Welcome to EE/CS120B!

Lecturer: Brian Grattan

e-mail: bgrattan@cs.ucr.edu

office: Surge 340

Welcome to EE/CS120B!

Check out policies and sign up for e-mail list at:

http://www.cs.ucr.edu/cs120b/

click on “Fall 2004”

Welcome to EE/CS120B!

What to expect in the class:

– Typically, each week will have a quiz or a homework

Welcome to EE/CS120B!

What to expect in the class:

– Typically, each week will have a quiz or a homework

– Labs will start with the assumption that you have a basic understanding of VHDL and C.

Welcome to EE/CS120B!

What to expect in the class:

- Typically, each week will have a quiz or a homework
- Labs will start with the assumption that you have a basic understanding of VHDL and C.
- Grading follows the 90-80-70-60 scale. There is no curve. So, you are encouraged to work together on homework problems, but not copy!

Welcome to EE/CS120B!

What to expect in the class:

- The course will follow the book pretty closely, but the book is getting a little dated, so some extra things may be covered outside the book.

Welcome to EE/CS120B!

What will you learn in CS120B?

- The trade-offs in designing an embedded system
- How to design a single purpose processor in hardware and in software
- What some of the trends in embedded design are
- How to optimize your design and make intelligent design choices up front
- and more…

Embedded Systems

What is an Embedded System?

Embedded Systems

What makes something an Embedded System?

- Single Function
- Tightly Constrained
- Reactive and Real Time

Design Metrics

When we say "tightly constrained" there are many different aspects that we are measuring:

- NRE (Non-Recurring Engineering) Cost
- Unit cost
- Size
- Speed
- Power (usually power consumed)
- Flexibility
- Time to prototype
- Time to market
- The list goes on….

Design Metrics

A couple of especially important metrics:

- Time to market
- Total cost (NRE + unit cost)
- Performance (speed—latency and throughput)

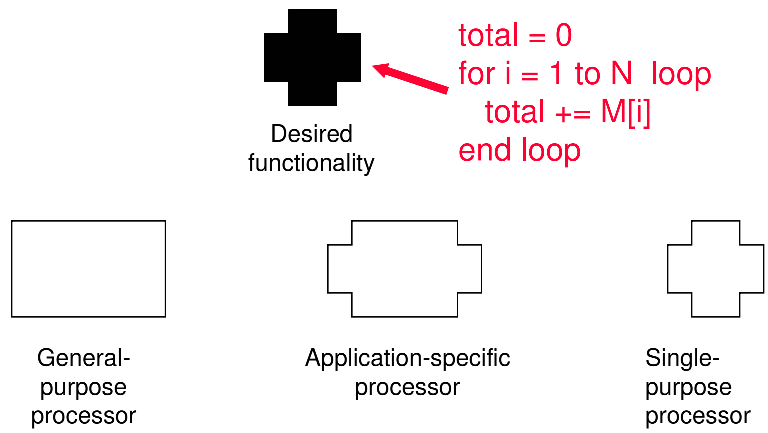
Processor technology

What is a "processor?"

- The architecture of the computation engine used to implement a system's desired functionality
- Processor does not have to be programmable
  - "Processor" *not* equal to general-purpose processor like it was in CS 61
  - It could be an Intel Itanium, the chip inside an optical mouse that determines where it is moving, or the chip inside a simple digital watch
- Basically, anything that computes values based on some input and/or memory.

