

# Introduction and Motivation

## *Microcontrollers vs. microprocessors*

uC: A complete computer system optimized for h/w control that encapsulates processor, memory, i/o on a single chip.

A uC is a uP with peripherals and memory.

uP: Z80, 8088, 68040, 68HC09, Pentium4

uC: 8051, AVR, PIC, Rabbit

Microcontrollers as:

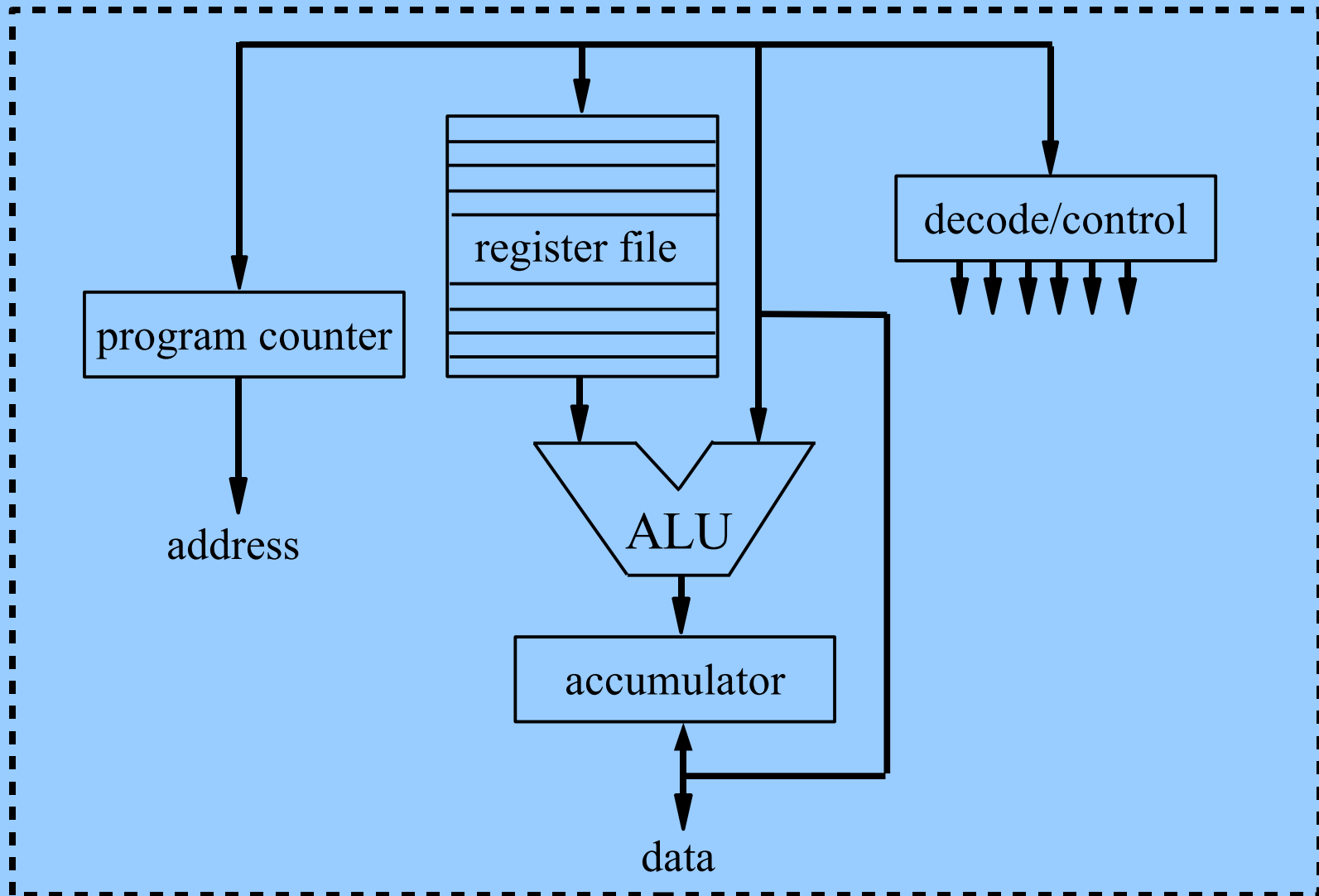
- logic devices
- reusable hardware

RISC vs CISC....whats the difference?

