Embedded Microprocessors Group Project

Dr. L. K. Verma

Email: Dr.Lalit@gmail.com

Coventry University

Embedded Microprocessors Group Project

- Introduction about Module
- Assessment details
- Embedded systems
- C programming
- PIC controllers, development environment
- PIC I/O
- PWM control, Stepper Motors
- Sensors
- Analog-to-Digital Converter

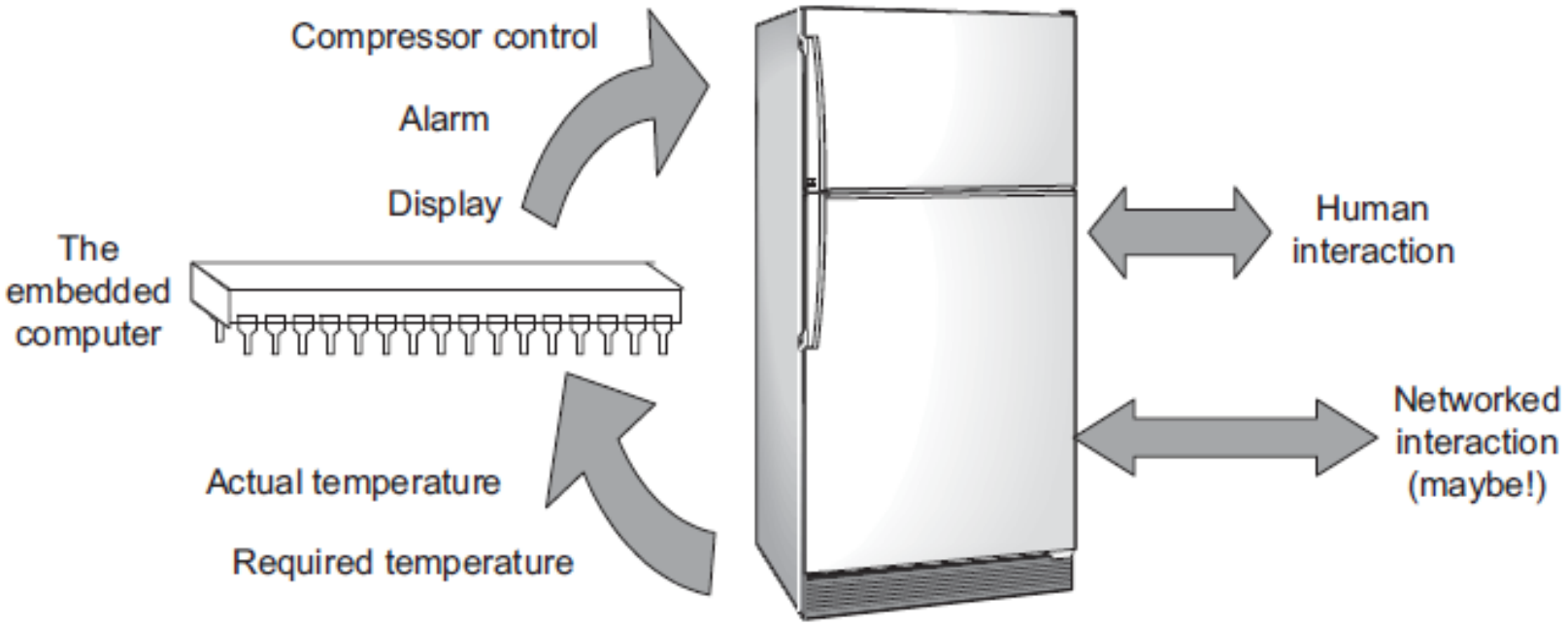
Coventry University

Embedded Microprocessors Group Project

- Key Board
- Interrupts
- PCB CAD design
- Project Management

Coventry University

Embedded systems



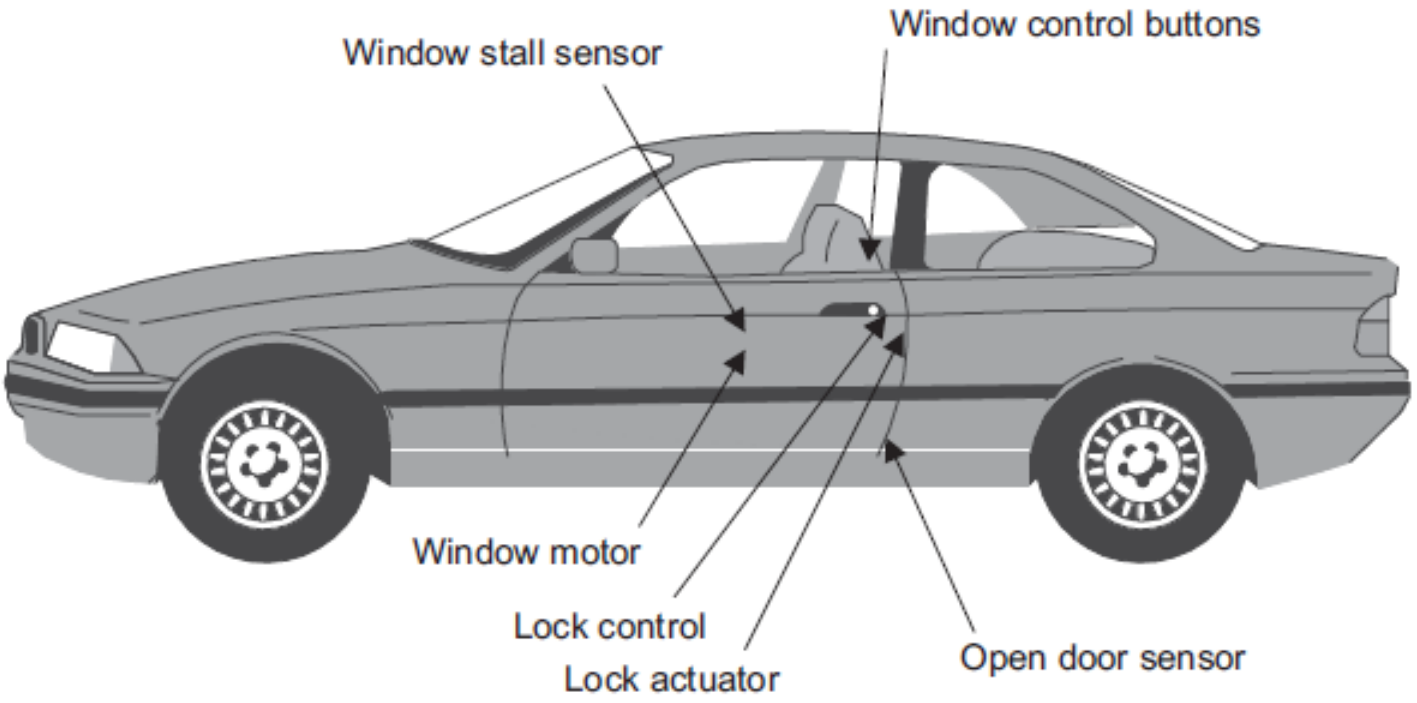
Coventry University

Some typical applications

- Automotive air bag systems
- Remote control
- Appliances – coffee pot, mixer, stove, refrigerator, dish washer, washer, dryer
- Major home systems – heating and cooling
- Cordless phones and cell phones
- Security systems
- TV, DVD player/recorder, DVR, PVR
- Sound system

