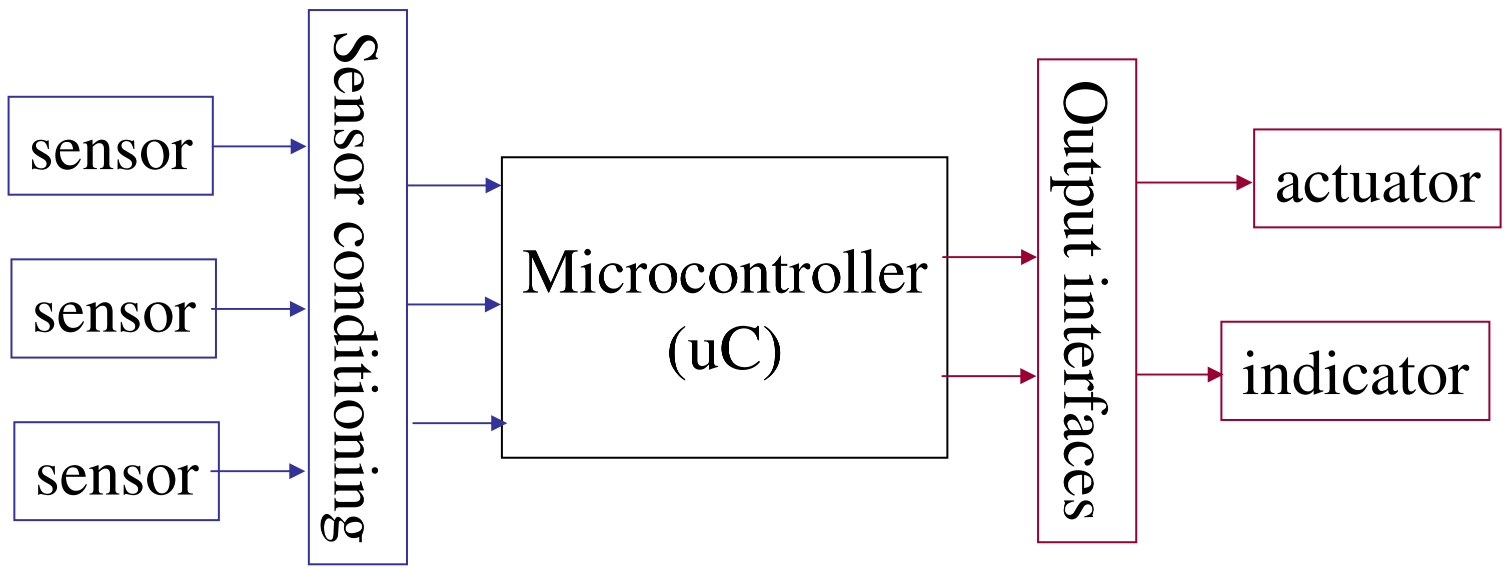
Characteristics

- Embedded computers have to be very low cost, simple and reliable.
- They can not use any moving parts (disk drives) because:
  1. These are power hungry
  2. They are bulky
  3. They are expensive

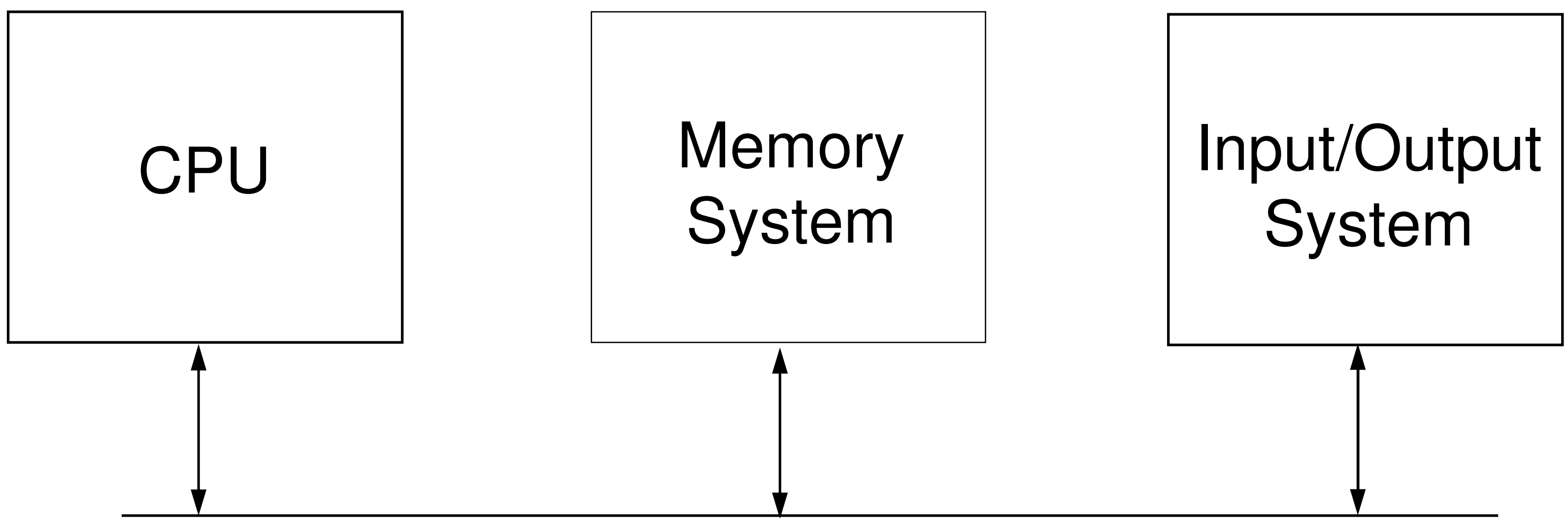
Features

- Program in Read Only Memory – ROM
- Limited RAM storage – variables only not code
- Built in I/O devices
- Use very little power

Embedded System General Block Diagram

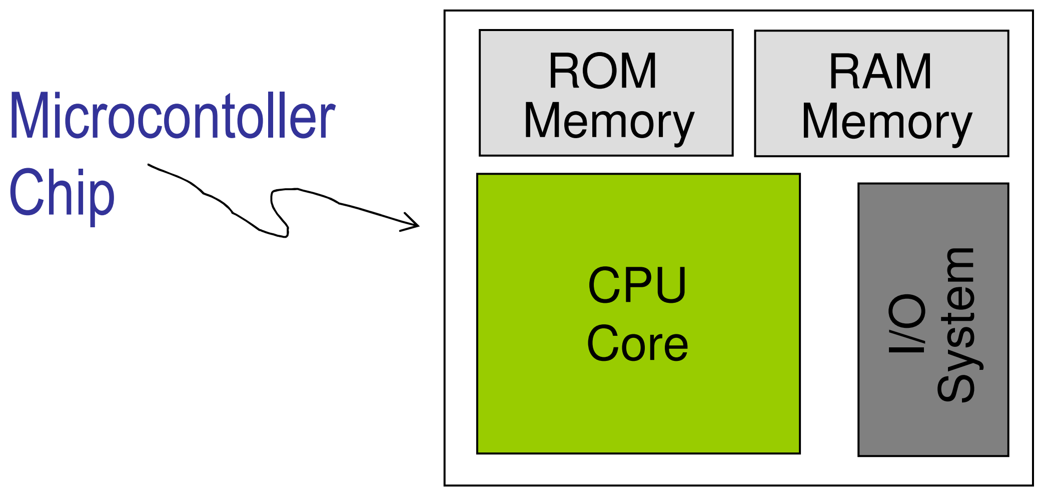


Components of a Computer System



Microcontrollers

- Microcontrollers integrate all the components of a computer system onto a single chip
- All components are optimized to perform the functions necessary to control a larger system
- Size, capability, cost, and power consumption are more important considerations
- 8 bit microcontrollers have the majority of the market right now, but 16 and 32 bit microcontrollers are available and gaining market share



Common Microcontrollers

- Atmel
- ARM
- Intel
    - 8-bit
        - 8XC42
        - MCS48
        - **MCS51** ←
        - 8xC251
    - 16-bit
        - MCS96
        - MXS296
- National Semiconductor
    - COP8
- Microchip
    - 12-bit instruction PIC
    - 14-bit instruction PIC
        - PIC16F84
    - 16-bit instruction PIC
- NEC
- Motorola
    - 8-bit
        - 68HC05
        - 68HC08
        - 68HC11
    - 16-bit
        - 68HC12
        - 68HC16
    - 32-bit
        - 683xx
- Texas Instruments
    - TMS370
    - MSP430
- Zilog
    - Z8
    - Z86E02

Families

- Zilog Z8 series
- Intel 8051 series
- Arm 32 bit microcontrollers
- MicroChip – PIC microcontrollers

Zilog

- This company did more than any other to spread use of microprocessors in the 1970s and early 80s with their Z80 micro.
- This was the mainstay of early personal computers.