# Introduction and Motivation

## *Microcontrollers vs. microprocessors*

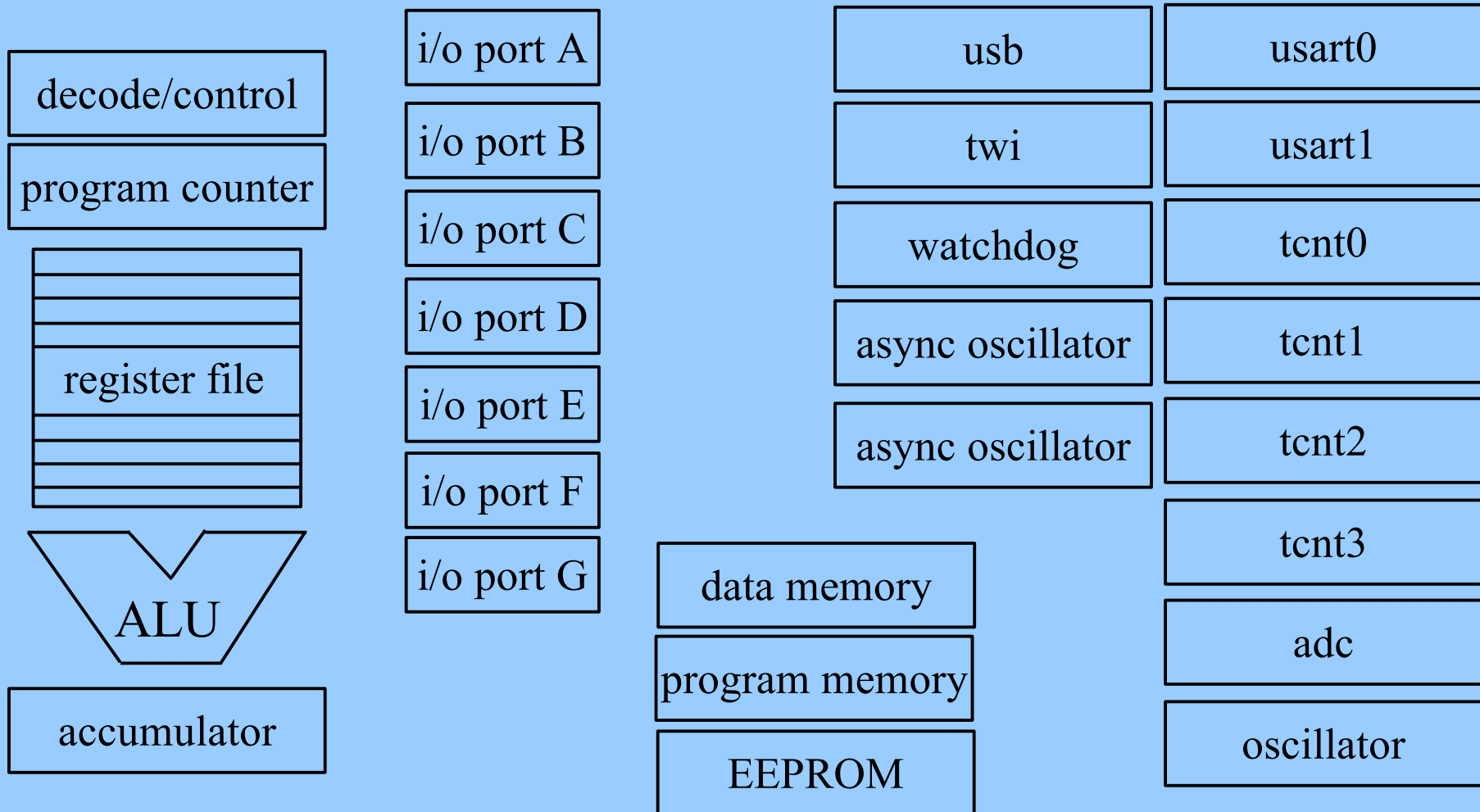
Microprocessors can be fairly straightforward.....



# Introduction and Motivation

## *Microcontrollers vs. microprocessors*

While microcontrollers many different specialized internal units.....



# Introduction and Motivation

## *Microcontrollers vs. microprocessors*

Microcontrollers usually lack cache memory – Why?

Microcontrollers are usually very RISC-like - Why?

Microcontrollers don't have very high clock rates – Why?

Microcontrollers have separate program and data memory – Why?

Microcontrollers have limited stack depth – Why?

Microcontrollers have a different instruction set – Why?

Microcontrollers have flash program memory – Why?

Microcontrollers typically have internal EEPROM - Why?

# Introduction and Motivation

*Microcontrollers change the way of solving hardware problems.*

With so much built-in peripheral logic included on a uC, we can think of them as complex logic chips.

Suppose you want to design a 4-digit digital voltmeter with LED display.

Do a quick back of the envelope design. (10min)

# Introduction and Motivation

*Microcontrollers change the way of solving hardware problems*

How many chips did your discrete voltmeter take?

At \$.075/chip what was the part cost?

How many wires did it take?

How big a PCB would it take?

What if you needed it to be IR remote controlled and have a LED display. Would you need to start on a new design?

# Introduction and Motivation

*Microcontrollers change the way of solving hardware problems*

Either design, and 100's of others could be done with a single uC.

Add a pushbutton and it could be a frequency counter.

Add a shaft encoder and it can be a audio frequency generator

Add a RF transceiver and it could be a Mars rover!

# Introduction and Motivation

## *What microcontrollers are out there?*

### 8051

- the grandpa of uCs
- 20+ years old and going strong
- installed code base very large
- 12 cycles/instruction, some now run at 4 cycles/instruction
- clock frequencies up to 50Mhz
- dog ugly architecture (but so is x86!)

### PIC

- cockroach of uC's (16F84, 16F628)
- low power, medium speed,
- very large product family, many variations
- ugly architecture / register banks, single accumulator

# Introduction and Motivation

*What microcontrollers are out there?*

AVR

- includes ATtiny, AT23xx through Mega series
- modern RISC architecture, mostly single cycle instruction
- optimized for C execution
- reasonably fast with emphasis on low power operation
- clock frequencies up to 16Mhz
- clean architecture (looks like a crippled MIPS uP)

Rabbit Semiconductor

Z8 Encore! (Zilog Z80/Z8 derivatives)

Hitachi H8

Motorola HC908, Coldfire (68k derivatives)

Many others