Coventry University

Embedded systems



Coventry University

Typical automotive use

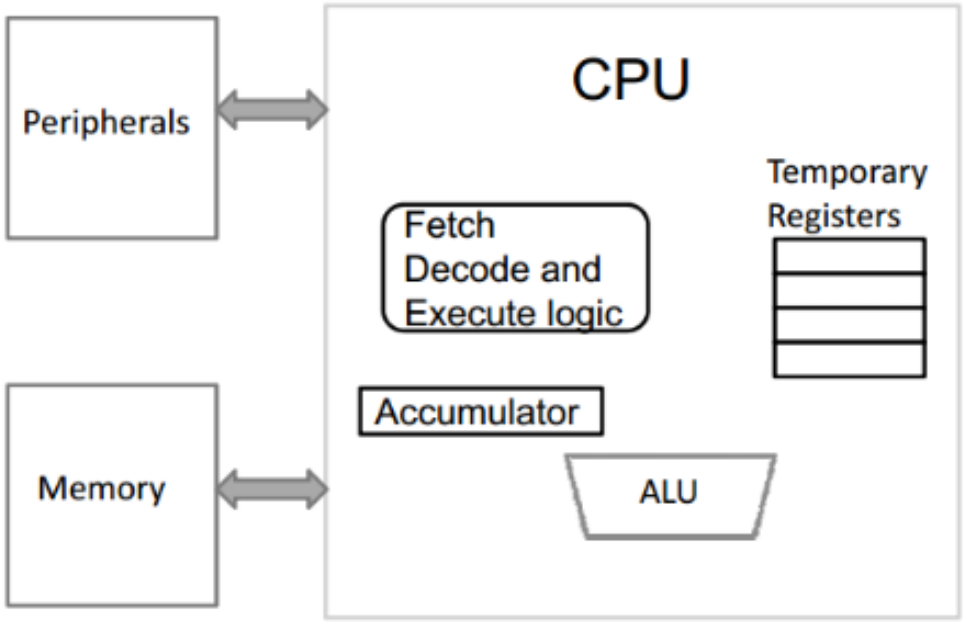
- Engine control – 32-bit microcontroller
  - Fuel flow, fuel mixture, valve timing, throttle body opening, spark timing
- Transmission control –16-bit microcontroller
- Audio system – 16-bit
- Antilock braking – 16-bit

Automotive today continued

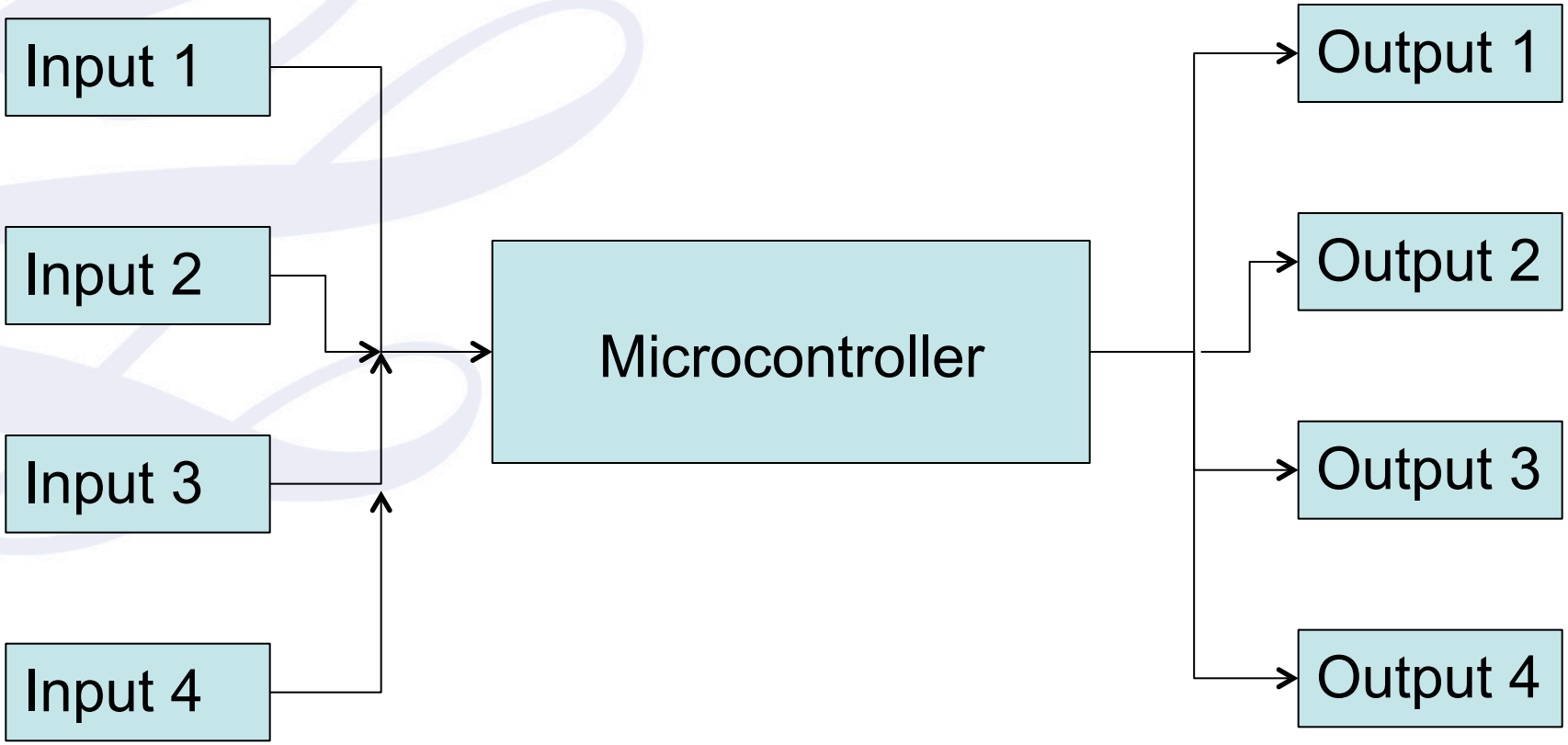
- Up to fifty 8-bit microcontrollers for functions of
  - Wiper control
  - Electric Mirrors
  - Air Bags
  - Fuel pump
  - Speedometer
  - Security system
  - Climate control system
  - Power windows …..

Microcontroller Solution

- Microcontroller features
  - CPU – processing unit
  - Non-volatile program memory
  - Re-settable non-volatile data memory (EEPROM)
  - RAM for data storage
  - Direct support for various input/output



Block Diagram



Coventry University

Identify Input/Outputs

- Dish washer
- Cell phones
- Robotic Vehicle
- Fingerprint based attendance system

Coventry University

Your Assignments

List of Assignments

Search YouTube with keyword 201AEE

- Assignment 1
  - https://www.youtube.com/watch?v=exP2Mm1N0wo
- Assignment 2
  - https://www.youtube.com/watch?v=7ovN5vVZdG8
  - https://www.youtube.com/watch?v=IzVmOO7Df4g

Coventry University

Introduction to Microchip Family

Description

- PIC – “Peripheral Interface Controller”
- Made by Microchip Technology
- Most popular by industry developers and hobbyists
  - Low cost (cents to dollars)
  - Availability
  - Extensive application notes
  - Serial programming