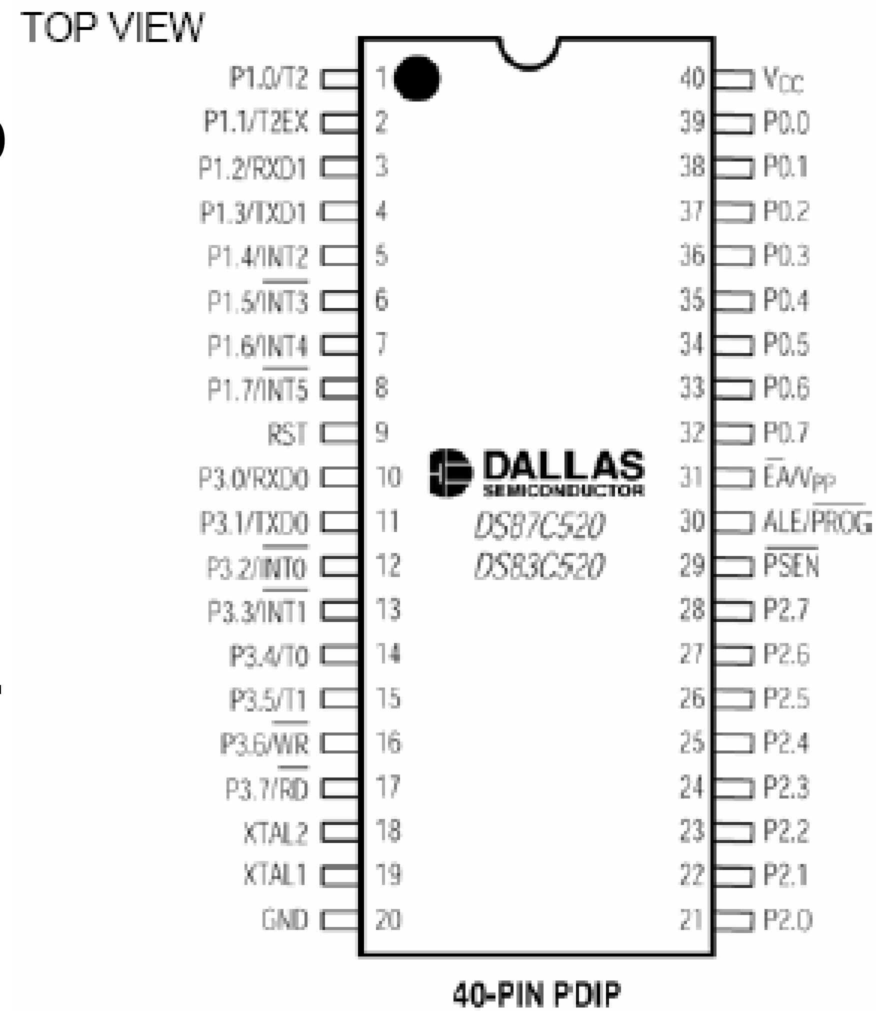
Current Z80e has

- eZ80® core operating at speeds up to 50MHz-Achieves high performance
- On-chip 10/100BaseT Ethernet MAC (eZ80F91 only) -Enables low-cost network applications
- Up to 256KB on-chip Flash memory
- Up to 16KB on-chip SRAM

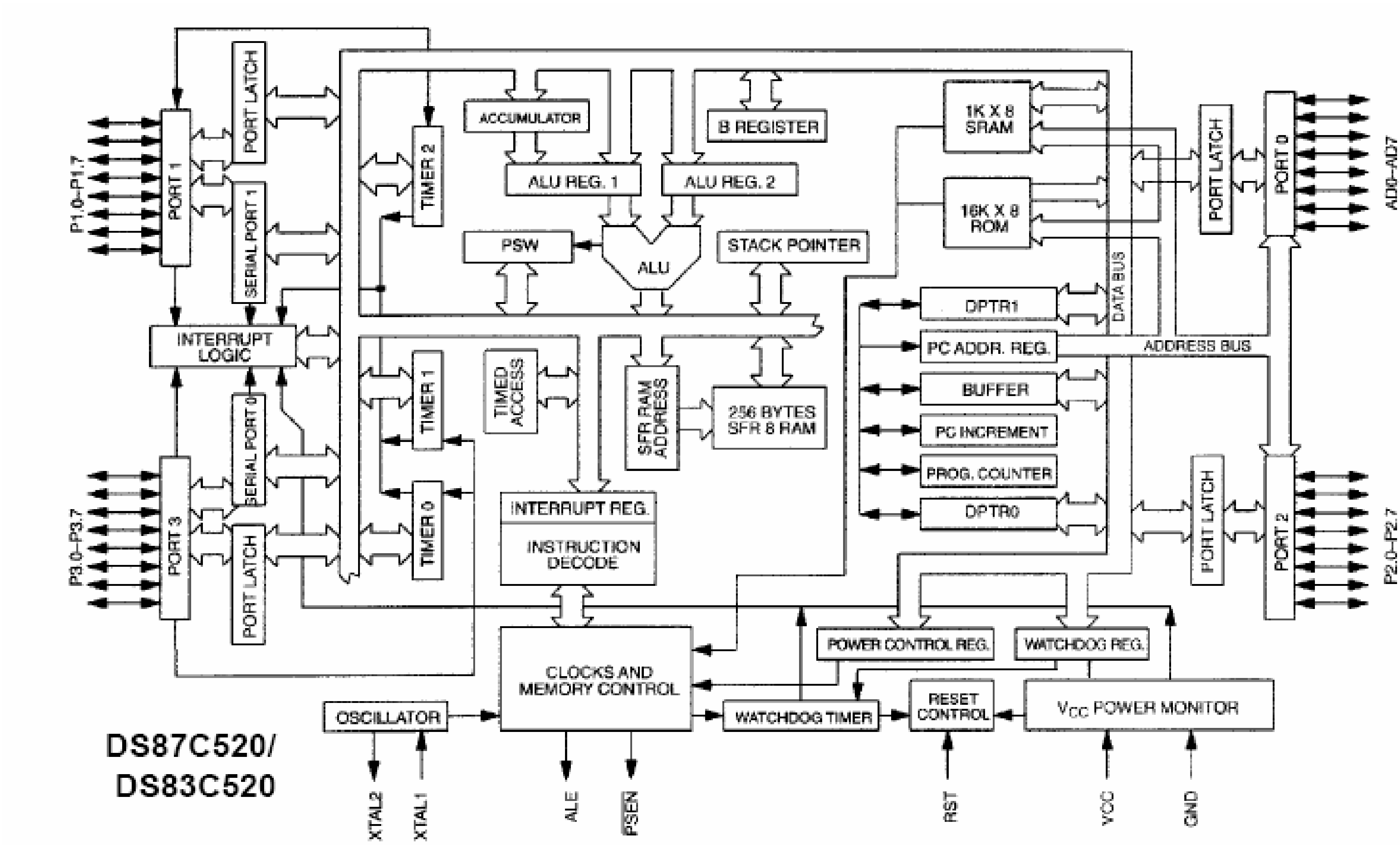
This is a relatively high end micro controller

Intel 8051

- This was one of the first microcontrollers to integrate ROM, RAM and I/O on one chip.
- Made by many other companies still, Amtel, Texas instruments etc.



8051Internal design



ARM series

- One of the first RISC processors
- 32 Bit architecture
- Low transitor count compared to conventional processors
- Very low power consumption
- Used in digital Phones and PDAs.
- At the top end of the micro-controller performance spectrum.

PIC

- Range of low end 8 bit microcontrollers.
- smallest have only 8 pins, largest 40 pins.
- Typical chip is an 18 pin one.
- Very cheap, you can pick them up at less than £1 each.
- Targeted at consumer products, burglar alarms etc.

