EE 3170 Microcontroller Applications

Lecture 13: Introduction to Microcontroller Hardware (Part I- Bus & Memory)
- Miller §6.1 - §6.12

Based on slides for ECE3170 by Profs. Davis, Kieckhafer, Tan, and Cischke

Lecture Overview

- Hardware Software Tradeoffs & Co-design
- Busses
  - bus organization
  - interfacing to & controlling memory chip(s)
- Memory
  - definitions and organization
  - implementation technologies

Hardware/Software Trade-offs

- General Purpose Computing:
  - Real-Time performance is usually not an issue
    - Little need for intimate knowledge of H-Ware
  - System provides layers of abstraction for programmer:
    - HLL compilers
    - OS GUI services
    - OS kernel to manage hardware utilization
    - Hardware device drivers
    - The actual hardware
  - System massively increases the user’s productivity
    - Allows programmer to focus on Application Semantics

- Embedded Computing:
  - Generally controlling a Real-Time system
    - sequencing and precise timing issues are important
    - price/performance constraints may be severe
  - System objectives may be very different
    - size
    - cost
    - performance
    - memory usage
    - power consumption
    - reliability issues
Design Issues

- Embedded System Designer needs to trade/off:
  - hardware complexity vs. software complexity
  - hardware speed vs. software speed
  - hardware size vs. program memory size
  - etc.

- Hardware-Software Co-design
  - an entire field of Computer Engineering that has grown out of the needs of embedded systems

Hardware Building Blocks

- Input/Output
- Processor
- Control Unit
- Memory
- Clock

Hardware Terminology

- Bus
  - Collection of binary signal wires
- 68HC11 Buses
  - 16-bit Address Bus
  - 8-bit Data Bus
  - Control Bus

- (Non) Volatile Memory
  - No power - no data
  - (No power - still data)

- Read/Write memory
  - Store/retrieve at full speed
- Read Only Memory
  - Store at much slower speed

- Random Access Memory
  - Equal time for all locations

- Sequential Access
  - Read/Write in order

Bus Architectures

- A modern Processor system has multiple busses
- Busses are arranged hierarchically by Bandwidth
  - Higher bandwidth busses closer to processor
- Any communication between system components requires a bus between the two components
  - Bus should be of sufficient bandwidth to handle communication by all components concurrently
  - Bandwidth: Amount of data that can be transmitted in a fixed amount of time
- A Typical Bus has three bus components
  - Address Bus, Data Bus, Control Bus
- Two types of bus synchronization architectures
  - Synchronous
    - one of the control lines is a common ‘clock’ line
    - all devices on the bus can synch, their actions to the clock
    - 68HC11 clock signal is named “E clock”
  - Asynchronous
    - There is no common clock line
    - devices use interlocked “handshaking” signals to synch.
### Microprocessor Bus Connections

- Microprocessor
- RAM
- EEPROM
- I/O
- ROM
- ABUS
- DBUS

### Bus Operation Example

**Example Instruction (STAA)**

- **Register and Memory contents**
  - **C100**: B7
  - **C101**: C2
  - **C102**: 00
  - **C200**: 44

**Memory contents to IR**
- **C100**: B7 (B7)
- **C101**: C2
- **C102**: 00
- **C200**: 44

### Instruction Fetch

**PC to Memory as Address**
- Memory contents to IR

**Memory contents to IR**
- **C100**: B7
- **C101**: C2
- **C102**: 00
- **C200**: 44
Instruction Fetch

- PC to Memory as Address
  - Memory contents to IR

<table>
<thead>
<tr>
<th>C100</th>
<th>B7</th>
<th>C1</th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td>C101</td>
<td>C2</td>
<td>IR</td>
<td></td>
</tr>
<tr>
<td>C102</td>
<td>00</td>
<td>B7</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>C200</td>
<td>44</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instruction Execution

- Operand Address to Memory
  - Accumulator A contents to Memory

<table>
<thead>
<tr>
<th>C100</th>
<th>B7</th>
<th>C1</th>
<th>03</th>
</tr>
</thead>
<tbody>
<tr>
<td>C101</td>
<td>C2</td>
<td>IR</td>
<td></td>
</tr>
<tr>
<td>C102</td>
<td>00</td>
<td>B6</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>C200</td>
<td>22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory

