Course Outline

Hardware Components
- Concept
- Specification

Software Components

HW/SW Partitioning
Estimation - Exploration

Hardware
- Design (Synthesis, Layout, ...)
- Design (Compilation, ...)

Software

Validation and Evaluation (area, power, performance, ...)

Hardware Components: Information Processing

Nikil Dutt

UC Irvine
ICS 212 Winter 2005

Copyrighted Material adapted from Peter Marwedel, Frank Vahid and Tony Givargis
Templates from Prabhat Mishra
Components of Embedded Systems

**Memory**
- Flash
- DRAM
- Memory Card

**Controllers**
- FLASH Adapter / Memory Control
- I/O Control
- Ethernet Interface

**Processor**
- CPU
- JFEG Co-Processor
- Graphics Controller
- LCD Interface
- Video Encoder

**Interface**
- I2C
- LED
- Bluetooth Module
- Ethernet PHY

**Coprocessors**
- Camera DSP
- ADC

**Converters**
- Lens
- A/D
- Camera

**Software**
- Application Programs

ES: Simplified Block Diagram

A/D converter

sample-and-hold

sensors

environment

information processing

display

D/A converter

actuators
Information Processing

- **ASIC**
- **Processor**
  - Energy efficient
  - Code-size efficient
  - Run-time efficient
- **Reconfigurable hardware**
- **Memory**

Processors

- **What is a processor?**
  - Artifact that computes (runs algorithms)
  - Controller and data-path
- **General-purpose processors (GP):**
  - Variety of computation tasks
  - Functional flexibility and low cost at high volumes (maybe)
  - Slow and power hungry
- **Application-Specific Instruction-set Processors (ASIP):**
  - Tuned for application domain, but programmable
  - Fast and power efficient (compared to GP)
- **Application-Specific Integrated Circuit (ASIC):**
  - Customized hardware for specific task/application
  - Fast, power efficient, minimal area
  - Functional inflexibility and high cost at low volumes (maybe)
**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

- **Examples**
  - Pentium, Athlon, PowerPC

---

**Application-specific IS processors (ASIPs)**

- **Programmable processor optimized for a particular class of applications having common characteristics**
  - Compromise between general-purpose and ASIC (custom hardware)

- **Features**
  - Program memory
  - Optimized datapath
  - Special functional units

- **Benefits**
  - Some flexibility, good performance, size and power

- **Examples**
  - DSPs, Video Signal Processors, Network Processors,...
Application-Specific ICs (ASICs)

- Digital circuit designed to execute exactly one program
  - coprocessor, hardware accelerator

- Features
  - Contains only the components needed to execute a single program
  - No program memory

- Benefits
  - Fast
  - Low power
  - Small size

Application Specific Circuits (ASIC)

- Custom-designed circuits necessary if ultimate speed or energy efficiency is the goal and large numbers can be sold.
- Approach suffers from long design times and high costs.
A digital implementation (gate-level) is mapped to silicon using various layers:
- Full-custom/VLSI
- Semi-custom ASIC (gate array and standard cell)
- PLD (Programmable Logic Device)

**Full-custom/VLSI**

- All layers are optimized for an embedded system’s particular implementation:
  - Placing transistors
  - Sizing transistors
  - Routing wires

**Benefits**
- Excellent performance, small size, low power

**Drawbacks**
- High NRE cost (e.g., $300k), long time-to-market
Semi-custom

- **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

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**
  - Penalty on area, cost (perhaps $30 per unit), performance, and power
Hardware Design Technology

- The manner in which we convert our concept of desired system functionality into a HW implementation

Compilation/Synthesis: Automates exploration and insertion of implementation details for lower level.

Libraries/IP: Incorporates pre-designed implementation from lower abstraction level into higher level.

Test/Verification: Ensures correct functionality at each level, thus reducing costly iterations between levels.