Advantages of PIC

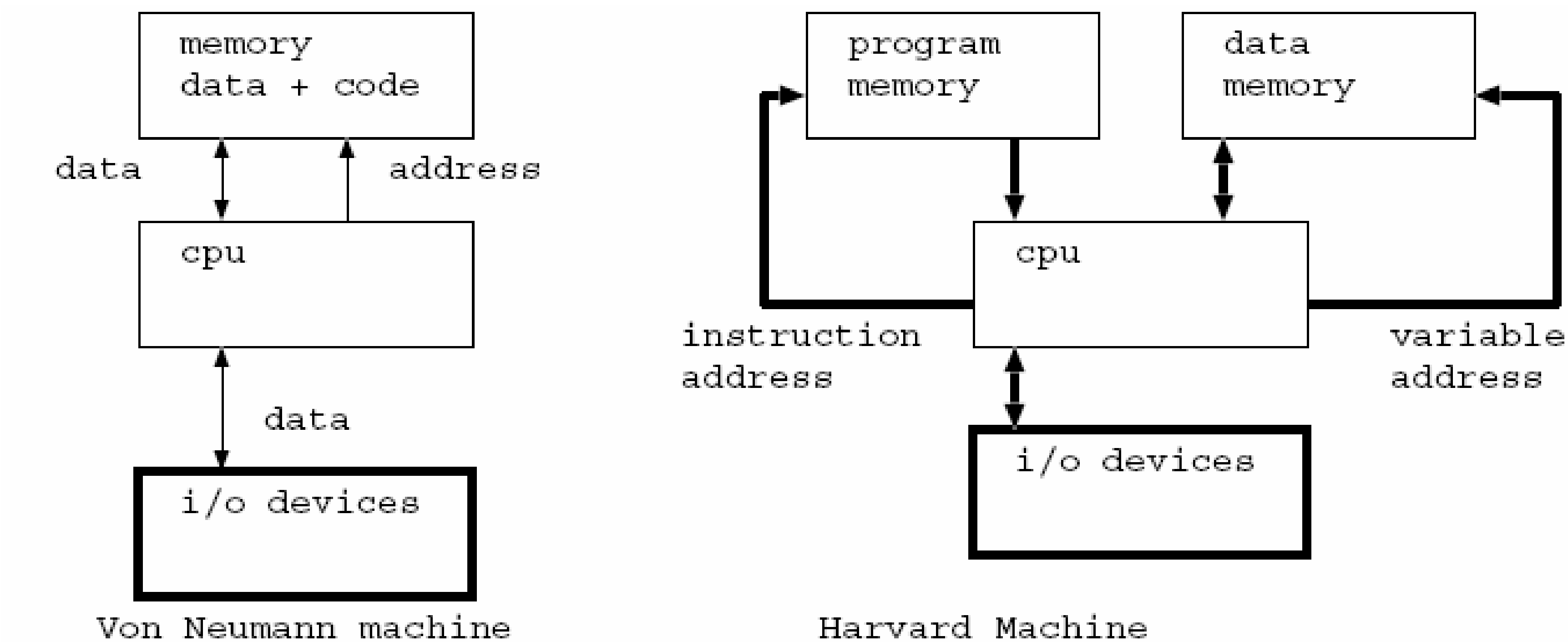
- It is a RISC (Reduced Instruction Set Computer) design
- Only thirty seven instructions to remember
- Its code is extremely efficient, allowing the PIC to run with typically less program memory than its larger competitors.
- It is low cost, high clock speed

Harvard architecture

- Like many micros the PIC is a Harvard not a von-Neumann machine
- This is simpler and faster

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

comparison

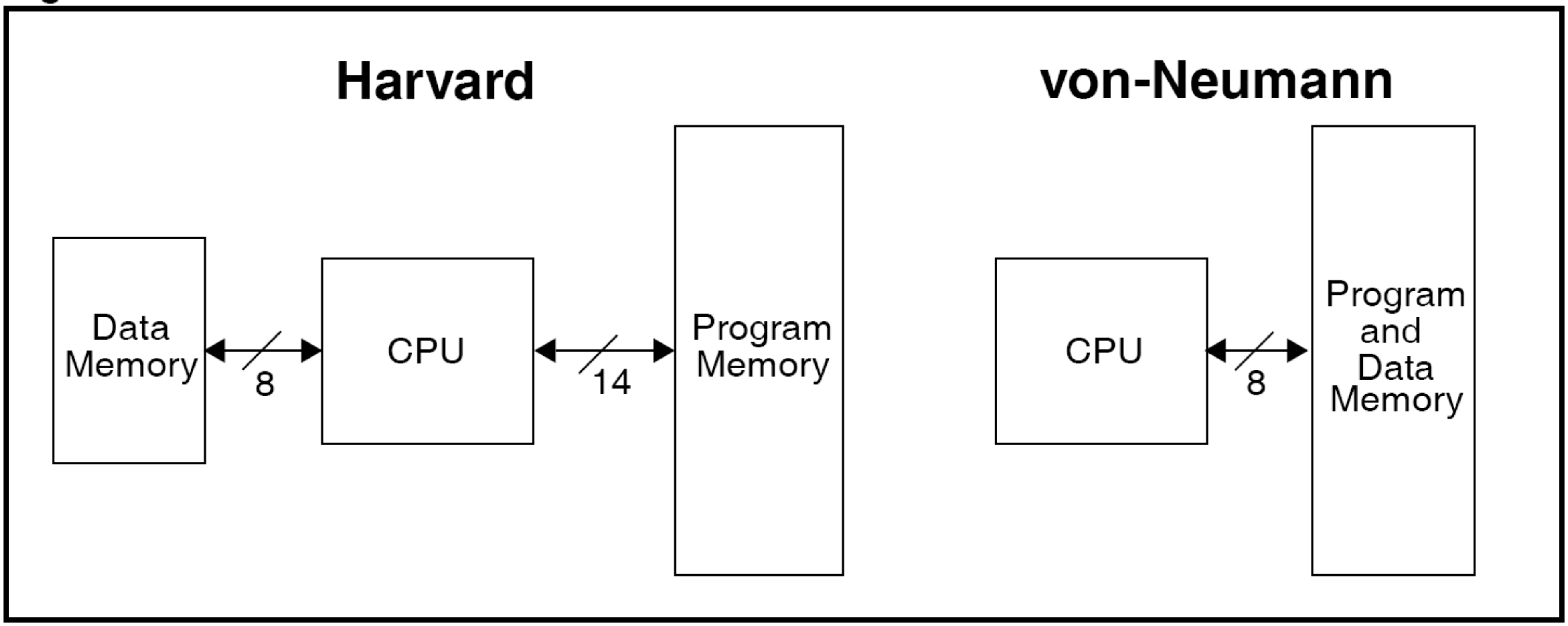

memory
data + code
data
address
cpu
data
i/o devices
Von Neumann machine
program
memory
data
memory
cpu
instruction
address
variable
address
i/o devices
Harvard Machine

- Harvard architecture is a newer concept than von-Neumann's. It rose out of the need to speed up the work of a microcontroller.
- In Harvard architecture, Data Access and Address Access are seperate. Thus a greater flow of data is possible through the central processing unit.
- PIC16F877 uses 14 bits for instructions which allows for all instructions to be one word instructions.

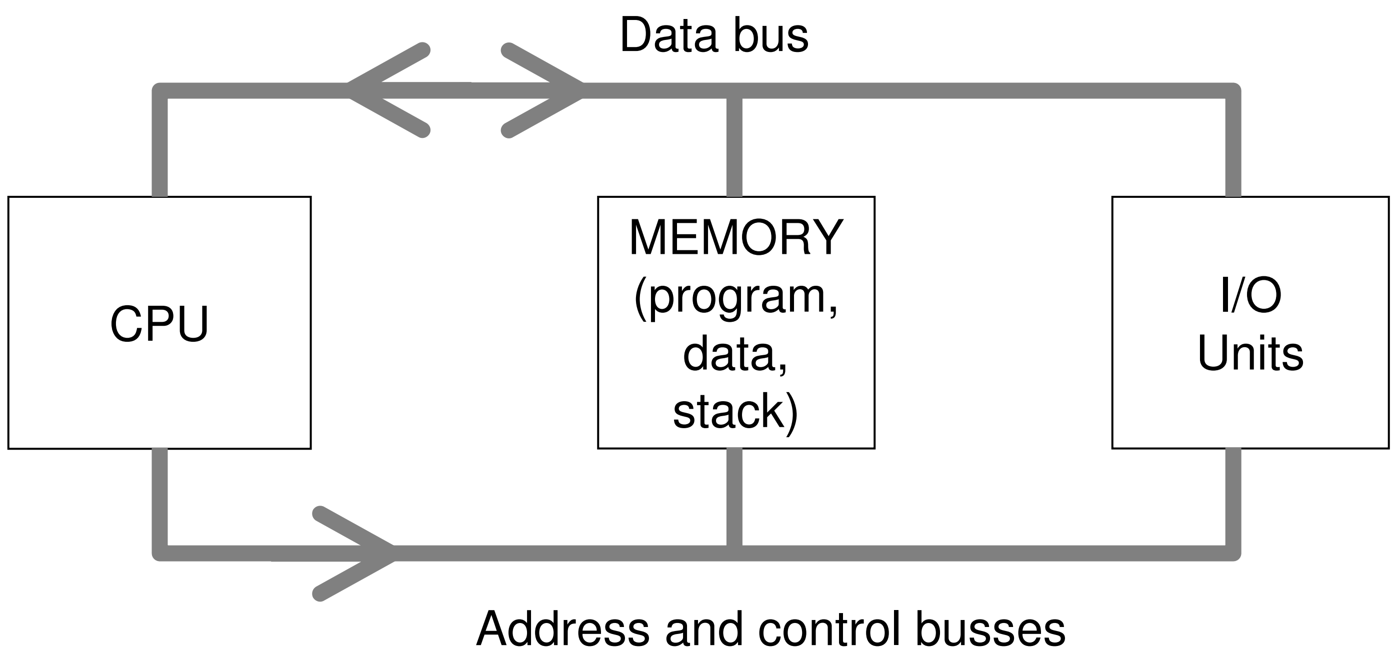
Advantages of Harvard model

- An add operation of the form a:=b+c must fetch 2 operands from memory and write 1 operand to memory. In addition it is likely to have to fetch 3 instructions from memory.
- With a single memory this will take 6 cycles. With 2 memories, we can fetch the instructions in parallel with the data and do it in 3 cycles.
- We have different word lengths for instructions and data – 8 bit data and perhaps 12 bit instructions.