- Memory Signals
  - Address Bus
    - A15 - A0
  - Data Bus
    - D7 - D0
  - Read/Write
    - R/W
  - Timing Pulse
    - CS
    - Tri-state outputs

Memory Chip Interfacing

- Data Bus
  - 8-bit wide chips connect directly to the 8 data lines
- Address Bus
  - Processor bus has 16 address lines 64K bytes
  - Memory chips may have less than 64K bytes
  - Smaller chip will use only some of the address lines
  - Number of addr. lines used = log2(number of bytes)
    - e.g. a 4K byte chip can only use 12 addr lines (A11 - A0)
    - Lines A15 - A12 are not used
- Control Bus
  - Many possible signals
  - R/W Line is connected to R/W pin of chip
    - R/W = 1 → Read command
    - R/W = 0 → Write command
  - E-Clock line is hooked to CE (chip enable) pin of chip
    - synchronizes chip enablement with the system
    - processor ensures all other chip inputs are stable at the time E-clock enables the chip
Memory Chip Interfacing

- Control Bus
  - Clock
  - Interrupt
  - Output Enable
  - Read/Write
  - Chip Select / Enable
  - Bus Master
  - Test Signals (JTAG)
  - Byte Enable(s)
  - REQ / ACK
  - Parity / PERR / SERR

Introduction to 68HC11 Hardware

- There are lots of versions of the 68HC11 available.
  - differ mainly in the amount and kind of on-chip memory,
  - there are also variations on the available I/O pins and on-chip hardware functions.

- Most representative members of 68HC11 family
  - 68HC11A8 & 68HC11E9 versions
  - 8K ROM + 256 bytes RAM; 12K ROM + 512 bytes RAM, respectively.
  - Each has 512 bytes EEPROM.

Operating Modes and Hardware Overview

- All 68HC11's have a total of 4 operating modes
  - determined at power-on or hardware-reset time by hardware settings.
  - Two are intended for testing and not normally used
  - The other two are
    - Single-chip mode: used for applications in which only internal (i.e., on-chip) memory and I/O devices are used.
      - This is the mode that we'll look at most closely, and is the one emulated by the 6811 EVB.
    - Expanded-multiplexed mode, or simply expanded mode
      - used when the 68HC11 is interfaced to external memory and/or I/O devices.
      - This is the mode that the 68HC11 on the EVB actually runs in.
  - Mode control is through the MODA and MODB pins:
    - 
    | MODA | MODE | Mode |
    |------|------|------|
    | 0    | 0    | Bootstrap |
    | 0    | 1    | Special test |
    | 1    | 0    | Single chip |
    | 1    | 1    | Expanded |

68HC11 Block Diagram
68HC11 Hardware Overview
- **Clock:** A crystal of frequency up to 8 should be connected across pins XTAL and EXTAL. Pin E is an output clock signal called the E clock. Its frequency is one-fourth that applied at XTAL/EXTAL.
- **Interrupt and reset** functionality is provided by IRQ/XIRQ and RESET pin, respectively.
- **Port A:**
  - provides **timing and event-counting** functions and/or general purpose digital I/O.
- **Port B:**
  - In single-chip mode: provides 8 general-purpose output pins.
  - In expanded mode: carries the most significant 8 bits of the address bus.
- **Port C:**
  - In single-chip mode: provides 8 programmable input/output pins: each can be configured, under software control, as either an input or an output.
  - In expanded mode: carries, through time-division multiplexing, both the least-significant 8 bits of the address bus and all 8 bits of the data bus.

68HC11 Hardware Overview
- **Port D:**
  - provides two kinds of **serial interface**: SCI (serial communications interface) and SPI (serial peripheral interface). These are used to connect to terminals, host computers, or other external devices with serial interfaces.
- **Port E:**
  - provides 8 analog input channels, all multiplexed to a single 8-bit A/D converter whose range can be set by applying appropriate voltages to the VRH and VRL pins, respectively.
- **STRA and STRB:**
  - in single-chip mode, are a **strobe** input & output, respectively, for **handshaking** with external peripheral devices.
  - In expanded mode, they provide address & data bus status signals.