<table>
<thead>
<tr>
<th>Compilation/ Synthesis</th>
<th>Libraries/ IP</th>
<th>Test/ Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>System specification</td>
<td>Hw/Sw/ OS</td>
<td>Model simulat./ checkers</td>
</tr>
<tr>
<td>Behavioral specification</td>
<td>Behavior synthesis</td>
<td>Cores Hw-Sw cosimulators</td>
</tr>
<tr>
<td>RT specification</td>
<td>RT synthesis</td>
<td>RT HDL simulators</td>
</tr>
<tr>
<td>Logic specification</td>
<td>Logic synthesis</td>
<td>Gates/ Cells Gate simulators</td>
</tr>
</tbody>
</table>

Design productivity gap

- 1981 leading edge chip required 100 man-months
  - 10,000 transistors / 100 transistors/month
- 2002 leading edge chip requires 30K man-months
  - 150,000,000 / 5000 transistors/month
- Designer cost increase from $1M to $300M
The mythical man-month

- In theory, adding designers to team reduces project completion time.
- In reality, productivity per designer decreases due to complexities of team management and communication overhead.
- In the software community, known as “the mythical man-month” (Brooks 1975).
- At some point, can actually lengthen project completion time!

- 1M transistors, one designer=5000 trans/month.
- Each additional designer reduces for 100 trans/month.
- So 2 designers produce 4900 trans/month each.

GP, ASIP, ASIC: Present a Range

GP…………….. ASIP…………… ASIC

- Programmable controller
  - Control logic is stored in memory
  - Fetch/decode overhead
- Highly general data-path
  - Typical bit-width (8, 16, 32, 64)
  - Complete set of arithmetic/logic units
  - Large set of registers
- High NRE/sale-volume

- Hardwired controller
  - No need for program memory and cache
  - No fetch/decode overhead
- Highly tuned data-path
  - Custom bit-width
  - Custom arithmetic/logic units
  - Custom set of registers
- Low NRE/sale-volume
Need for efficiency (power + energy):

“Power is considered as the most important constraint in embedded systems”

Current UMTS (3-G) phones can hardly be operated for more than an hour, if data is being transmitted.
[from a report of the Financial Times, Germany, on an analysis by Credit Suisse First Boston; http://www.ftd.de/tm/tk/9580232.html?nv=se]
The energy/flexibility Conflict

- Technology: Operations/Watt [MOps/mW]
- Ambient Intelligence
- DSP-ASIPs
- μPs
- poor design generation techniques

Necessary to optimize!

[H. de Man, Keynote, DATE’02; T. Claasen, ISSCC99]
In many cases, faster execution also means less energy, but the opposite may be true if power has to be increased to allow faster execution.

Low Power vs. Low Energy

- Minimize the power consumption
  - design of the power supply
  - design of voltage regulators
  - dimensioning of interconnect
  - short term cooling

- Minimizing the energy consumption
  - restricted availability of energy (mobile systems)
    - limited battery capacities (only slowly improving)
    - very high costs of energy (solar panels, in space)
  - cooling
    - high costs
    - limited space
  - dependability
    - long lifetimes, low temperatures
Storage

- **What is a memory?**
  - Artifact that stores bits
  - Storage fabric and access logic

- **Write-ability**
  - Manner and speed a memory can be written

- **Storage-permanence**
  - Ability of memory to hold stored bits after they are written

- **Many different types of memories**
  - Flash, SRAM, DRAM, etc.

- **Common to compose memories**

---

Write-ability

- **Ranges of write ability**
  - High end
    - Processor writes to memory simply and quickly
    - E.g., RAM
  - Middle range
    - Processor writes to memory, but slower
    - E.g., FLASH, EEPROM
  - Lower range
    - Special equipment, “programmer”, must be used to write to memory
    - E.g., EPROM, OTP ROM
  - Low end
    - Bits stored only during fabrication
    - E.g., Mask-programmed ROM
Storage-permanence