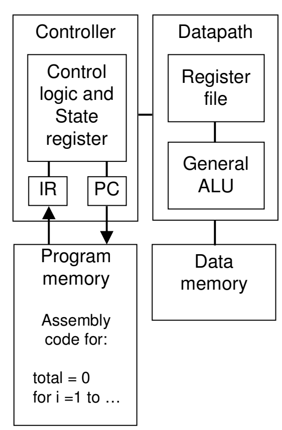
Processor technology

- Processors vary in their customization for the problem at hand



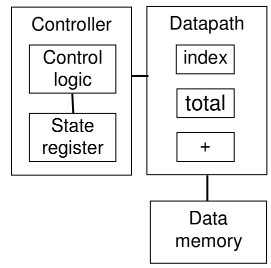
General-purpose processors

- 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
- "Pentium" the most well-known, but there are hundreds of others



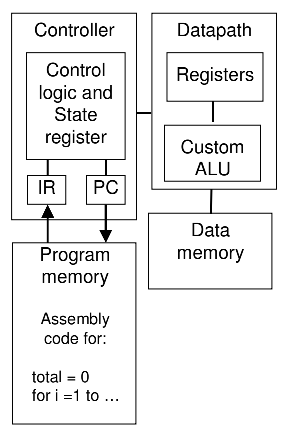
Single-purpose processors

- 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

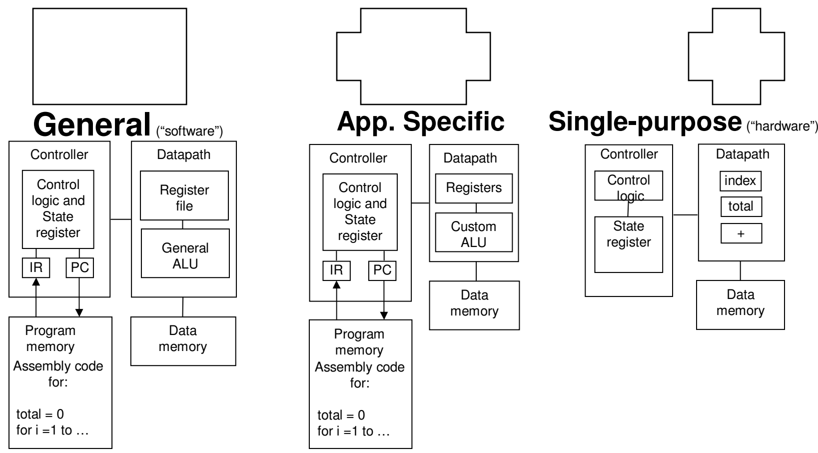


Application-specific processors

- 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



Overview of Processors



IC technology

- The manner in which a digital (gate-level) design gets implemented
- Basically, you have a design, which is an architecture, or model of how things are going to be computed and you want to implement it.

IC technology

- Three types Design Methods for building your IC
  - –Full-custom
  - –Semi-custom (gate array and standard cell)
  - –PLD (Programmable Logic Device)

Full-custom

- All layers are optimized for an embedded system's particular digital implementation
  - – Placing transistors
  - – Sizing transistors
  - – Routing wires
- Benefits
  - – Excellent performance, smallest size, low power
- Drawbacks
  - – High Non-Recurring Engineering (NRE) cost (e.g., $300k), long time-to-market, high risk

Gate Array

- Lower layers are fully or partially built
  - Designers are left with routing of wires and maybe placing some blocks
- Benefits
  - Good performance, good size, less NRE cost than a full-custom implementation (perhaps $10k to $100k)
- Drawbacks
  - Still require weeks to months to develop

Standard Cell

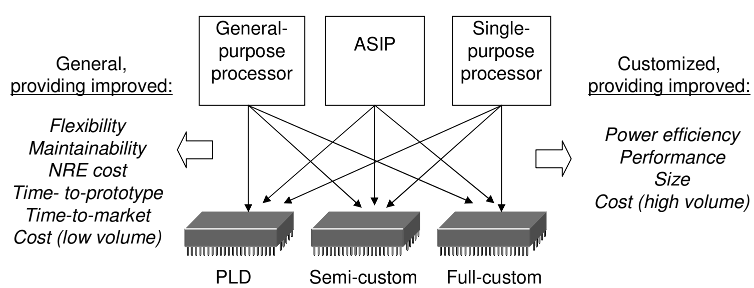
- Instead of designing the actual silicon, the designer uses pre-built "blocks" that have already been tested and proven.
- Benefits
  - Good performance, good size, less NRE cost than a full-custom implementation (perhaps $10k to $100k)
- Drawbacks
  - Still require weeks to months to develop