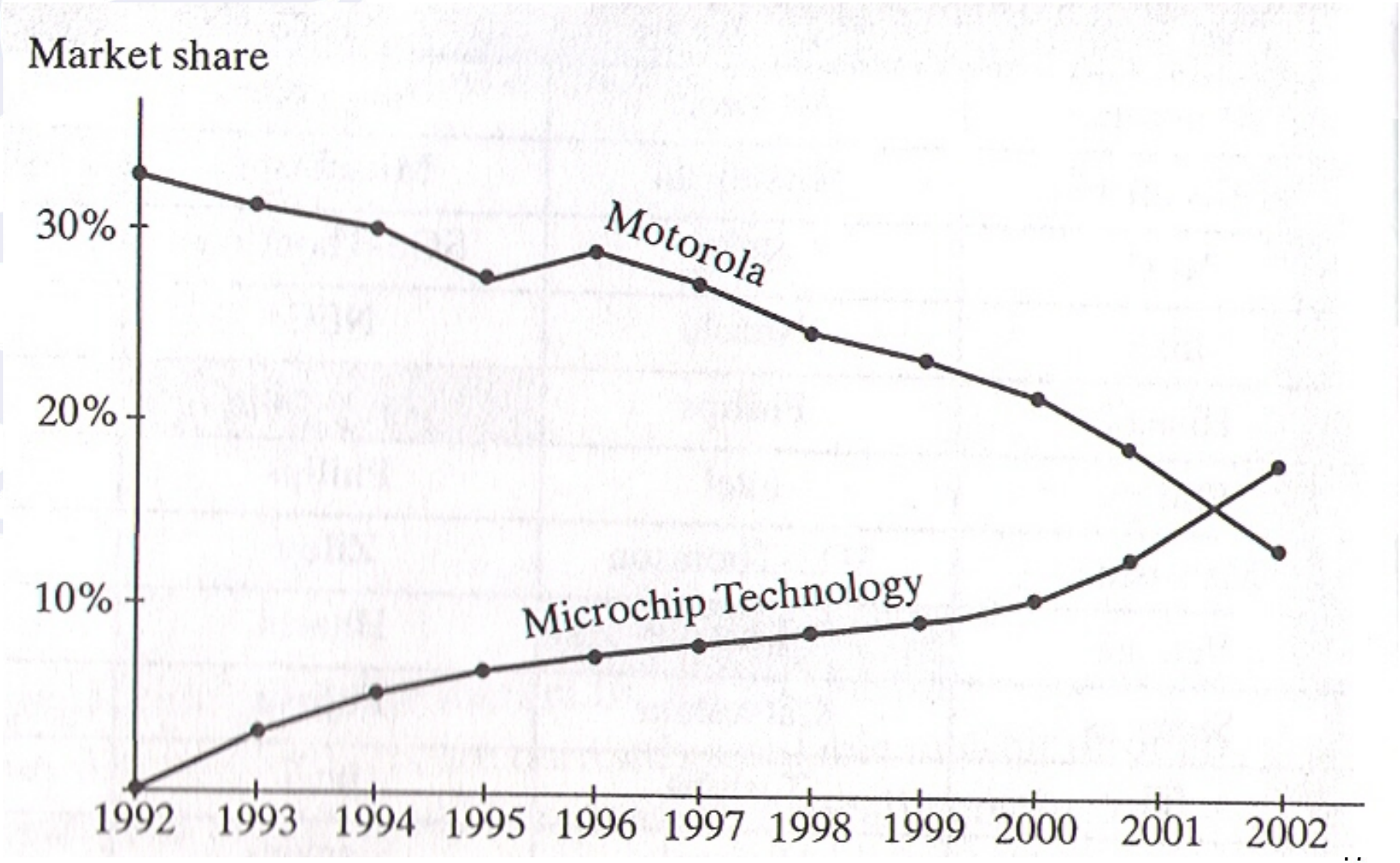
Coventry University

Microchip's position

1990 Rank	1993 Rank	1995/96 Rank	1997-01 Rank	2002 Rank
Motorola	Motorola	Motorola	Motorola	Microchip
Mitsubishi	Mitsubishi	Mitsubishi	Microchip	Motorola
NEC	NEC	SGS-Thomson	ST-Micro	ST-Micro
Intel	Hitachi	NEC	NEC	NEC
Hitachi	Philips	Microchip	Philips	Atmel
Philips	Intel	Philips	Atmel	Sunplus
Matsushita	SGS-Thomson	Zilog	Hitachi	Hitachi
National	Microchip	Hitachi	Toshiba	Fujitsu
Siemens	Matsushita	Fujitsu	Samsung	Philips
TI	Toshiba	Intel	Elan	Toshiba
Sharp	National	Siemens	Zilog	Mitsubishi
Oki	Zilog	Toshiba	Matsushita	Samsung
Toshiba	TI	Matsushita	Infineon	Elan
SGS-Thomson	Siemens	TI	Fujitsu	Winbond
Zilog	Sharp	National	Mitsubishi	Zilog
Matra MHS	Oki	Temic	Sanyo	Sanyo
Sony	Sony	Sanyo	Winbond	Matsushita
Fujitsu	Sanyo	Ricoh	National	Infineon
AMD	Fujitsu	Oki	Sony	Holtek
Microchip	AMD	Sharp	Holtek	National

Figure 1-1 Worldwide market share for producers of 8-bit microcontrollers, sorted by units shipped. (2002 Microcontroller Market Share and Unit Shipments, Tom Starnes, Gartner Dataquest, June 2003)

