Announcements

◆ Many posted materials are accessible only from a CMU IP Address
  • Look for this on course web page:
    If you can't access a file due to access restrictions, you need to get a campus IP address for your web browsing requests. Use Cisco VPN Anyconnect...

◆ Course web page has schedules, assignments, other important info
  • http://www.ece.cmu.ecu/~ece348
  • Blackboard will have grades, announcements, sample tests
  • Look at blackboard announcements before sending e-mail to course staff

◆ Lab board handouts in progress
  • See Blackboard/admin page for TA office hours
  • OK to go to any scheduled lab section (but, give priority to scheduled students)
  • For Friday prelab give a good faith attempt to get things working by the deadline
    – If you hit a showstopper get it fixed on Tuesday so you can do Prelab 2 on time.
Design Example: Rack-Mount Power Supply

- **Power supply for server**
  - AC to DC conversion (750-1000W)
  - Coordinates 2 redundant supplies to maximize uptime
  - Safeguard against power problems (under/over-voltage; over-current; over-temp)

- **Typical approach:**
  - General microcontroller for AC, alarms, housekeeping
  - DSP runs control loop on DC side at > 10 KHz to provide stable DC power

- **Key requirement:**
  “Doesn’t emit smoke”
Where Are We Now?

◆ Where we’ve been:
  • Course Intro

◆ Where we’re going today:
  • Embedded system hardware

◆ Where we’re going next:
  • Microcontroller assembly language
  • Engineering design approaches
  • Embedded-specific C
  • …

Preview

◆ Microcontroller Hardware
  • How does a microcontroller connect to the rest of the system?
  • I/O bus
  • Support circuitry
  • Power supplies

◆ Hardware implementation
  • Prototyping techniques
  • Printed circuit boards

◆ Data sheets
  • Tour of typical data sheet values
Hardware Schematics For Digital Electronics

◆ Conventions:
  • Chips are rectangles (except small logic gates); inputs on left; outputs on right
  • Pin numbers shown to make wiring easier
  • Thick blue line indicates a “bus” (8 wires bundled into one in this case)

Schematic Capture Tools

◆ OrCad or other professional-grade tools
  • (Schematic on previous page drawn with demo OrCad)

◆ Free tools from PCB vendors (“Printed Circuit Board”)
  • For example, www.expresspcb.com (although I’ve never used their actual board service); there are several such vendors
  • http://www.freepcb.com/ open source (GPL)
  • Search term: printed circuit board prototype
Stuff that goes around a microcontroller/overview

- A microcontroller can’t do much without surrounding parts
  - Even though it has a lot of things already built into it
  - Let’s talk about how you hook a chip up into a system

CPU 12 Microcontroller

- The actual “CPU” is only a part of the chip
  - Many peripherals and memory already integrated onto the chip
Power

- DC electrical power to run the CPU and power I/O circuits
- Some standard voltages
  - 5V DC – old style from the first commercial logic chips, but still in common use ("TTL logic levels")
  - 3.3V DC – common in newer designs
  - Lower voltages often used for low power
- MC9S12C family:
  - 5V for Analog functions and 5V interface
    - 2.97 to 5.5 volts allowed; can run at 3.3V
  - 2.5V for internal logic
    - 2.35 to 2.75 volts allowed
  - 25 mA maximum per pin drive current
  - Multiple power/ground pins
- Actual voltages used depend on power strategy
  - Will this chip run on two NiCd or NiMH batteries? Does that provide 3V?
Embedded Power Supplies

- **Battery**
  - Primary battery – alkaline is 1.5V nominal
  - Secondary battery (rechargeable) – NiMH is 1.2V nominal (so is NiCd)

- **Wall transformer**
  - A/C to DC conversion (a.k.a. “wall wart”) – usually 5V to 12V DC output

- **Sometimes, on-board battery recharging (e.g., solar cells)**

- **Need DC voltage regulation – even for batteries**
  - Battery voltage isn’t constant
  - Nominal rating at mid-point voltage
  - 4 @ NiMH cells 1.2V nominal => 4.8V
  - Mostly discharged, 1.1V/cell => 4.4V total

- **On-circuit board power regulation:**
  - Usually DC to DC converters
  - “Boost” converter increases DC voltage
    - Usually inefficient
    - But, reduces # of battery cells needed
  - “Buck” converter decreases DC voltage

High Power or High Energy (but not both)

Oscillator/“clock”

- Periodic square wave for clocking the CPU
  - Has an internal low-quality oscillator
  - Permits use of an external high-quality oscillator
  - Why both options? -- cost