PLD (Programmable Logic Device)

- All layers already exist
  - Designers can purchase an IC
  - Connections on the IC are either created or destroyed to implement desired functionality
  - Field-Programmable Gate Array (FPGA) very popular
- Benefits
  - Low NRE costs, almost instant IC availability
- Drawbacks
  - Bigger, expensive (perhaps $30 per unit), power hungry, slower

Independence of processor and IC technologies

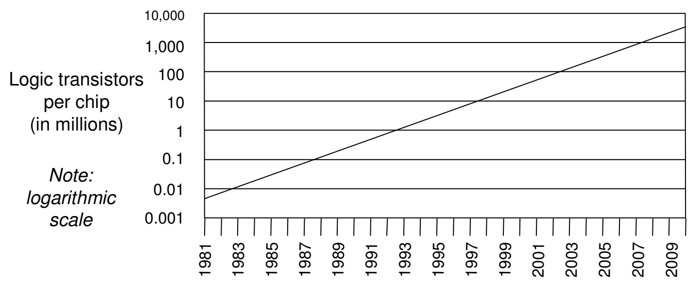
- Basic tradeoff
  - General vs. custom
  - With respect to processor technology or IC technology
  - The two technologies are independent



Moore's law

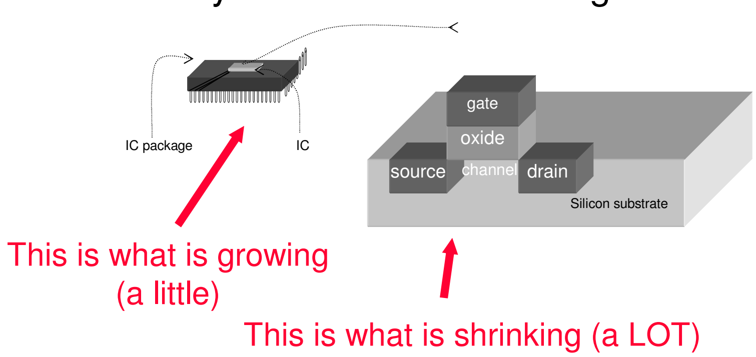
- The most important trend in embedded systems
  - Predicted in 1965 by Intel co-founder Gordon Moore

<u>IC transistor capacity has doubled roughly every 18 months for the past several decades</u>



IC technology

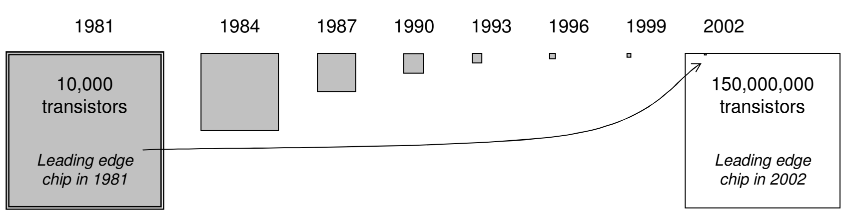
- This is a very basic illustration of a gate



Moore's law

- Wow
  - This growth rate is hard to imagine, most people underestimate
  - How many ancestors do you have from 20 generations ago
    - i.e., roughly how many people alive in the 1500's did it take to make you?
    - 2^{20} = more than *1 million people*
  - *(This underestimation is the key to pyramid schemes!)*

Graphical illustration of Moore's law



- Something that doubles frequently grows more quickly than most people realize!
  - A 2002 chip can hold about 15,000 1981 chips inside itself
  - Intel Itanium: 325 Million Transistors!

To do before next Tuesday…

- Read Chapter 1 in Embedded System Design (ESD)
- Go to lab (Surge 173)