Microchip and microcontroller use



Microchip and microcontroller use

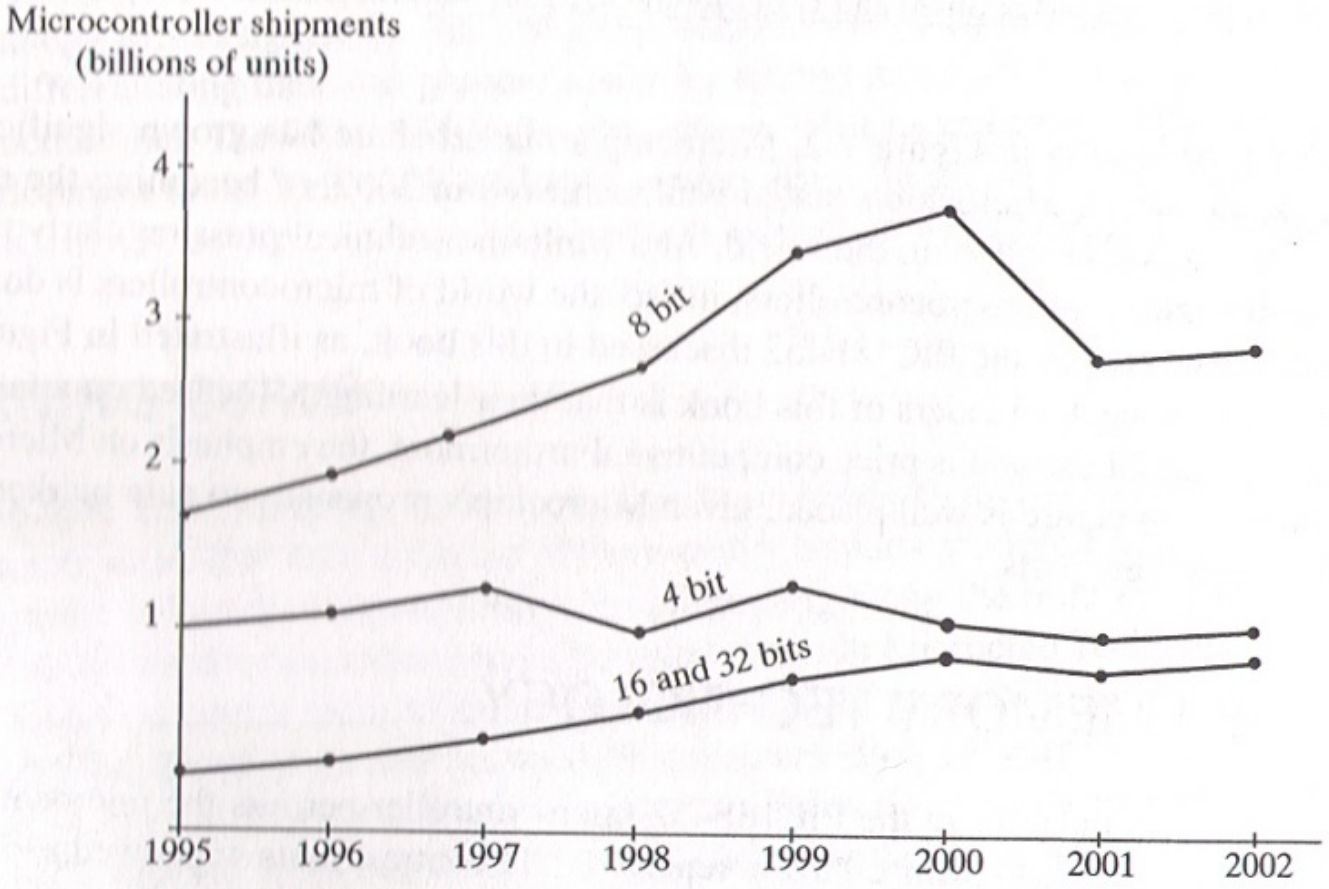


Figure 1-3 Microcontroller unit shipments per year, as distinguished by data word length (Dataquest).

4-bit through 32-bit

Product Families

- 8-bit
  - PIC10
    - 6 pins, 2 MIPs, 768 Bytes
  - PIC12
    - 8 pins, 5 MIPs, 3584 Bytes
  - PIC16
    - 14 - 80 pins, 10 MIPs, 14336 Bytes
  - PIC18
    - 18 - 128 pins, 12 MIPs, 131072 Bytes

Microcontroller choice parameters

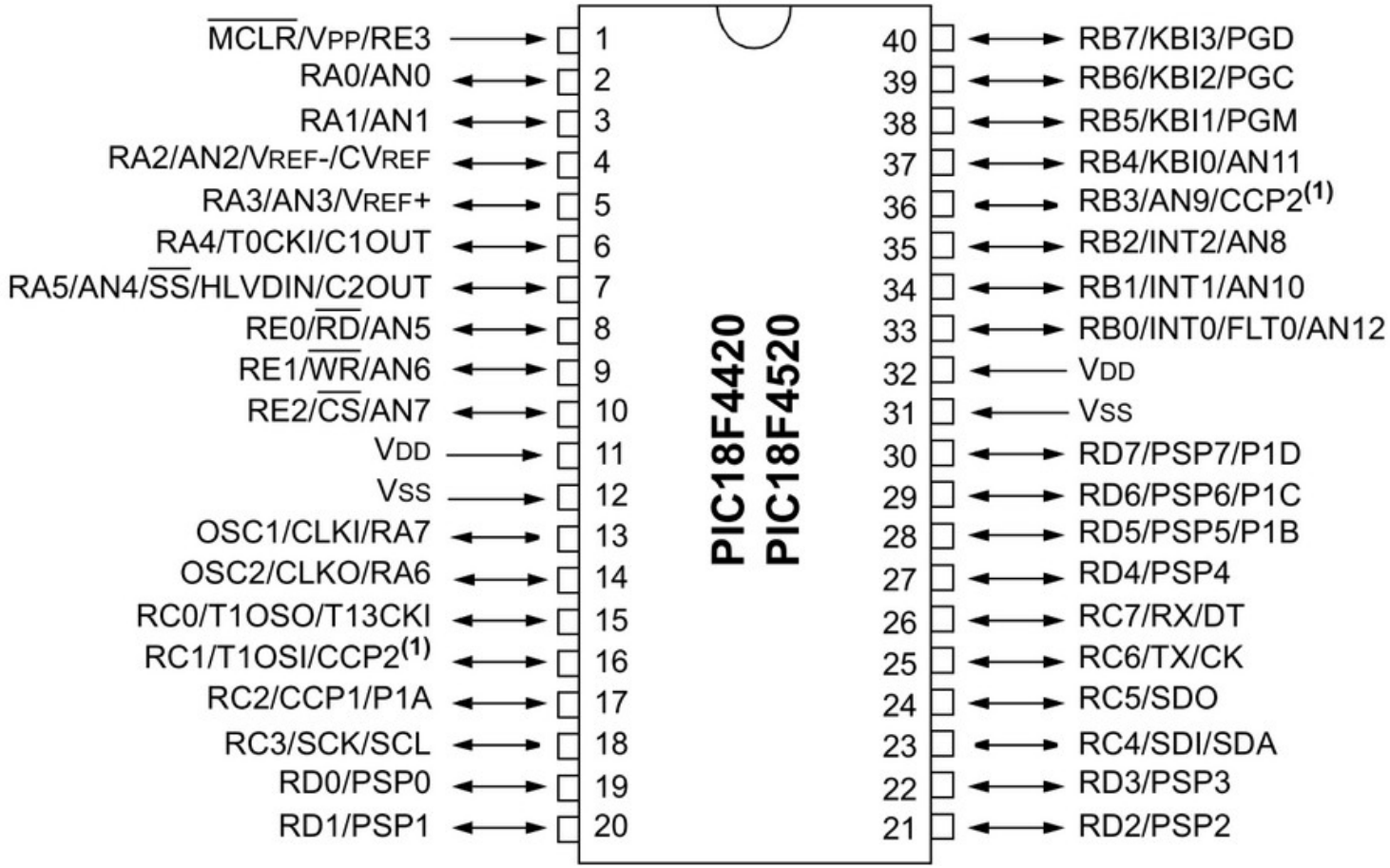
- Number of I/O pins
- Amount of program and data memory
- Speed
- Timer Resources
- Interrupt control
- Robustness
- Error recovery – watchdog timers
- Power, I/O expansion, math support

Coventry University

	Baseline Architecture	Mid-Range Architecture	Enhanced Mid-Range Architecture	PIC18 Architecture
Pin Count	6-40	8-64	8-64	18-100
Interrupts	No	Single interrupt capability	Single interrupt capability with hardware context save	Multiple interrupt capability with hardware context save
Performance	5 MIPS	5 MIPS	8 MIPS	Up to 16 MIPS
Instructions	33, 12-bit	35, 14-bit	49, 14-bit	83, 16-bit
Program Memory	Up to 3 KB	Up to 14 KB	Up to 28 KB	Up to 128 KB
Data Memory	Up to 138 Bytes	Up to 368 Bytes	Up to 1.5 KB	Up to 4 KB
Features	•Comparator •8-bit ADC •Data Memory •Internal Oscillator	In addition to Baseline: •SPI/I²C™ •UART •PWMs •LCD •10-bit ADC •Op Amp	In addition to Mid-Range: •Multiple Communication Peripherals •Linear Programming Space •PWMs with Independent Time Base	In addition to Enhanced Mid-Range: •8x8 Hardware Multiplier •CAN •CTMU •USB •Ethernet •12-bit ADC
Families	PIC10, PIC12, PIC16	PIC12, PIC16	PIC12FXXX, PIC16F1XX	PIC18

8-bit architecture

PIC18F4520 pinning



Become expert at reading datasheets

Coventry University

PIC Microcontroller Peripherals

The 18F452 PIC has the following peripherals:

Data ports A (6-bit), B (8-bit), C (8-bit), D (8-bit), E (3-bit)

Timer/counter modules 0 (8-bit), 1 (16-bit), 2 (8-bit), 3 (16-bit)

CCP/PWM modules (2)

I^2C/SPI serial port

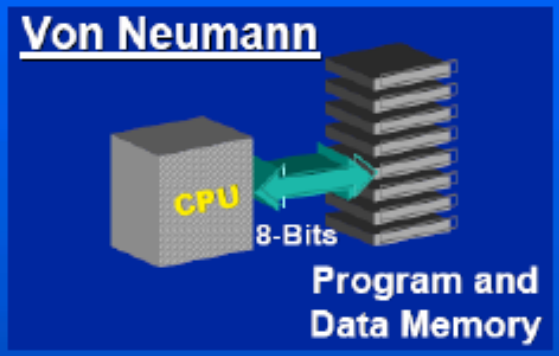
USART (RS-232, RS-485)

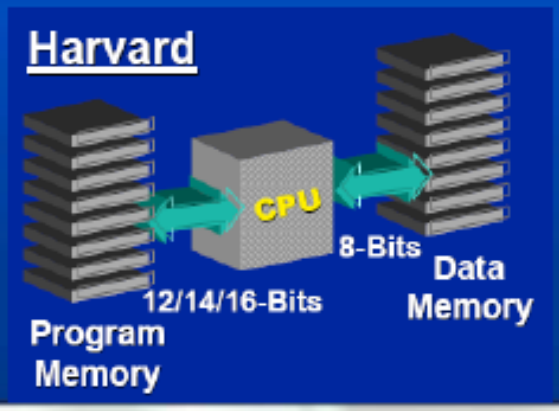
Analogue-to-digital converter (10-bit) with 10 way input multiplexer

EEPROM (256 byte)

Coventry University

Processor Architecture



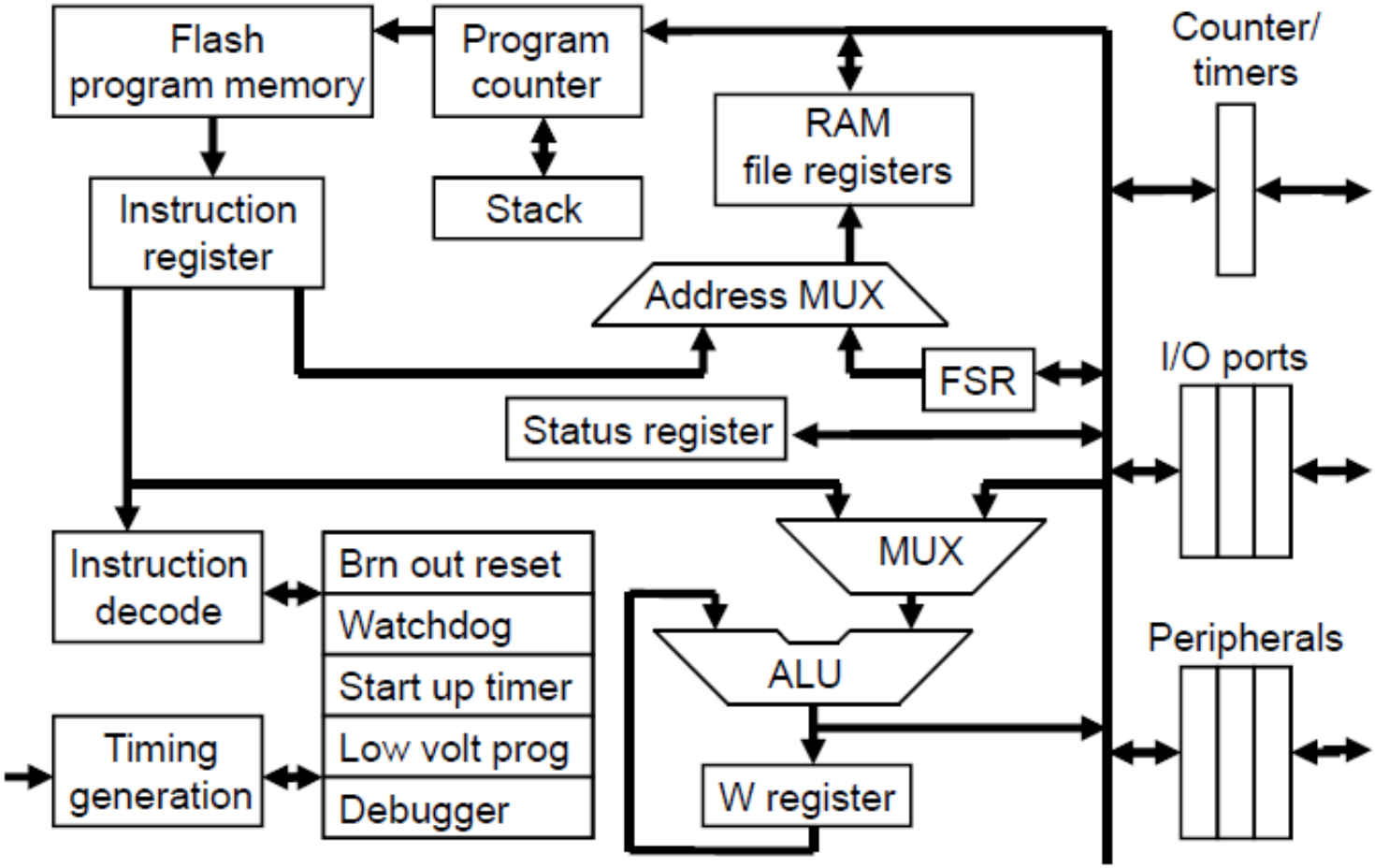


- Von Neumann
  - Single data bus for instructions and data
  - Motorola HC11
- Harvard
  - Separate memory spaces for data and instruction
  - Fetch instruction and data simultaneously
  - Microchip PIC