- MC9S12C family oscillator speeds
  - 0.5 to 16 MHz on internal oscillator
  - 0.5 to 40 MHz external (input must be ~50% duty cycle)
    - Each edge triggers internal actions
  - XCLKS used at startup to select internal vs. external oscillator
  - Why “slow” clock speeds?
    - Keeps costs down – can use old process tech
    - Saves power; reduces need for cooling
    - Fast enough for many purposes
    - Reduces emitted radio interference

I/O bus: data; address

- We’re using a smaller package (48 pin) to reduce costs
  - That package doesn’t put the memory bus on the pins
Available I/O pins

- Pins are used for multiple purposes to reduce packaging costs
  - Configuring chip to put right signals on the right pins is chapter 2 of data sheet
    - It’s really tricky and confusing
  - Initially, we’ll give you code to set up chip the right way for labs

- Most important pins:
  - PAD00..PAD07 // AN00 .. AN07 provides digital and analog I/O
  - PT0..PT7 provides additional digital I/O
  - PW0..PW3 provides hardware-assisted pulse generation

- Data sheet lists all the signals

---

Registers & Memory Maps

- How do you get data on and off the pins?
  - Interface to I/O is done via “registers” (a set of flip-flops on the chip)
  - Write to registers to configure the pins – e.g., is it digital or analog?; in or out?
  - Read/write other registers to actually do I/O
    - Read a byte from switches by reading register associated with digital inputs
    - Write a byte to LEDs by writing a register associated with digital outputs
    - But in both cases, first configure I/O via setting some register, then read/write values from a different register

- How do you access these registers?
  - In some processors, an I/O instruction (x86: IN and OUT)
  - In our processor, I/O is “memory mapped”
    - Use “load” and “store” instructions to special memory addresses

- A memory map tells you where things are in memory
  - Some of memory is RAM
  - Some of memory is ROM
  - Some of memory is I/O register space
  - Look for the memory map in the data sheet. Lots more detail in later lectures
Chip packaging

• 48-pin LQFP (low-profile, quad, flat package)
  • (surface mount – pins soldered to top layer of circuit board)
  • Pin 1 is found at the molded circle in the package
Which one is pin 1???

- **DIP = “dual in-line package”**
  - Through-hole mounting – chip pins go through circuit board
  - For N-pin DIP, ground is pin N/2; power is pin N
    - e.g., 20-pin DIP – pin 10 is ground; pin 20 is power
  - *If you put the chip in backward, you reverse power & ground; smoke ensues*

Breadboards/proto-boards

- **Simple to use – push in wires, DIPs, other components**
  - Easy to use, but fragile (easy to pull out wires)
  - After a few hundred insertions, they wear out

- **Tips:**
  - Try to keep wires neat
  - Put all pin 1s to the left (if horizontal) or top (if vertical)
**How to insert a chip in a socket or proto-board**

- **Pins are further apart than socket holes**
  - Dimension “H” is bigger than Dimension “D”
  - but sockets are sized for Dimension “D”
  - This keeps pins from dropping out of holes for printed circuit boards without sockets – but it makes using sockets a problem

- **To insert a chip**
  - Touch something metal first to discharge any static (in industry, use a grounding strap on your wrist or ankle)
  - Use a chip insertion tool if you have one (it pushes the pins straight)
  - OR gently bend the pins together using a flat table top so they are straight
  - Push the DIP in, making sure than no pins get bent under

- **To remove a chip**
  - Use a chip removal tool if you have one
  - Else use a small screwdriver to pry the chip loose at each end, then rock it free

---

**Other Prototyping techniques**

- **“Perf board”**
  - Boards with “perforated” (punched) holes on 0.1” centers
  - Can put in sockets and solder wires to make connections

- **Wire wrap**
  - Usually perf board, but with special sockets with long square pins
  - Wire wound around the square pins makes the connections
  - Pins are long enough to fit three wrapped wires

- **Printed circuit boards**
  - You can get ‘quick turn’ boards in small numbers fairly inexpensively
  - But, making changes is painful
**Printed circuit boards**

- One or more sheets of thin fiberglass coated with copper on both sides
  - Copper is etched away to leave circuit traces and “pads”
  - Holes are drilled through to make “vias” and places for DIP pins
  - Insulation between fiberglass is “prepreg” – pre-impregnated bonding layers
- Good idea to have plenty of power and ground
  - Usually want dedicated ground layer & dedicated power layer

*Multi layer Design*

**Through-hole vs. surface mount**

- Through-hole
  - DIP pins and resistor leads, etc. go all the way through the PCB
  - Each pin eats up space on every layer of the board
  - Older technology – requires wide pin spacing and works poorly with more than about 8 layer PCBs

- Surface mount
  - Pins only attach on top layer
  - Finer pitch pins, higher density
  - Newer technology
  - Difficult (or with Ball Grid Array pretty much impossible) to hand-solder prototypes without using sockets.