Figure 4-1: Harvard vs. von Neumann Block Architectures



Von Neuman Architecture



Von Neumann 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

RISC Architecture

- Complex/Reduced Instruction Set Computers
- A minimal set of instructions, combined, can do every operation
- Usually execute in a single cycle
- CPU is smaller
- Other hardware can be added to the space: (overlapping register windows)

PIC Architecture: Background

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 (Tcyc) to execute
- Example:

MC68HC05: **LDAA 0x55**

1000	**1100**
01010101	

2 bytes, 2 cycles

PIC Architecture: Background

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: **MOVLW 0x55**

1100XX	01010101

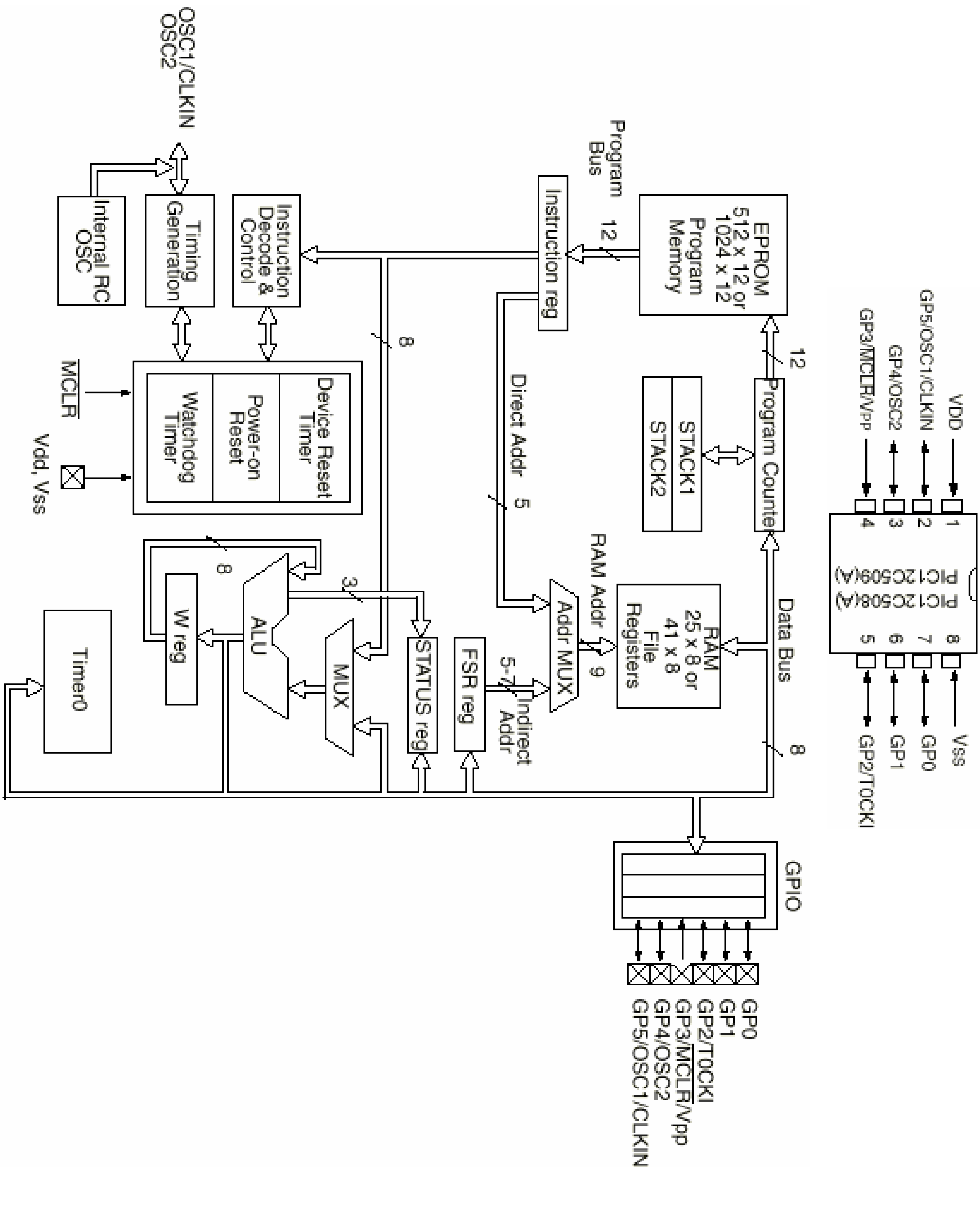
1 word, 1 cycle

PIC Architecture: Convergence

Many Microcontrollers and DSP chips are “converging”

- Heading towards some mean between RISC and CISC
- Large CPUs (DSPs) are adding microcontroller like options
  (the 32bit, 100MHz StrongARM draws only 70mA)
- Small microcontrollers are getting more powerful, now able to do some DSP
- General trend: Smaller packages, less power consumption, faster
- Future possibility: “Sea of gates” reconfigurable processor

Example PIC: 12C508 Block Diagram



The PIC Family: Cores

PICs come with 1 of 4 CPU ‘cores’:

- ⌫12bit cores with 33 instructions: 12C50x, 16C5x
- ⌫14bit cores with 35 instructions: 12C67x,16Cxxx
- ⌫16bit cores with 58 instructions: 17C4x,17C7xx
- ⌫‘Enhanced’ 16bit cores with 77 instructions: 18Cxxx

The PIC Family: Packages

PICs come in a huge variety of packages:

8 pin DIPs, SOICs: 12C50x (12bit) and 12C67x (14bit)
18pin DIPs, SOICs: 16C5X (12bit), 16Cxxx (14bit)
28pin DIPs, SOICs: 16C5X (12bit), 16Cxxx (14bit)
40pin DIPs, SOICs: 16Cxxx (14bit), 17C4x (16bit)
44 - 68pin PLCCs*: 16Cxxx (14bit), 17C4x / 17Cxxx (16bit)

* also TQFPs, etc.

The PIC Family: Speed

PICs require a clock to work.

- ⌫ Can use crystals, clock oscillators, or even an RC circuit.
- ⌫ Some PICs have a built in 4MHz RC clock
  - – Not very accurate, but requires no external components!
- ⌫ Instruction speed = 1/4 clock speed ($T_{cyc} = 4 * T_{clk}$)
- ⌫ All PICs can be run from DC to their maximum spec'd speed:

12C50x	4MHz
12C67x	10MHz
16Cxxx	20MHz
17C4x / 17C7xxx	33MHz
18Cxxx	40MHz

The PIC Family: Program Memory

PIC program space is different for each chip.

Some examples are:

12C508	512 12bit instructions
16C71C	1024 (1k) 14bit instructions
16F877	8192 (8k) 14bit instructions
17C766	16384 (16k) 16bit instructions

The PIC Family: Program Memory

PICs have two different types of program storage:

1. EPROM (Erasable Programmable Read Only Memory)

- Needs high voltage from a programmer to program (~13V)
- Needs windowed chips and UV light to erase
- Note: One Time Programmable (OTP) chips are EPROM chips, but with no window!
- PIC Examples: Any ‘C’ part: 12C50x, 17C7xx, etc.

The PIC Family: Program Memory

PICs have two different types of program storage:

2. FLASH

- Re-writable (even by chip itself)
- Much faster to develop on!
- Finite number of writes (~100k Writes)
- PIC Examples: Any ‘F’ part: 16F84, 16F87x, 18Fxxx (future)

The PIC Family: Data Memory

PICs use general purpose “file registers” for RAM (each register is <u>8bits</u> for all PICs)

Some examples are:

12C508	25 Bytes RAM
16C71C	36 Bytes RAM
16F877	368 Bytes (plus 256 Bytes of nonvolatile EEPROM)
17C766	902 Bytes RAM

- Don’t forget, programs are stored in program space (not in data space), so low RAM values are OK.

The PIC Family: Control Registers

PICs use a series of “special function registers” for controlling peripherals and PIC behaviors.