MICROCHIP

Coventry University

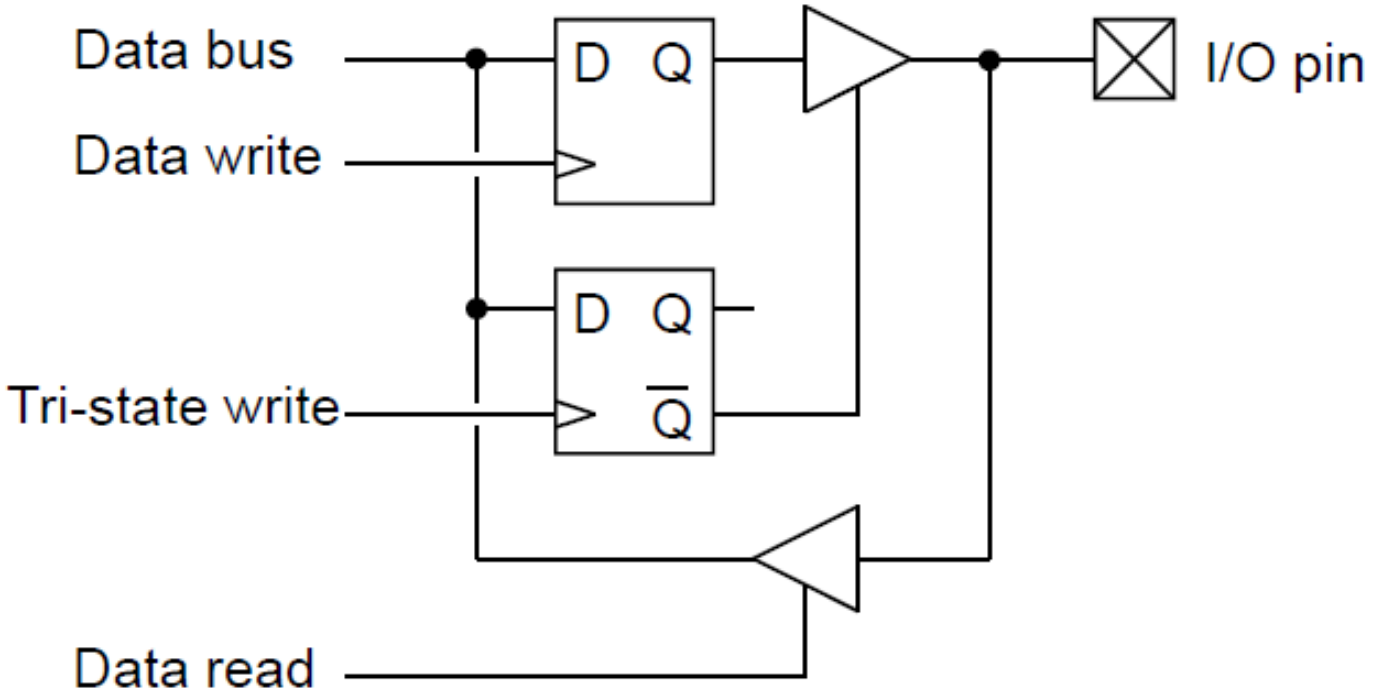
PIC Microcontroller Architecture



Coventry University

Data Ports

Simplified diagram representing a single data I/O pin of one of the ports A-E:



Coventry University

Clock Generator

PICs use a fully static design so that any clock frequency up to the specified maximum can be used

There are 4 possible clock configurations:

- external clock (eg crystal oscillator module)
- self-oscillating with external crystal or ceramic resonator
- external or self-oscillating with phase-locked loop
- self-oscillating with external RC

In practice the choice will normally be a compromise between cost and clock speed or clock stability

Reset

A reset puts the PIC in a well-defined initial state so that the processor starts executing code from the first instruction

Resets can result from:

- external reset by MCLR pulled low
- reset on power-up
- reset by watchdog timer overflow
- reset on power supply brown-out

Reset can be used as a last resort for recovering from some catastrophic software event but all current data will be lost

Coventry University

Central Processing Unit

The CPU fetches instructions from memory, decodes them, and passes them to the ALU for execution

The arithmetic logic unit (ALU) is responsible for adding, subtracting, shifting and performing logical operations

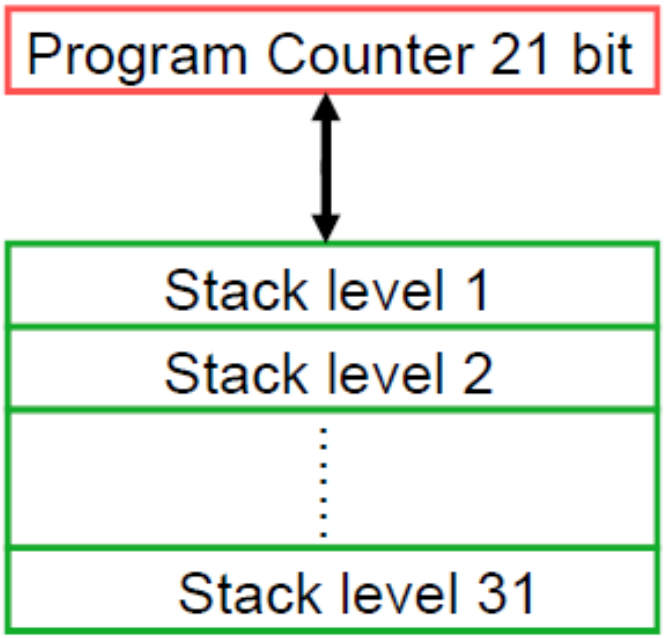
The ALU operates in conjunction with:

- a general-purpose register called the W register
- an f register that can be any location in data memory
- literals embedded in the instruction code

Coventry University

Memory Organization

A 31-level stack stores the return address during interrupts and subroutine calls