[Wikipedia]
Getting Components On To The Board

- **Wave soldering:**

  RoHS: Restriction of Hazardous Substances (get the lead out of solder)

  [Jones94]

- **Reflow Soldering:**

  [Jones94]

Layout tools & challenges

- **Layout is a difficult 2.5-dimension puzzle**
  - K layers, where K is usually even
  - Pretty much like IC layout – very similar algorithms
  - Auto-routers have gotten better over time, but still some hand-assist
  - Using a CAD tool, each layer is a different color

- **Common design strategies**
  - Dedicated layer for power and for ground (helps with noise)
  - Even layers are mostly horizontal, odd layers are mostly vertical

[eeinternational.net]
External connectors: headers; edge connectors

- **Edge connectors**
  - Plated areas of circuit board
    - Usually gold or copper
    - Tin tends to corrode – unreliable
  - Old-style PCs used these to connect onto motherboard (ISA bus)

- **Headers**
  - A set of posts for connecting
  - Posts insert into sockets or can be clipped, wire-wrapped, soldered to, etc.
  - Especially popular – ribbon cable
    (caution, most ribbon cables only good for a half-dozen insertions before becoming unreliable!)

Switches

- **Single pole, single throw switch**
  - E.g., ordinary house lighting switch
  - It’s “on” or “off” and switches a single line
  - Stays “on” or “off” unless moved

- **Double pole, single throw switch**
  - 3-way house lighting switch
  - Either “side A on” or “side B on”
  - Stays on A or B once there, unless moved

- **Momentary switch**
  - “normally open” – “off” normally, “on” when pressed
    - These are the switches on the lab project board
  - “normally closed” – “on” normally, “off” when pressed
    - The brake pedal on your car is this type of switch
Data sheets overview

Data sheets are the roadmap to a chip
• Vary between 1 sheet and 500+ sheets
• Every circuit part has a data sheet – even a resistor or socket
• In industry, there is a library of data sheets for approved parts
  (and getting a new part approved is a huge deal – so you use parts that are already approved)

Data sheet content
• Pinout
• Physical characteristics (package size, pin type, etc.)
• Electrical characteristics
• Thermal limits
• Etc.

MC9S12C data sheet Appendix A has electrical characteristics

Key items in data sheet: speeds

Propagation delays
• Low to high and high to low are sometimes different speeds
• Speeds depend on operating conditions

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>TEST CONDITIONS</th>
<th>(V_{CC} ) (V)</th>
<th>25°C</th>
<th>40°C TO 85°C</th>
<th>-40°C TO 125°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN TYP MAX MIN MAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Propagation Delay</td>
<td>( \uparrow \text{PHL, } \downarrow \text{PHL} )</td>
<td>( C_L \geq 50 \text{pF} )</td>
<td>2</td>
<td>-</td>
<td>165</td>
<td>205</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.5</td>
<td>-</td>
<td>33</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>-</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>-</td>
<td>28</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( C_L = 15 \text{pF} )</td>
<td>2</td>
<td>-</td>
<td>135</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.5</td>
<td>-</td>
<td>27</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( C_L = 15 \text{pF} )</td>
<td>5</td>
<td>-</td>
<td>11</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( C_L = 50 \text{pF} )</td>
<td>6</td>
<td>-</td>
<td>23</td>
<td>29</td>
</tr>
</tbody>
</table>
Outputs & Inputs

◆ Regular output driver and input buffer:
  • Amplifies on-chip to off-chip current capability
  • Usually just an inverter with big (high drive) transistors
    – (real chip implementations more complex; but that’s the basic idea)

◆ Tri-state output drivers:
  • Has ability to output “0”, “1”, or “Hi-Z” (off)
    – CL high turns it on, propagating input A
    – CL low turns output off (Hi-Z)
  • Allows multiple chips to drive the same signal

Key items in data sheet: electrical specifications

◆ Be sure to look for
  • Power consumption when running
  • Input and output parameters, especially
    – Input switching thresholds: $V_{IH}$, $V_{IL}$
    – Output drive currents: $I_{OH}$, $I_{OL}$ (in mA or ‘standard loads’)

<table>
<thead>
<tr>
<th>DC Electrical Specifications (Continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Quiescent Device Current</td>
</tr>
<tr>
<td>Three-State Leakage Current</td>
</tr>
<tr>
<td>HCT Types</td>
</tr>
<tr>
<td>High Level Input Voltage</td>
</tr>
<tr>
<td>Low Level Input Voltage</td>
</tr>
</tbody>
</table>