Address DeMultiplexing

![Figure 4-1. Address/Data Demultiplexing](image)

Bus Architectures & Characteristics

<table>
<thead>
<tr>
<th>Type of Bus</th>
<th>Bus Clock Signal</th>
<th>Bus Width</th>
<th>Throughput Max. Transfer Rate</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry Standard Architecture ISA</td>
<td>6 MHz</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>Low cost, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>MicroChannel Architecture</td>
<td>6 MHz</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>Low cost, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>Peripheral Component Interconnect</td>
<td>64-bit</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>CompactPCI</td>
<td>100 MHz</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>Advanced Peripheral Interconnect</td>
<td>100 MHz</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>Universal Serial Bus</td>
<td>100 Mbps</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>USB 1.1</td>
<td>50 Mbps</td>
<td>64-bit</td>
<td>166 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
<tr>
<td>IEEE 1394</td>
<td>no</td>
<td>no</td>
<td>50 Mb/sec</td>
<td>High speed, space efficient, compatible with older systems</td>
<td>Limited uses</td>
</tr>
</tbody>
</table>

...
### I/O Devices

- **Memory**
  - Non-Volatile: some is essential
  - Mass Storage: SRAM, DRAM, Working Set Memory

- **Devices**
  - Video output
  - Keyboard / Mouse Input
  - UART: Serial Communication
  - USB: Serial Communication
  - NIC: Network Communication
  - Hard Disk(s)

- **Bus Bridges** (AGP/PCI/ISA/VESA/etc..)

### Memory Permanence

- **Non-Volatile** = Data is retained across shutdowns
  - e.g. Magnetic Disk, CD

- **Volatile** = Data disappears when powered down
  - Static = With power up, data stays stable forever
    - Static RAM (SRAM) generally used in cache
    - it is also used in simple embedded systems
  - Dynamic = Even with power up, data slowly decays
    - Dynamic RAM (DRAM) generally used in main memory
    - periodic "refresh" cycles restore data to full strength

### Memory Organization

- **SAM** = Sequential Access Memory
  - must execute read & writes in a specified order
    - e.g. magnetic tape
  - access time depends on the address of the data

- **RAM** = Random Access Memory
  - access time is independent of address
  - all addresses take the same time
    - e.g. semiconductor memory chip

### SRAM Block Diagram (skipped)
Memory Alterability

- ROM = Read Only Memory
  - processor can only read data
  - processor cannot write to the memory at all, OR
  - it cannot write “at processor-like speeds”
    - e.g. devices taking millisecond to write are called ROMs
  - ROMs are by definition non-volatile

- RWM = Read/Write Memory
  - processor can read and write to the memory
  - at “processor-like speeds”, e.g. 10’s of nanoseconds.
  - Usually, Semiconductor RAM is the only RWM in system

Memory Interface

- Single Memory Chip Interface
  - Given $2^N$ words @ w bits per word
    
    ![Memory Interface Diagram]

Memory Address Decoding

- All Memories have some sort of matrix architecture
- The Address is decoded from an $N$-bit number
  - into $2^N$ mutually exclusive “select lines”
  - exactly one line is active (logic 1) at any given time

Memory Array Structure

- Highly regular structure
- At the junction of each word line and bit line (pair) is a cell containing a BIT
- Row decoder activates one word line
- Data propagates down bit line from cell
- Sense Amps are necessary in large arrays
- Column Decoder selects a subset of bits available on bit lines
- Roughly square aspect ratio is desirable
RAM Data Array

- Each RAM word is a "register"
  - built out of 1-bit storage cells
  - cells have tri-state bidirectional input/output lines
  - all cells in 1 word are activated by the same select line

RAM Technology

- Does not use a cross-connect array technology
- Each RAM word is a "register"
  - built out of 1-bit storage cells
  - cells have tri-state bidirectional input/output lines
  - all cells in 1 word are activated by the same select line

Static RAM Cell Technology

- Each bit cell is essentially a D latch with tri-state I/O
- Very fast read and write times
- Not very dense
  - Needs approx 6 - 10 transistors per cell
  - Needs power and ground lines distributed to all cells
- Primary uses
  - wherever speed is of primary importance (Cache)
  - wherever the complexity of Dynamic RAM refreshing is not practical