Some examples are:

STATUS	Bank select bits, ALU bits (zero, borrow, carry)
INTCON	Interrupt control: interrupt enables, flags, etc.
TRIS	Tristate control for digital I/O: which pins are ‘floating’
TXREG	UART transmit register: the next byte to transmit

The PIC Family: Peripherals

Different PICs have different on-board peripherals

Some common peripherals are:

- Tri-state ("floatable") digital I/O pins
- Analog to Digital Converters (ADC) (8, 10 and 12bit, 50ksps)
- Serial communications: UART (RS-232C), SPI, I^2C, CAN
- Pulse Width Modulation (PWM) (10bit)
- Timers and counters (8 and 16bit)
- Watchdog timers, Brown out detect, LCD drivers

PIC Peripherals: Ports (Digital I/O)

- All PICs have digital I/O pins, called ‘Ports’
  - the 8pin 12C508 has 1 Port with 4 digital I/O pins
  - the 68pin 17C766 has 9 Ports with 66 digital I/O pins
- Ports have 2 control registers
  - TRISx sets whether each pin is an input or output
  - PORTx sets their output bit levels
- Most pins have 25mA source/sink (directly drives LEDs)
- WARNING: Other peripherals SHARE pins!

PIC Peripherals: ADCs

- Only available in 14bit and 16bit cores
- Fs (sample rate) < 54KHz
- Most 8bits, newer PICs have 10 or 12bits
- All are +/- 1LSB and are monotonic
- Theoretically higher accuracy when PIC is in sleep mode (less digital noise)
- Can generate an interrupt on ADC conversion done
- Multiplexed 3 (12C671) - 12 (17C7xxx) channel input
  - Must wait Tacq to charge up sampling capacitor (see datasheets)

PIC Peripherals: USART: UART

- Serial Communications Peripheral: Universal Synchronous/Asynchronous Receiver/Transmitter
- Only available in 14bit and 16bit cores
- Interrupt on TX buffer empty and RX buffer full

- Asynchronous communication: UART (RS-232C serial)
  - Can do 300bps - 115kbps
  - 8 or 9 bits, parity, start and stop bits, etc.
  - Outputs 5V so you need a RS232 level converter (e.g., MAX232)

PIC Peripherals: USART: USRT

- Synchronous communication: i.e., with clock signal

- SPI = Serial Peripheral Interface
  - 3 wire: Data in, Data out, Clock
  - Master/Slave (can have multiple masters)
  - Very high speed (1.6Mbps)
  - Full speed simultaneous send and receive (Full duplex)

- I^2C = Inter IC
  - 2 wire: Data and Clock
  - Master/Slave (Single master only; multiple masters clumsy)
  - Lots of cheap I^2C chips available; typically < 100kbps
    (For example, 8pin EEPROM chips, ADC, DACs, etc.)

PIC Peripherals: Timers

- Available in all PICs.
- 14+bit cores may generate interrupts on timer overflow.
- Some 8bits, some 16bits, some have prescalers
- Can use external pins as clock in/clock out
  (ie, for counting events or using a different F_{OSC})
- Warning: some peripherals share Timer resources

PIC Peripherals: CCP Modules

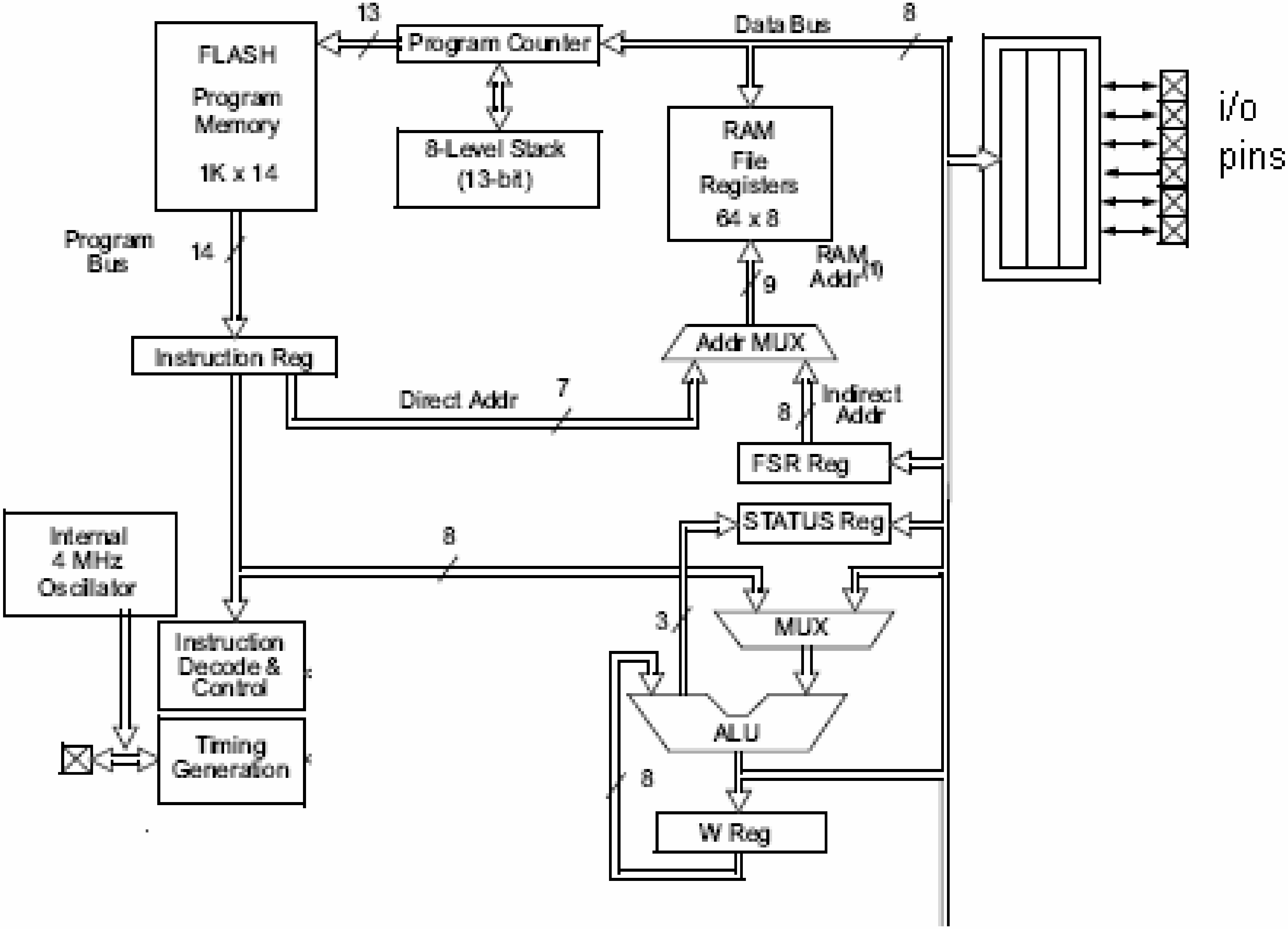
- Capture/Compare/PWM (CCP)
- 10bit PWM width within 8bit PWM period (frequency)
  - Enhanced 16bit cores have better bit widths
- Frequency/Duty cycle resolution tradeoff
  - 19.5KHz has 10bit resolution
  - 40KHz has 8bit resolution
  - 1MHz has 1bit resolution (makes a 1MHz clock!)
- Can use PWM to do DAC - See AN655
- Capture counts external pin changes
- Compare will interrupt on when the timer equals the value in a compare register

PIC Peripherals: Misc.

- Sleep Mode: PIC shuts down until external interrupt (or internal timer) wakes it up.
- Interrupt on pin change: Generate an interrupt when a digital input pin changes state (for example, interrupt on keypress).
- Watchdog timer: Resets chip if not cleared before overflow
- Brown out detect: Resets chip at a known voltage level
- LCD drivers: Drives simple LCD displays
- Future: CAN bus, 12bit ADC, better analog functions

- VIRTUAL PERIPHERALS:
  – Peripherals programmed in software. UARTS, timers, and more can be done in software (but it takes most of the resources of the machine)

- For example: PIC16F877 is a RISC microcontroller,
  - it has a reduced set of instructions,
  - more precisely 35 instructions
  - (ex.Intel's and Motorola's microcontrollers have over hundred instructions).
  - All of these instructions are executed in one cycle except for jump and branch instructions.