[TI]
Key Voltage & Current Specifications

- **Input switching thresholds:**
  - $V_{IH} =$ Input Voltage that is seen as a “high” input
    - For example, any input above 2 Volts is High; $V_{IH} = 2V$
  - $V_{IL} =$ Input Voltage that is seen as a “low” input
    - For example, any input below 0.8 Volts is Low; $V_{IL} = 0.8V$
    - In these examples: anything between 0.8V and 2.0V is indeterminate

- **Output drive currents:**
  - $I_{OH} =$ Current driven if output is high
    - For example, a high output at 4.5V might drive .5 mA
  - $I_{OL} =$ Current driven if output is low
    - For example, a low output at 0.5V might drive 25 mA
    - $I_{OL}$ is often much higher than $I_{OH}$! And usually only a few mA for TTL chips
    - But, the course microcontroller has both at 25 mA
**Noise Margin: Input vs. Output Voltage**

- Real chips have three output voltages that matter:
  - High enough to be a “1” -- \( V_{oh} > V_{ih} \)
  - Low enough to be a “0” -- \( V_{ol} < V_{il} \)
  - Something in between – a “I’m not sure if I’m a 1 or a 0” voltage (this is **Not Good**)

![Noise Margin Diagram]

[Weste95]

---

**Fanout – How Many Gates Can You Drive?**

- Fanout of 3 means an output can drive at most 3 inputs
  - E.g., \( V_{ol} < V_{il} \) and \( \text{sum}(I_{il}) < I_{ol} \)
  - If you violate fanout limits, you get an indeterminate voltage (or sometimes, you get smoke)

![Fanout Diagram]
**Decoupling Capacitors**

- Use decoupling caps to reduce switching noise
  - Provides fast-response temporary power supply for switching
  - Left: no cap  Right: with capacitor

![Image of voltage and current traces](http://www.vagrearg.org/?p=decoupling)

Top trace is voltage  
Bottom trace is Current


---

**Ground Bounce**

- CMOS power is largely consumed halfway through switch
  - Momentary near-short between power and ground when switching – near zero resistance
  - This causes power to be pulled down and ground to be pulled up
    - Decoupling caps help with this too!
  - Note that ground bounce of \( >V_{II} \) on TTL causes “maybe” values

![Diagram of ground bounce](http://www.altera.com/literature/wp/wp_grndbnce.pdf)

Decoupling ("bypass") capacitor placement

- Put capacitors as close to chip as possible
  - Minimize total wire length between power and ground pins
  - (picture is suggested layout for course CPU – note extra-wide ground trace!)

Noise issues

- Electronic “noise” is a fact of life
  - Digital signals are a nice fiction; real signals are analog
  - Inductive/capacitive coupling among circuit traces
  - Switching transients affecting power supplies

- Good design practices (for noise and other matters):
  - Use decoupling capacitors to act as mini-power-supplies for chips
    - Use a capacitor as close as possible to power/ground pin pair on a chip
    - Generally this is enough under 50 MHz (usually 0.01 to 0.1 uF)
    - Above 50 MHz more care is required (but most small embedded systems are slow)
  - Separate analog and digital portions of the PCB (don’t intermix traces)
    - Video, radio, and backlight power traces are especially nasty radiators of noise
    - Audio is especially sensitive to picking up interference from other traces
  - Run ground traces on all sides of critical lines
  - Dedicated layers for power and ground planes or grids
  - Socket external interface chips that could get burned out via transients
  - Put power on an un-populated PCB to check for power/ground faults
How Much $I_{OL}$ is Enough?

• Assume 2V drop across LED – how much current is drawn?
  
  • Assume $V_{OL} = 0.5V$
  
  • Reminder – Ohm’s Law: $V=IR$

• What is $I_{OL}$?

• Does $I_{OH}$ matter for this circuit?
**Lecture 2 Review**

- **General pinout of course microcontroller**
  - Types of pins
    - But not “what does pin 17 do” without a pinout diagram
  - General voltages, speeds, packaging

- **General electronic hardware**
  - Packaging types
  - Where’s pin 1 on a package?
  - Printed circuit board construction and related topics
  - Circuit parameters and meanings (e.g., what does “I_{OH} = 4 mA” really mean?)
  - Be able to compute current through an LED
    - LED components are most expensive after CPU – almost $5 apiece
    - Over-driving CPU outputs can easily burn out CPU module (about $75)
    - Use resistors with LEDs – and get resistor value right!
  - Good design practices

**Lab Skills**

- **Be able to hook a simple circuit on a proto-board**
  - D-register (including chip insertion into the proto-board)
  - LEDs
  - Resistors