SRAM Technology

- Requires two lines for bit transmission (b & ~b)
- High speed
- Moderate power consumption
  - CMOS – Complimentary, lower power than nMOS/pMOS
  - Static leakage & Disappartion
- Lower density than other technologies
Dynamic RAM Cell Technology

- Each bit cell stores a charge on a Capacitor
  - “Write” places a charge on the gate
  - “Read” senses the charge on the gate
- DRAM is the CHEAPEST memory in per-bit cost
  - DRAM cells are much small and fast
  - Insufficient insulator to hold a charge for long ⇒ periodic refresh is essential
- Slower than SRAM
- But, needs only 1 transistors per cell
  - used where density is of primary import (main memory)

DRAM Technology

1-Transistor Memory Cell (DRAM)

- Write:
  1. Drive bit line
  2. Select row
- Read:
  1. Freecharge bit line to VDDL
  2. Select row
  3. Cell and bit line share charges
    - Very small voltage changes on the bit line
    - Sense (fancy sense amp)
    - Can detect changes of ~1 million electrons
  4. Write: restore the value
  5. Refresh:
    - Just do a dummy read to every cell.

DRAM Technologies

- Asynchronous DRAM
  - Conventional -> Fast Page Mode -> Extended Data Out
    - Burst EDO
- Synchronous DRAM
  - SDRAM -> SDR PC66-133 -> DDR PC1800-2400 ->
    - DDR2 200Mhz-??Mhz
- Revolutionary DRAM
  - Packetized : Rambus, Direct Rambus & SyncLink
  - Cached : Virtual Channel & Enhanced

Classes of ROM

- ROM = data is loaded at the factory
  - only good for high-volume final products
- PROM = Programmable ROM
  - can be programmed by a “prom burner” in the lab
  - good for low-volume products
- EPROM = Erasable PROM
  - can be erased by intense UV light
  - good for design and prototyping
- EEPROM = Electrically EPROM
  - can be erased by a higher-than-normal voltage
  - too slow to qualify as RWM
ROM Matrix Technology

- A Diode Matrix is the primary ROM technology
  - All semicond. ROMs are some variation on the theme

- Start with two orthogonal sets of wires
  - horizontal wires are the select lines
  - vertical wires are the data bit-lines
  - The wires cross over but do not touch

- At desired points, install diodes
  - from the select line to the data line
  - each diode generates a possible “1” on its data line

PROM Matrix Technology

- Install a diode at every select line / data line junction
  - each cross-point is initially a “one”

- Install a “fusible link” in series with the diode
  - To get a “zero”, simply burn out a link

- PROM Burner
  - applies higher than normal voltage
  - burns out the selected links

- PROM is not reprogrammable
  - can go back and add a few more zeroes,
  - can not change a zero back to a one
EPROM Matrix Technology

- Replace Diodes with Field Effect Transistors (FETs)
- FET gate is electrically isolated from rest of device
- Placing a charge on the gate “turns on” the transistor
  - Charged gate $\Rightarrow$ device conducts
  - Discharged gate $\Rightarrow$ device does not conduct
- EPROM Programmer charges gates to store a “1”
  - Charging is done in repeated pulses to control heating
  - Insulator can hold the charge for years
- Erasure is done with UV light
  - Causes all gates to discharge

EEPROM Matrix Technology

- Same general concept as EPROM
  - Charge on an FET gate makes the connection
- However, gates can be electrically discharged
- To prevent accidental erasure, it usually requires:
  - A special voltage AND/OR
  - A complex command sequence
- Applications
  - Upgradable bootstrapping ROM (FLASH memory)
  - “Non-Volatile Ram” (NVRAM) although it is slow

Controlling Multiple Memory Chips

- Suppose we want a total of 8K of memory
- Can use two 4KB chips
  - chip 0 = addresses (0 ... 4K -1)
  - chip 1 = addresses (4K ... 8K -1)
- 8K memory space 13 address bits
  - Each chip only uses 12 address lines
- What do we do with the 13th address bit (A12)?
  - Use it to select the desired chip for enable
    - A12 = 0 $\Rightarrow$ data is on chip 0
    - A12 = 1 $\Rightarrow$ data is on chip 1
Interface to Two 4KB Mem Chips

Chip Enable Logic
- Implements conditions that enable that one chip
- Logic Block for Chip 0
- Logic Block for Chip 1

Interface to One 4KB Mem Chip