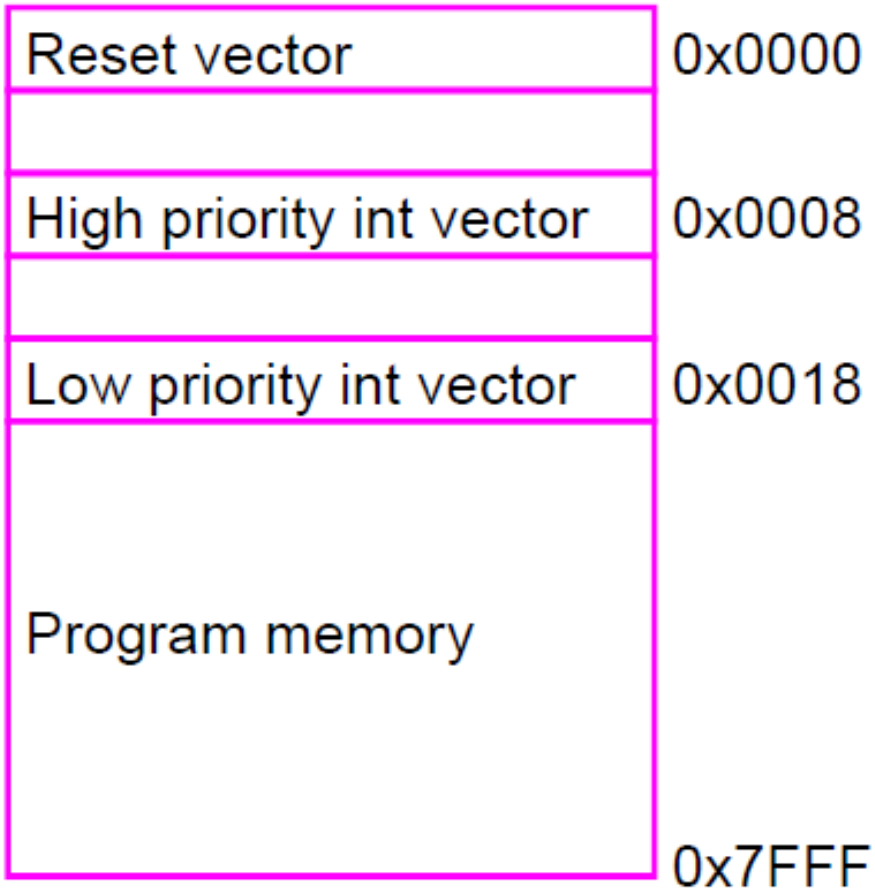


Coventry University

Memory Organization – Program

Program memory contains the Reset and Interrupt vectors

The PIC18F452 has 32k (0x8000) locations of program memory

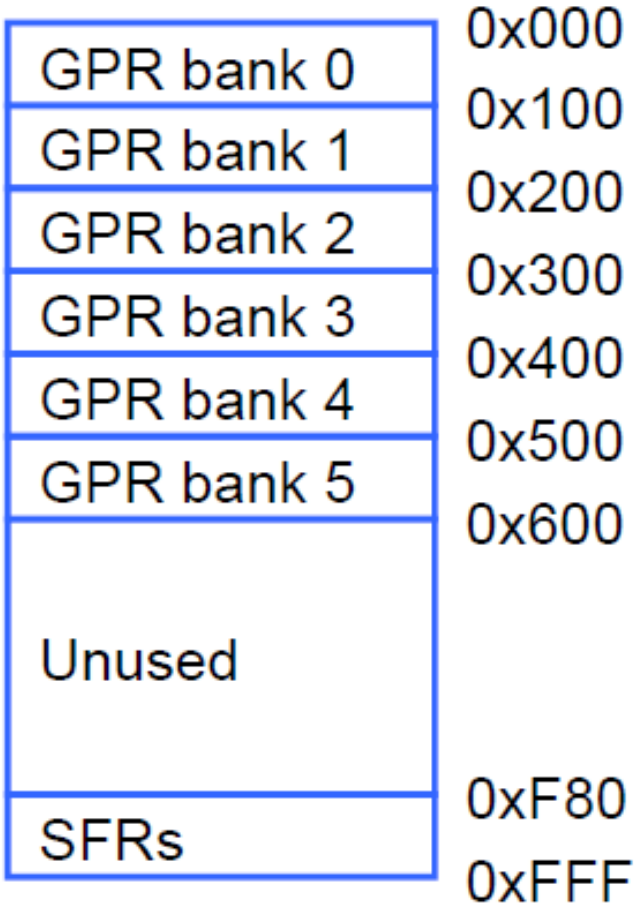


Coventry University

Memory Organization – Data

Data memory contains general purpose registers (GPRs) and special function registers (SFRs)

The PIC18F452 has 1536 (0x600) locations of GPR data memory



Coventry University

Memory Organization – SFR

The memory block 0xF80 to 0xFFF (128 locations) references special function registers (SFRs)

Some of the SFRs are shown here

Port A	0xF80
Port B	0xF81
Port C	0xF82
Port D	0xF83
Port E	0xF84
⋮	
Tris A	0xF92
Tris B	0xF93
Tris C	0xF94
Tris D	0xF95
Tris E	0xF96

SPBRG	0xFAF
⋮	
Timer1L	0xFCE
Timer1H	0xFCF
⋮	
Timer0L	0xFD6
Timer0H	0xFD7
⋮	
Wreg	0xFE8
⋮	
StkPtr	0xFFC

Coventry University

PIC Instruction Set

The PIC instruction set has a small number of simple (RISC) instructions

PIC16 series: 35 instructions coded into 14 bits

PIC 18 series: 59 instructions coded into 16 bits

PIC 24 series: 71 instructions coded into 24 bits

Most instructions are executed in one instruction cycle which corresponds to 4 clock cycles

Thus a PIC operating at 40 MHz clock frequency will have an instruction rate of 10 MIPS.

Coventry University

PIC 18Fxxx Instruction Set

Most PIC 18Fxxx instructions occupy a single 16-bit program memory location

Each instruction consists of an opcode and one or more operands

The instruction set is highly orthogonal and can be partitioned:

- 31 byte-oriented file register operations
- 5 bit-oriented file register operations
- 23 control instructions
- 10 literal instructions
- 8 data memory – program memory operations

PIC 18Fxxx Instruction Set

Byte-oriented file register operations :

ADDWF	Add W and f: result in W or f
CLRF	Clear f
DECF	Decrement f
MOVF	Move contents of f to f or W

Bit-oriented file register operations:

BCF	Clear bit in f
BTFSC	Test bit in f; skip if clear

Coventry University

PIC 18Fxxx Instruction Set

Control instructions :

BRA	Branch unconditionally
CALL	Call subroutine (function)
RETURN	Return from subroutine (function)
BNZ	Branch if not zero

Literal instructions :

MOVLW	Move literal to W
ADDLW	Add literal to W

Data memory – program memory operations:

TBLRD*+	Table read with post-increment

Coventry University

Status Register

The 8-bit status register is set during arithmetic operations

-	-	-	N	OV	Z	DC	C

N	Negative bit - result of arithmetic operation was negative
OV	Overflow bit – overflow occurred for signed arithmetic
Z	Zero bit - result of arithmetic operation was zero
DC	Digit Carry bit – carry out from 4th low order bit of result
C	Carry bit – carry out from most-significant bit of result

The bits of the status register can then be used in conditional branches, for example:

BNZ	Branch if Not Zero
BOV	Branch of OVerflow

Coventry University

PIC18F4520

- Links:
  - http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010297
- PIC18 explorer board

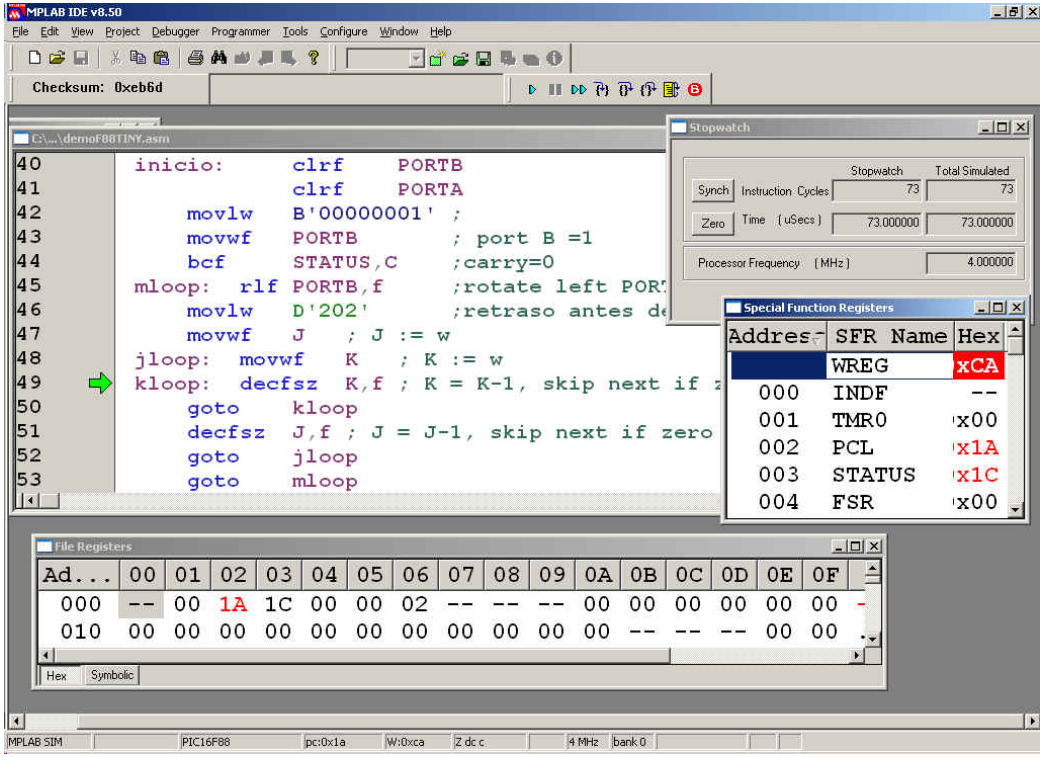
Coventry University

Development Tools

- MPLAB IDE
  - Integrated Development Environment
  - Free Download (http://www.microchip.com)
- MPLAB SIM – Simulate Program in MPLAB
  - Included with MPLAB
- ICD2 – InCircuit Debugger
  - Debugger/Programmer
  - $159
- PICSTART Plus
  - Programmer for all DIP packages
  - $199

