Introduction to Embedded Systems

55:036

January 18,2007

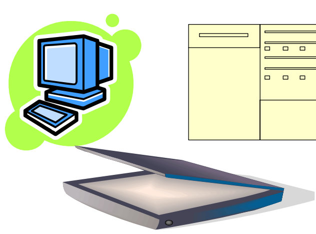
Acknowledgement

- The bulk of the material in this lecture is adapted from:

  Embedded System Design– A Unified Hardware/Software Introducton, by Frank Vahid and Tony Givargis, John Wiley & Sons Inc., 2002

What is a Computer?

- Most of us think of "desktop" computers
  - PC's
  - Laptops
  - Mainframes
  - Servers



- But, there is another kind of computing system that is far more common

Embedded Systems

- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps 50 per household and per automobile

Embedded Systems are Everywhere

Picture is from the cover of **Embedded Systems Design, A Unified Hardware/Software Approach**, by Frank Vahid and Tony Givargis

A "short list" of embedded systems

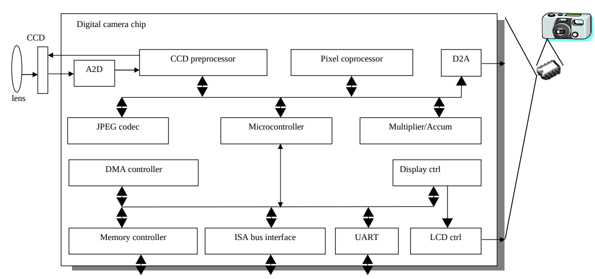
- Anti-lock brakes
- Auto-focus cameras
- Automatic teller machines
- Automatic toll systems
- Automatic transmission
- Avionic systems
- Battery chargers
- Camcorders
- Cell phones
- Cell-phone base stations
- Cordless phones
- Cruise control
- Curbside check-in systems
- Digital cameras
- Disk drives
- Electronic card readers
- Electronic instruments
- Electronic toys/games
- Factory control
- Fax machines
- Fingerprint identifiers
- Home security systems
- Life-support systems
- Medical testing systems
- Modems
- MPEG decoders
- Network cards
- Network switches/routers
- On-board navigation
- Pagers
- Photocopiers
- Point-of-sale systems
- Portable video games
- Printers
- Satellite phones
- Scanners
- Smart ovens/dishwashers
- Speech recognizers
- Stereo systems
- Teleconferencing systems
- Televisions
- Temperature controllers
- Theft tracking systems
- TV set-top boxes
- VCR's, DVD players
- Video game consoles
- Video phones
- Washers and dryers

Today, almost all nontrivial electronic systems include one or more embedded processors

Embedded Systems vs. Desktop Computing

- Most Embedded Systems are single-functioned
  - Executes a single program, repeatedly
- Generally, Embedded Systems are tightly-constrained
  - Low cost, low power, small, fast, etc.
- Most Embedded Systems are reactive and real-time
  - Continually react to changes in the system's environment
  - Must compute results in "real-time"

An Embedded System Example: Digital Camera



Embedded Design Challenge: Optimizing Design Metrics

- Obvious design goal:
  - Construct an implementation with desired functionality
- Key design challenge:
  - Simultaneously optimize numerous design metrics
- Design metric
  - A measurable feature of a system's implementation
  - Optimizing design metrics is a key challenge

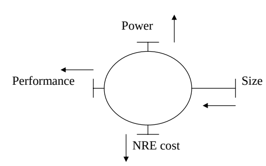
Embedded Systems Design Metrics

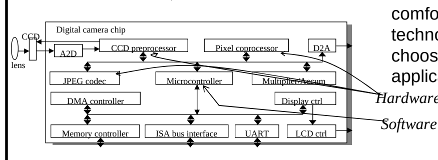
- Common metrics:
  - Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
  - NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
  - Size: the physical space required by the system
  - Performance: the execution time or response time of the system
  - Memory: The amount of memory required to hold the program and data
  - Power: the amount of power consumed by the system

Embedded System Design Metrics (Continued)