- **Range of storage permanence**
  - **High end**
    - Essentially never loses bits
    - E.g., mask-programmed ROM
  - **Middle range**
    - Holds bits days/months/years after memory’s power source turned off
    - E.g., NVRAM
  - **Lower range**
    - Holds bits as long as power supplied to memory
    - E.g., SRAM
  - **Low end**
    - Begins to lose bits almost immediately after written
    - E.g., DRAM

---

Memory Types

- Mask-programmed ROM
- OTP ROM
- EPROM
- EEPROM
- Flash
- NVRAM
- SRAM/DRAM

Nonvolatile | In-system programmable | Ideal
---|---|---

Write-ability

---
Communication

- What is a bus?
  - An artifact that transfers bits
  - Wires, air, or fiber and interface logic

- Associated with a bus, we have:
  - Connectivity scheme
    - Serial Communication
    - Parallel Communication
    - Wireless Communication
  - Protocol
    - Ports
    - Timing Diagrams
    - Read and write cycles
  - Arbitration scheme, error detection/correction, DMA, etc.

Serial Communication

- A single wire used for data transfer
- One or more additional wires used for control (but, some protocols may not use additional control wires)
- Higher throughput for long distance communication
  - Often across processing node
- Lower cost in terms of wires (cable)
- E.g., USB, Ethernet, RS232, I²C, etc.
Parallel Communication

- Multiple buses used for data transfer
- One or more additional wires used for control
- Higher throughput for short distance communication
  - Data misalignment problem
  - Often used within a processing node
- Higher cost in terms of wires (cable)
- E.g., ISA, AMBA, PCI, etc.

Wireless Communication

- Infrared (IR)
  - Electronic wave frequencies just below visible light spectrum
  - Diode emits infrared light to generate signal
  - Infrared transistor detects signal, conducts when exposed to infrared light
  - Cheap to build
  - Need line of sight, limited range
- Radio frequency (RF)
  - Electromagnetic wave frequencies in radio spectrum
  - Analog circuitry and antenna needed on both sides of transmission
  - Line of sight not needed, transmitter power determines range
Peripherals

- Perform specific computation task
- Custom single-purpose processors
  - Designed by us for a unique task
- Standard single-purpose processors
  - “Off-the-shelf”
  - pre-designed for a common task

Timers

- Timers: measure time intervals
  - To generate timed output events
  - To measure input events
  - Top: max count reached
- Range and resolution
Counters

- **Counter**: like a timer, but counts pulses on a general input signal rather than clock
  - e.g., count cars passing over a sensor
  - Can often configure device as either a timer or counter

![Diagram of Counter Circuit]

Watchdog Timer

- **Must reset timer every X time unit, else timer generates a signal**
- **Common use**: detect failure, self-reset
**UART**

- **UART: Universal Asynchronous Receiver Transmitter**
  - Takes parallel data and transmits serially
  - Receives serial data and converts to parallel
- **Parity: extra bit for simple error checking**
- **Start bit, stop bit**
- **Baud rate**
  - Signal (or phase) changes per second possible
  - Bit rate, sometimes different (e.g., due to encoding)
    - E.g., older modems: 14400 bps @ 2400 baud
      - Encode 6 bits per signal transition

**Pulse Width Modulator (PWM)**

- **Generates pulses with specific high/low times**
- **Duty cycle: % time high**
  - Square wave: 50% duty cycle
- **Common use: control average voltage to electric device**
  - Simpler than DC-DC converter or digital-analog converter
  - DC motor speed, dimmer lights
**LCD**

- Liquid Crystal Display
- N rows by M columns
- Controller build into the LCD module
- Simple microprocessor interface using ports
- Software controlled

**Keypad**

- N=4, M=4

```plaintext
N1
N2
N3
N4
M1
M2
M3
M4
```
Summary

- Hardware: Information Processing
  - Processors
  - Memories
  - Communication
  - Peripherals

Evaluation and Grading

- 40\%  Homeworks
  - 4 HWs, 10\% each

- 30\%  ES case study  OR  ES Project
  - Details in next class
  - Talk to me if you have a specific project in mind

- 30\%  Comprehensive Final Exam