Programming Tools – MPLAB IDE

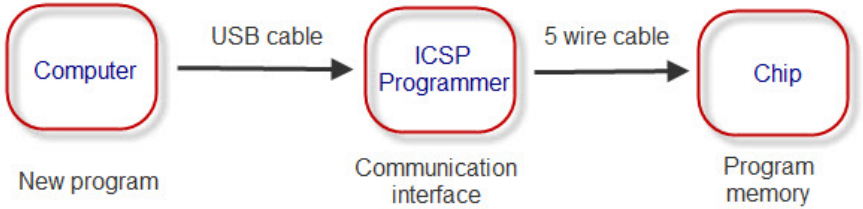
- Basic based environments are available, but don't offer the functionality of C
- Third party IDE's might be preferred due to an enhanced software library or debugging tools



http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

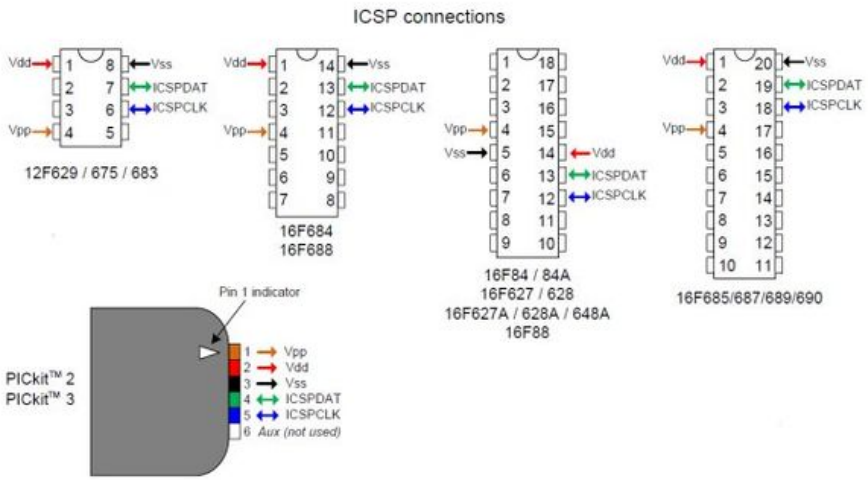
Device Programmer

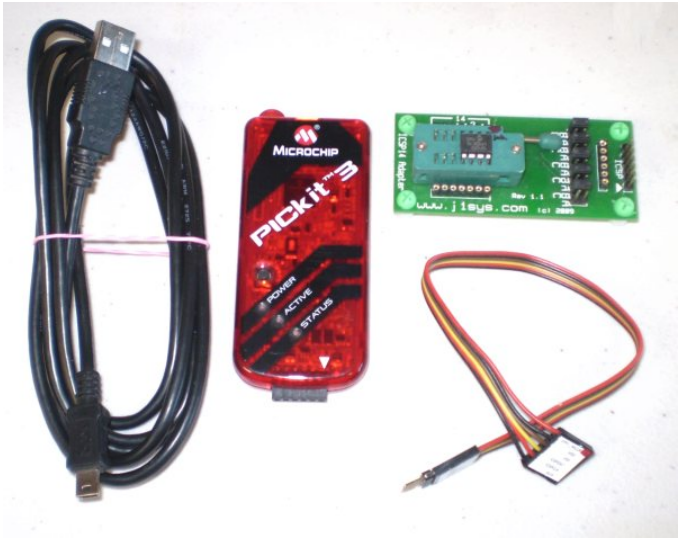
- Need device to store machine code into PIC's memory (EEPROM or Flash)
- Can be external device, but ICSP is easier:
  - Don't have to remove chip from its circuit
  - Provides interface between computer (USB) and PIC
  - Specific to circuit (due to interconnect scheme and surrounding circuit)
  - Communication protocol requires 5 signals



http://en.wikipedia.org/wiki/PIC_microcontroller

Device Programmer





Five Signals:

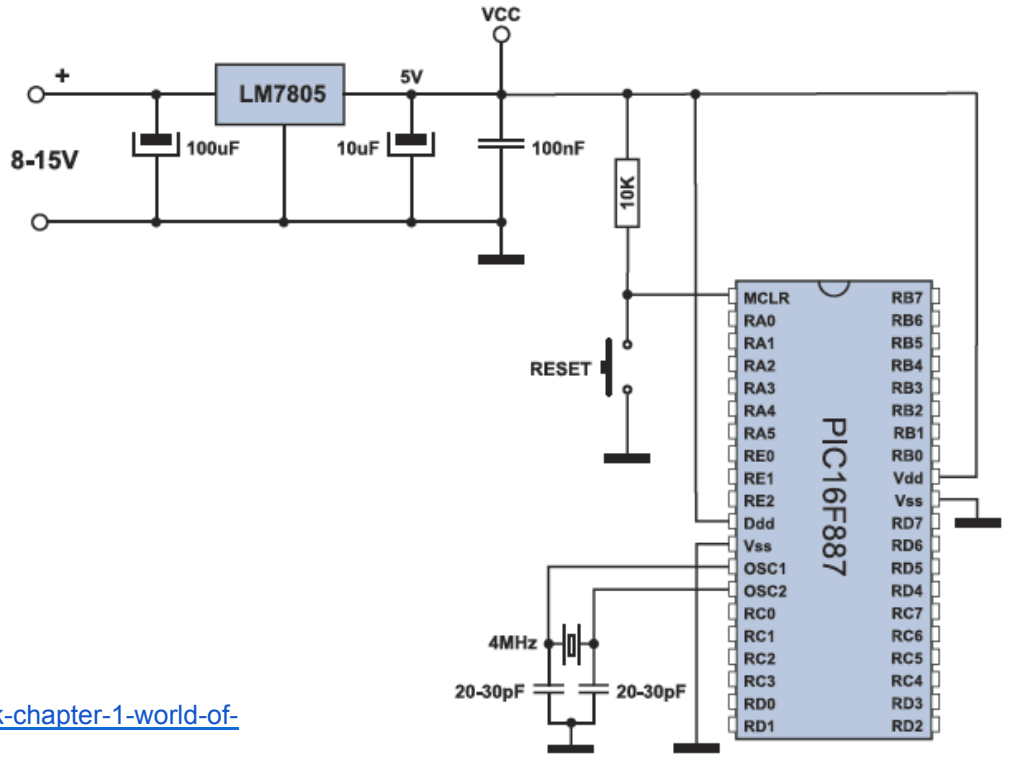
- Vpp (programming voltage)
- Vdd (power)
- Vss (ground)
- IC SPCLK (clock)
- IC SPDAT (data)

http://en.wikipedia.org/wiki/PIC_microcontroller

Device Programmer

PIC can be bread-boarded, with the following important connections:

- Power
- Ground
- Reset signal
- Crystal (oscillator)



http://www.mikroe.com/eng/chapters/view/74/pic-basic-book-chapter-1-world-of-microcontrollers/

Introduction to PIC C programming

http://www.ermicro.com/blog/?p=365

LED example: code

```
//LED example program written for
//PIC programming tutorial. From //(
http://seniord.ece.iastate.edu/dec0604/index_files/tutorialDec0604.pdf)
//standard include files
#include <stdlib.h>
#include <pic.h>
#include “delay.h”
//main function
        void main()
        {
                PORTA = 0x00; //set RA0-RA5 low
                TRISA = 0x00; //set PORTA to output
                //superloop
                while(1)
                {
                        PORTA = ~PORTA;
                        DelayMs(250);
                }
        }
```

Development Boards

Prepackaged boards come with a multitude of peripherals for development and debugging:

- Programmer
- User I/O: buttons, port pinouts, LED's
- Displays: LCD and seven segment
- Power
- Serial connection interface