- Common metrics (continued)
  - Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost
  - Time-to-prototype: the time needed to build a working version of the system
  - Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
  - Maintainability: the ability to modify the system after its initial release
  - Robustness: System stability and reliability
  - Safety: Assurance that the system will not expose people to dangers

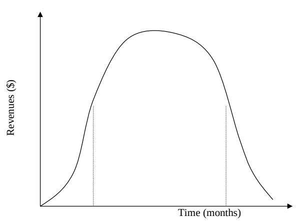
Design Metrics May Be At Odds With One Another





- Expertise with both **software and hardware** is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints

Time-to-market: a Critical Design Metric



- Time required to develop a product to the point it can be sold to customers
- Market window
  - Period during which the product would have highest sales
- Average time-to-market constraint is about 8 months
- Delays can be costly

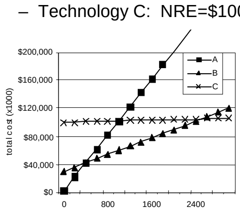
NRE and Unit Cost Metrics

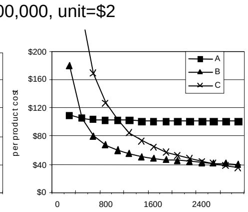
- Costs:
  - Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
  - NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
  - *total cost = NRE cost + unit cost * # of units*
  - *per-product cost = total cost / # of units*
    = (NRE cost / # of units) + unit cost
- Example
  - NRE=$2000, unit=$100
  - For 10 units
    - total cost = $2000 + 10*$100 = $3000
    - per-product cost = $2000/10 + $100 = $300

Amortizing NRE cost over the units results in an additional $200 per unit

NRE and Unit Cost Metrics

- Compare technologies by costs -- best depends on quantity
  - Technology A: NRE=$2,000, unit=$100
  - Technology B: NRE=$30,000, unit=$30
  - Technology C: NRE=$100,000, unit=$2





- But, must also consider time-to-market

The Performance Design Metric

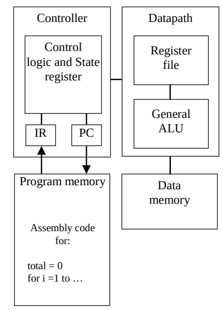
- Widely-used performance measure, widely-abused
  - Clock frequency, instructions per second – not good measures
  - Digital camera example – a user cares about how fast it processes images, not clock speed or instructions per second
- Latency (response time)
  - Time between task start and end
  - e.g., Camera's A and B process images in 0.25 seconds
- Throughput
  - Tasks per second, e.g. Camera A processes 4 images per second
  - Throughput may involve more than just latency due to concurrency, e.g. Camera B may process 8 images per second (by capturing a new image while previous image is being stored).
- *Speedup* of B over A = B's performance / A's performance
  - Throughput speedup = 8/4 = 2

Embedded Processor Technologies

- General-Purpose
- Single-Purpose
- Application-Tailored

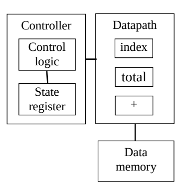
General-Purpose Processors

- Programmable device used in a variety of applications
  - Also known as “microprocessor”
- Features
  - Program memory
  - General datapath with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- Intel “Pentium” the most well-known, but there are hundreds of others



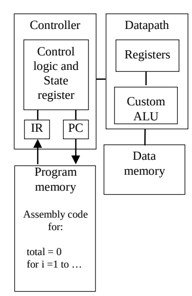
Single-Purpose Processors

- Digital circuit designed to execute exactly one program
  - a.k.a. coprocessor, accelerator or peripheral
- Features
  - Contains only the components needed to execute a single program
  - No program memory
- Benefits
  - Fast
  - Low power
  - Small size



Application-Tailored Processors

- Programmable processor optimized for a particular class of applications having common characteristics
  - Compromise between general-purpose and single-purpose processors
- Features
  - Program memory
  - Optimized datapath
  - Special functional units
- Benefits
  - Some flexibility, good performance, size and power



A Key Player in Embedded Design: The Microcontroller

- Compromise between general-purpose and application-tailored processor
- Simple processor architecture
  - reduced instruction set and functionality
  - small data path (often only 4 or 8 bits vesus 32 or 64 bits for typical general purpose processor
- On-board memory (volatile and non-volatile)
- Multiple on-chip devices to support embedded applications
  - timers
  - digital I/O
  - serial I/O
  - support for various interfacing protocols—e.g. I^2C
- Available in many different configurations, performance levels, etc.