Basic PIC architecture

Characteristics

- RISC CPUs
  - 8-bit
  - 16-bit
- Number of I/O pins: 4-70
- Memory types and sizes:
  - Flash; OTP; ROM
  - 0.5k – 256k

Speeds

- All PICs require oscillators to execute instructions:
  - Internal* (low speeds, up to 8 MHz)
  - External (high speeds, up to 40 MHz)
- Instructions are executed at least at ¼ oscillator speed (4 clocks/instruction)

(*Note: not all PICs have internal oscillators)

A/D converters and C/C modules

- All PICs have between 0 and 16 A/D converters with 8/10-bit resolution
- 8-16 bit Timers/Counters
- Comparator Modules (0-2)

Example: PIC16F877A

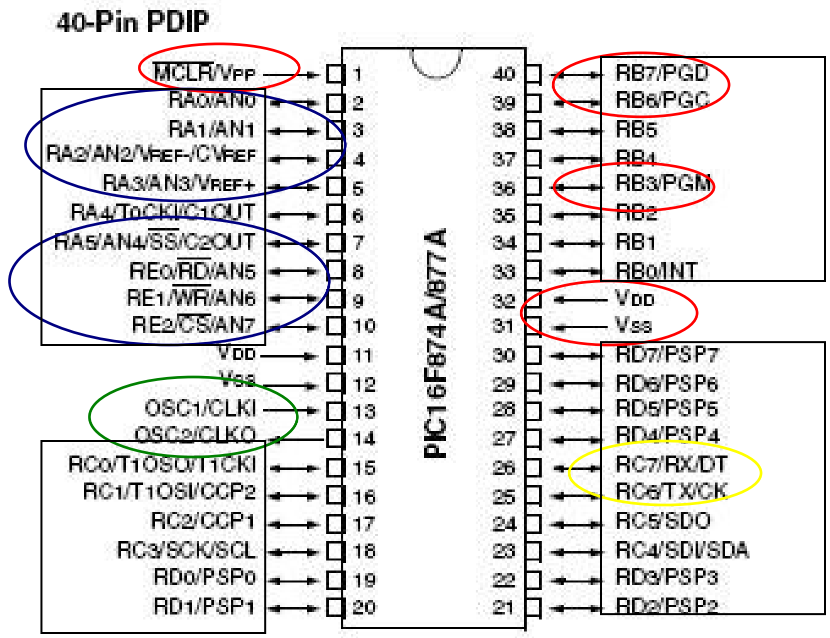
5/6 Programming pins

8 A/D channels

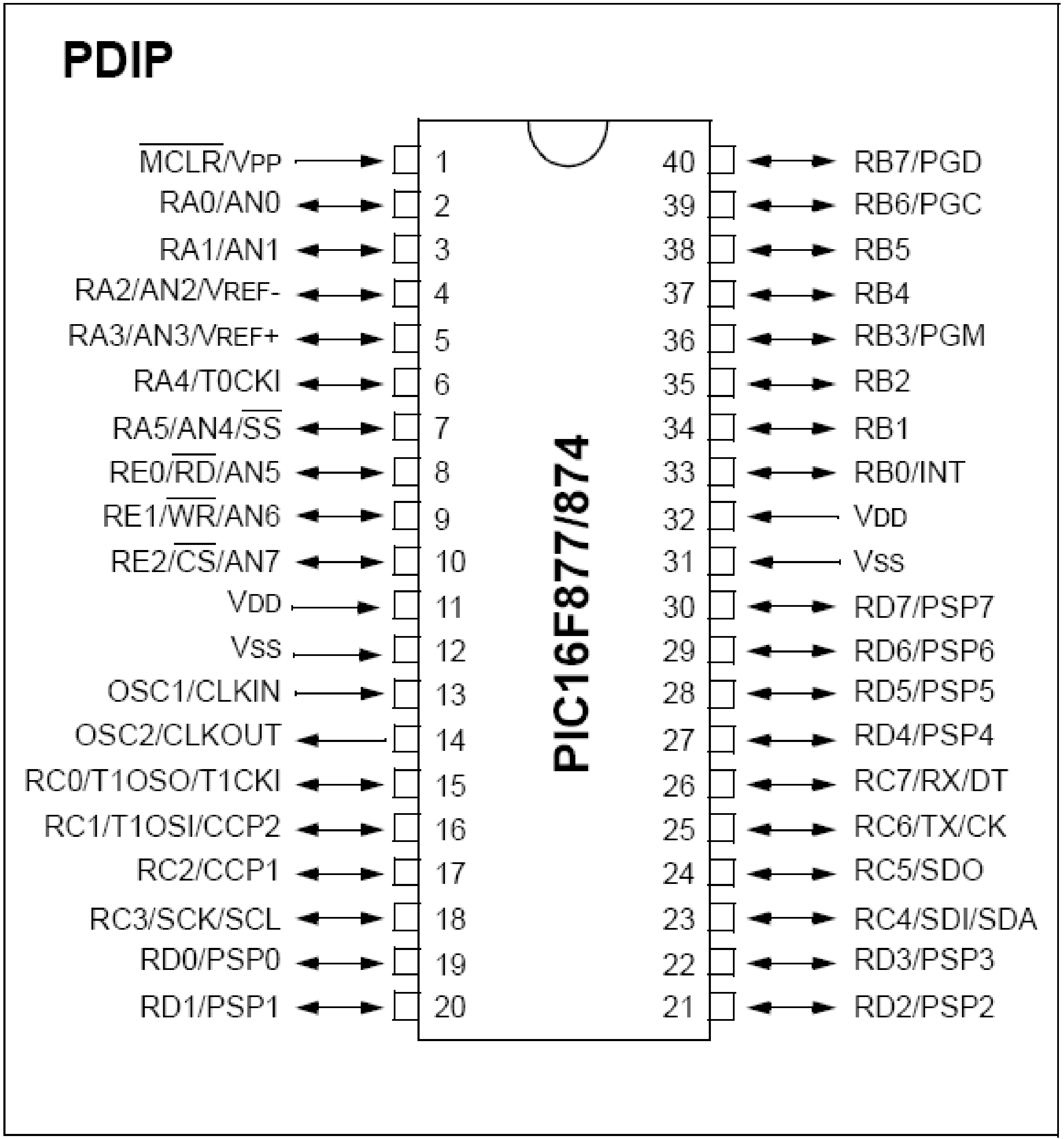
2 Oscillator Inputs

2 RS-232 inputs

33 I/O ports

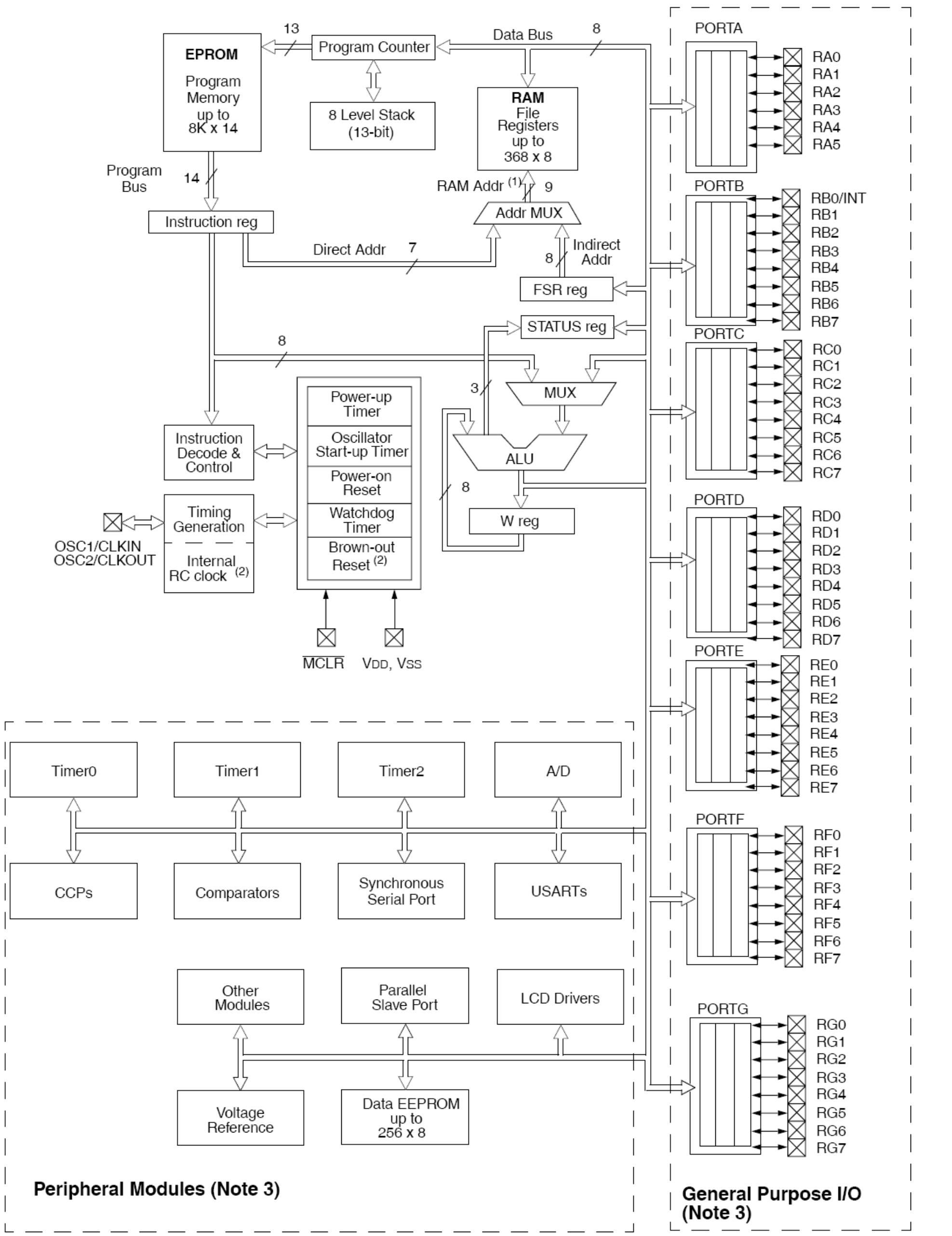


Pin Diagram



PDIP
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
VDD
VSS
OSC1/CLKIN
OSC2/CLKOUT
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RD0/PSP0
RD1/PSP1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
PIC16F877/874
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
RB7/PGD
RB6/PGC
RB5
RB4
RB3/PGM
RB2
RB1
RB0/INT
VDD
VSS
RD7/PSP7
RD6/PSP6
RD5/PSP5
RD4/PSP4
RC7/RX/DT
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
RD3/PSP3
RD2/PSP2

Figure 4-2: General Mid-range PICmicro Block Diagram

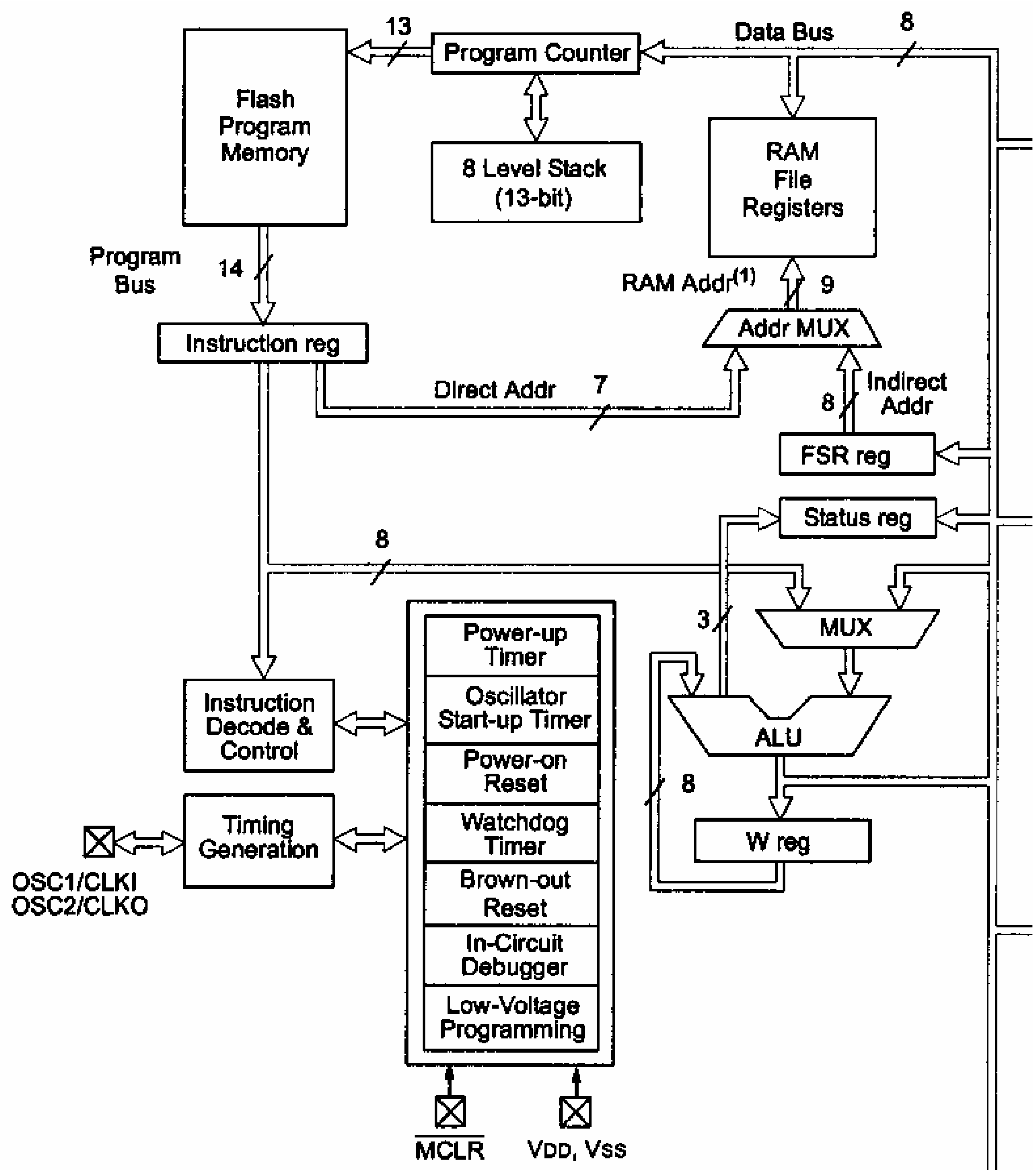


Note 1: The high order bits of the Direct Address for the RAM are from the STATUS register.

2: Not all devices have this feature, please refer to device data sheet.

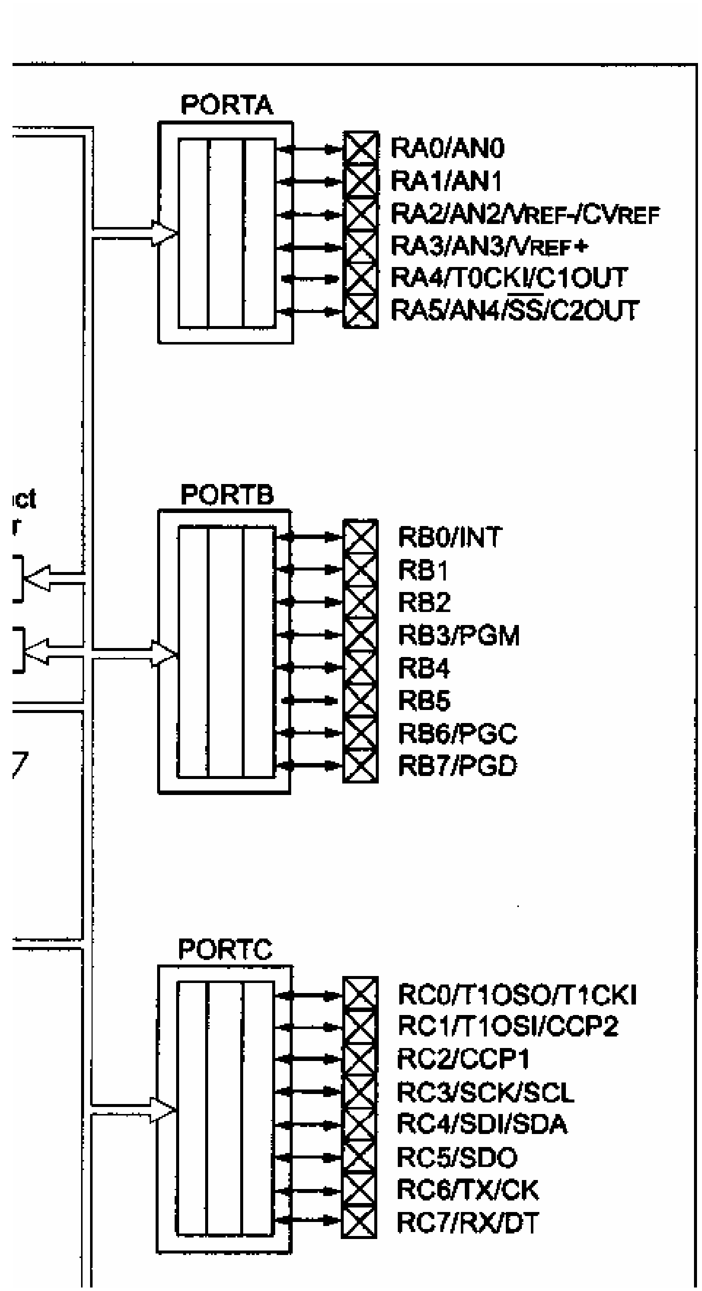
3: Many of the general purpose I/O pins are multiplexed with one or more peripheral module functions. The multiplexing combinations are device dependent.