Processor Comparison

Processor	Clock speed	Periph.	Bus Width	MIPS	Power	Trans.	Price
General Purpose Processors							
Intel PIII	1GHz	2x16 K L1, 256K L2, MMX	32	~900	97W	~7M	$900
IBM PowerPC 750X	550 MHz	2x32 K L1, 256K L2	32/64	~1300	5W	~7M	$900
MIPS R5000	250 MHz	2x32 K 2 way set assoc.	32/64	NA	NA	3.6M	NA
StrongARM SA-110	233 MHz	None	32	268	1W	2.1M	NA
Microcontroller							
Intel 8051	12 MHz	4K ROM, 128 RAM, 32 I/O, Timer, UART	8	~1	~0.2W	~10K	$7
Motorola 68HC811	3 MHz	4K ROM, 192 RAM, 32 I/O, Timer, WDT, SPI	8	~.5	~0.1W	~10K	$5
Digital Signal Processors							
TI C5416	160 MHz	128K, SRAM, 3 T1 Ports, DMA, 13 ADC, 9 DAC	16/32	~600	NA	NA	$34
Lucent DSP32C	80 MHz	16K Inst., 2K Data, Serial Ports, DMA	32	40	NA	NA	$75

Sources: Intel, Motorola, MIPS, ARM, TI, and IBM Website/Datasheet; Embedded Systems Programming, Nov. 1998

The Advantages of Microcontrollers

- Low cost due to high volume production
- Low "chip count" due to integrated on-board features
- Good development tools and environments
- Extensive product families allow tailoring of processor to system design metrics
- Short product design cycles (compared to custom hardware design).
- Compatible with hardware/software co-design
  - Many microcontrollers are available as "VHDL Cores" for integration into a custom VLSI chip

Most Embedded Applications Require only Modest Computational Power

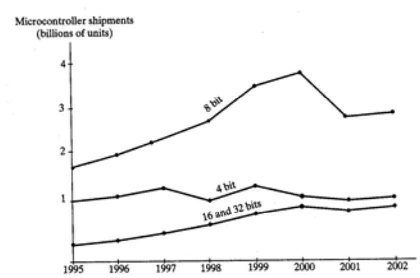
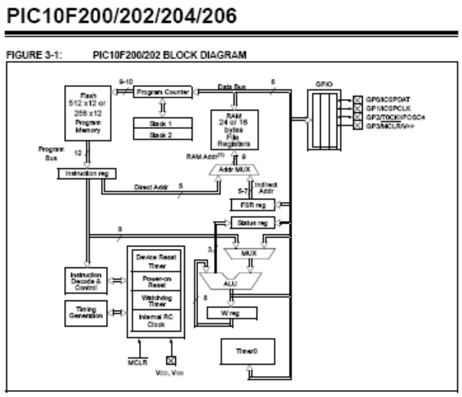
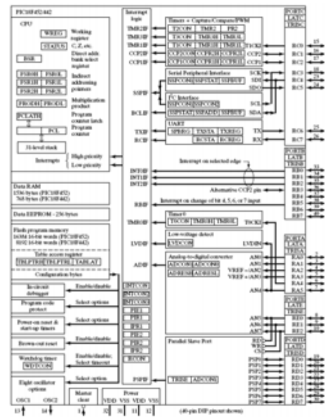


Figure 1-3 Microcontroller unit shipments per year, as distinguished by data word length (Dataquest).

A Low End Microcontroller



The PIC18F452 Microcontroller



PIC18F452 Features

- Clock rate up to 40 MHZ
- 32KB program memory (FLASH)
- 1536B data memory (RAM)
- 256B data EEPROM
- 35 digital I/O bits
- 4 timers
- 2 PWM/Capture/Compare modules
- UART for serial I/O
- 10-bit A/D converter (8 channels)
- 18 interrupt sources
- Cost: Approx $5.00 in quantity