Core PIC



13
Program Counter
Data Bus
8
Flash
Program
Memory
8 Level Stack
(13-bit)
RAM
File
Registers
Program
Bus
14
RAM Addr(1)
9
Instruction reg
Addr MUX
Direct Addr
7
Indirect
Addr
8
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
Timing
Generation
ALU
8
W reg
OSC1/CLKI
OSC2/CLKO
MCLR
VDD, VSS

PIC Ports


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

Included I/O Units

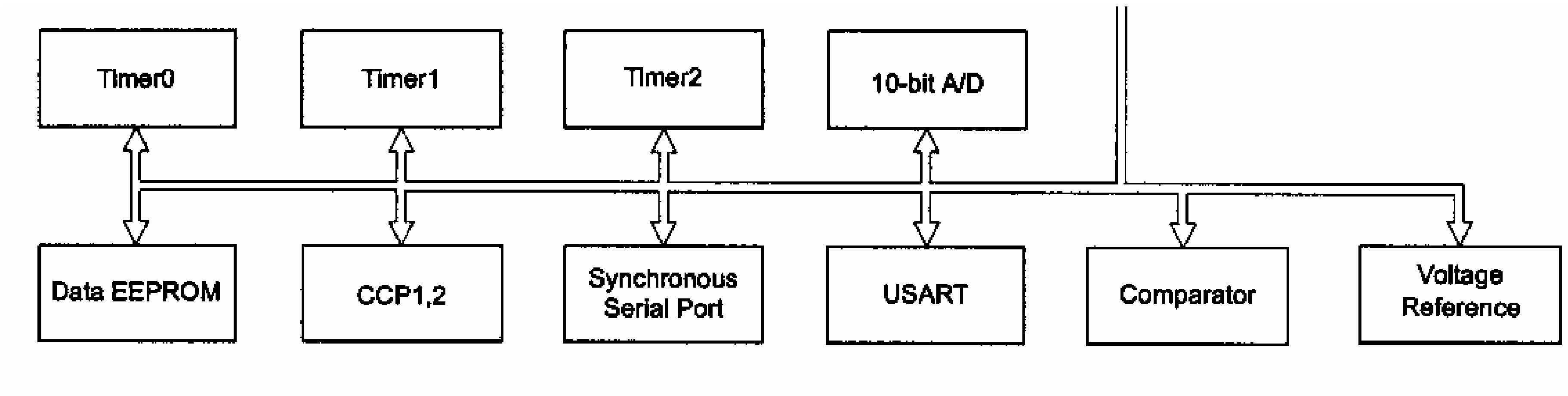


FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK

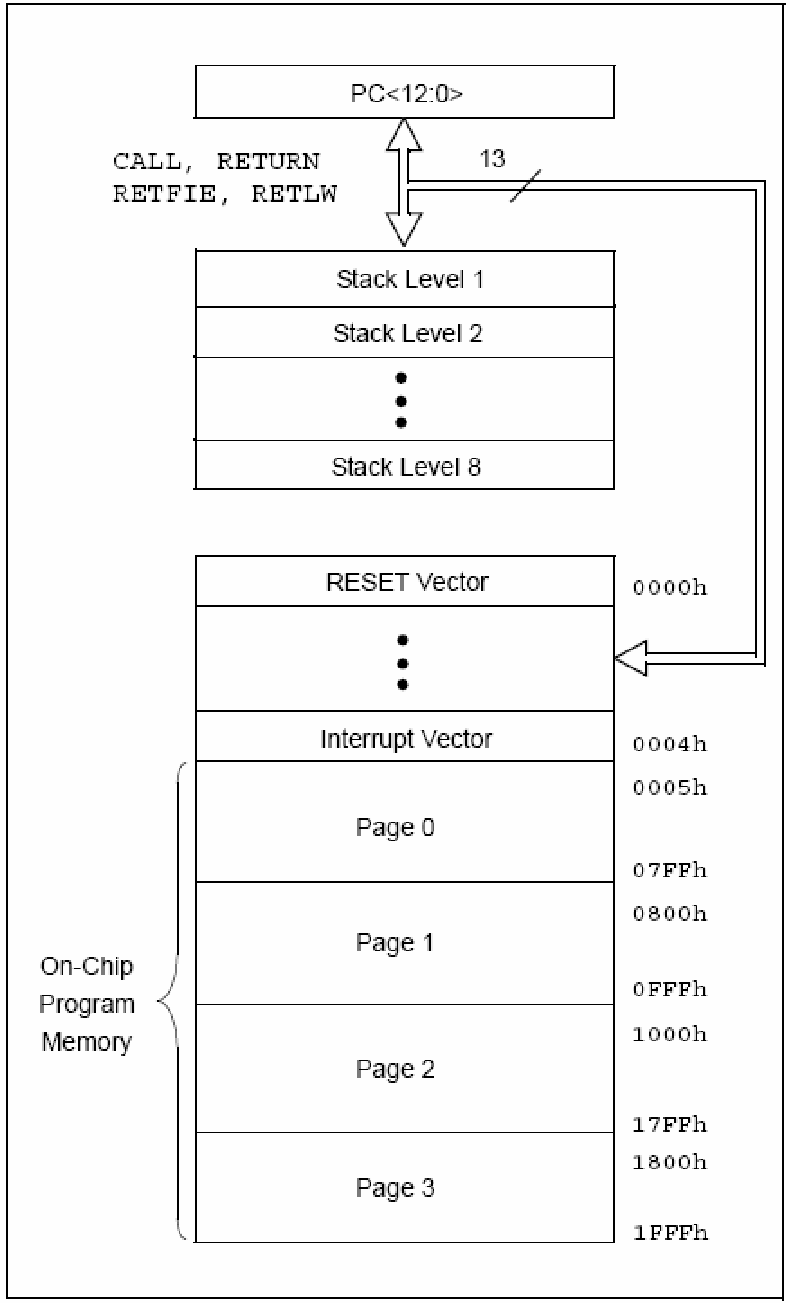


FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
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	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	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		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh

Unimplemented data memory locations, read as '0'. (shaded)

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.

2: These registers are reserved, maintain these registers clear.

STATUS (0x03, 0x83, 0x103, 0x183)								
Pin	7	6	5	4	3	2	1	0
Name	IRP	RP1	RP0	TO	PD	Z	DC	C

Instruction Set

- 35 instructions of 6 types:
- Set and clear bits
- Move (memory / working register)
- Arithmetic ops (add subtract)
- Logical ops (and, or, exor)
- Shift
- Goto, call, return, skip next instr on condition

Single Bit Operations			Flags Affected
bcf	PORTB,0	; clear bit 0 of Port B (pin #21)	
bsf	STATUS,C	; set the carry bit	
Clear & Move			
clrw		; Clear the W register	Z
clrf	TEMP1	; Clear the register with label TEMP1	Z
movlw	5	; load #5 into W	
movwf	TEMP1	; move W into TEMP1	
movf	TEMP1,W	; move TEMP1 into W	Z
swapf	TEMP1,F	; swap nibbles of TEMP1. Place the result in TEMP1.	
swapf	TEMP1,W	; swap nibbles of TEMP1. Place the result in W	
Increment & Decrement			
incf	TEMP1,F	; TEMP1 = TEMP1 + 1	Z
incf	TEMP1,W	; W = TEMP1 + 1	Z
decf	TEMP1,F	; TEMP1 = TEMP1 - 1	Z
decf	TEMP1,W	; W = TEMP1 - 1	Z
Bit Operations			
andlw	b'00000111'	; clear bits 7:4 of W	Z
movlw andwf	b'00000111' TEMP1,F	; clear bits 7:4 of TEMP1	Z
movlw iorwf	b'00000111' TEMP1,F	; (set bits 0:3 of TEMP	Z
movlw xorwf	b'00000111' TEMP1,F	; (toggle bits 0:3 of TEMP1)	Z
Addition / Subtraction			
addlw	5	; W = W + 5	C,DC,Z
addwf	TEMP1,F	; TEMP1 = TEMP1 + W	C,DC,Z
sublw	5	; W = 5 - W	C,DC,Z
subwf	TEMP1,F	; TEMP1 = TEMP1 - W	C,DC,Z
Rotate			
rlf	TEMP1,F	; nine-bit rotate left (carry = 9th bit)	C
rrf	TEMP1,W	; nine-bit rotate right (carry = 9th bit)	C
Conditional Branch			
btfsc	TEMP1,0	; test bit #0 of TEMP1	
btfss	STATUS,C	; test if a carry resulted	
incfsz	TEMP1,F	; TEMP1 = TEMP1 + 1. Skip next inst if zero.	
decfsz	TEMP1,F	; TEMP1 = TEMP1 - 1. Skip next inst if zero	
Program Control			
goto	There	; jump to label There	
call	There	; save the return address on the stack	
return		; pull the return address off the stack	
nop		; do nothing	