EECS 373
Introduction to Embedded System Design

Robert Dick
University of Michigan

Lecture 3: Logic, MMIO, and project

23 January 2024
Outline

- Logic review
- Memory-mapped I/O
- Project
Logic review

• If you are ever stuck on a design problem involving a combinational network, write the truth table.

• If you are ever stuck on a design problem involving a sequential network, draw the state diagram.
Gates

NAND

NOR

AND

OR

NOT

XOR

XNOR
De Morgan's Laws and bubble pushing

\[(ab)' = a' + b'\]
\[(a + b)' = a'b'\]

Example:
\[a'b' + b'c + ac\]
\[((a'b)' + (b'c)' + (ac)')'\]
Outline

- Logic review
- **Memory-mapped I/O**
- Project
Old-style I/O

- Special instructions for reading/writing peripherals.
- Wasteful: Already have load/store instructions.
Lab 3: Memory-mapped I/O

- Put peripherals at memory addresses.
- Turn LED turn on by writing 1 to address 5.
- Read button state (active-high) at address 4.
- Use a bus on which the peripheral sits.
Bus access example

• Bus protocol.
• Asynchronous (no clock).
• Initiator and Target.
• REQ#, ACK#, Data[7:0], ADS[7:0], CMD.
  • CMD=0 is read, CMD=1 is write.
  • REQ# low means initiator is requesting something.
  • ACK# low means target has done its job.
A read transaction

• Initiator wants to read location 0x24.
• Initiator sets ADS=0x24, CMD=0.
• Initiator then sets REQ# to low.
  • Delay first.
• Target sees read request.
• Target drives data onto data bus.
• Target then sets ACK# to low.
• Initiator grabs the data from the data bus.
• Initiator sets REQ# to high, stops driving ADS and CMD.
• Target stops driving data, sets ACK# to high terminating the transaction.
Read transaction

<table>
<thead>
<tr>
<th>ADS[7:0]</th>
<th>CMD</th>
<th>Data[7:0]</th>
<th>REQ#</th>
<th>ACK#</th>
</tr>
</thead>
<tbody>
<tr>
<td>??</td>
<td></td>
<td>0x55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A B C D E F G HI
Write transaction

(write 0xF4 to location 0x31)

• Initiator sets ADS=0x31, CMD=1, Data=0xF4.
• Initiator then sets REQ# to low.
• Target sees write request.
• Target reads data from data bus.
• Just has to store in a register, need not write all the way to memory!
• Target then sets ACK# to low.
• Initiator sets REQ# to high & stops driving other lines.
• Target sets ACK# to high terminating the transaction.
The push-button
(if ADS=0x04 write 0 or 1 depending on button)

Button (0 or 1)
The LED
(1 bit reg written by LSB of address 0x05)

Flip-flop which controls LED
What does this look like from software perspective?

```c
volatile char * button_ads = (char *)(0x24);

char read_button(void) {
    return *button_ads;
}
```

```
.equ BUTTON_ADS, 0x24

movw r0, #:lower16:BUTTON_ADS
movt r0, #:upper16:BUTTON_ADS
ldr r1, [r0, #0]
```
Outline

- Logic review
- Memory-mapped I/O
- Project
Project timeline

- 29 Jan: Every student finishes preparing three, high-level, half-page shared proposals.
  - Customer
  - Values
  - Approach
  - Components
- 30 Jan: Team formation meeting.
- 6 Feb: Draft proposals on which we provide feedback
- 13 Feb: Final proposals about which we will meet.
- 27 Feb: Parts list ready for ordering and submitted.
- 14 Mar: Checkpoint 1.
- 4 Apr: Checkpoint 2.
- 23 Apr: Demo day.
Problem selection

- Early in the course, many EECS 373 students feel that they are too inexperienced to make something useful to others.
- **That’s wrong!!!** You will be able to do amazing things before the course is finished.
- Ideal: Find a problem you understand and want to solve for your own sake, that many other people also want to have solved for them.
- But this takes some work / interview up-front.
- Acceptable: Find a problem you understand and want to solve for your own sake.
Identifying important problems

• The easy, luck-dependent way
  • Pick a problem that you understand deeply and that is very important to you.
  • Only works if many people are similar to you.
  • That's rare, especially for engineers.

• The easy, risky way
  • Take the problem from a competitor.

• The hard, more reliable way
  • Customer discovery interview process
  • Counterintuitive. Hard.
  • Social pressures undermine the process.
How you see your idea
How others see your idea
How others see you
That's a great idea!

• That's a great idea!
  • I don't want to hurt your feelings you cute little engineer.
• I would definitely use that!
  • There are almost no circumstances in which I would use that, let alone consider paying for or supporting it.
• I have some ideas on how to make it better!
  • I'm pretending to be someone who would use it and leading you down a false path.
• Follow your dreams man!
  • Squander your life, man!
If you want to follow the reliable path

• Talking to Humans by Giff Constable
• The Startup Owner's Manual by Steve Blank and Bob Dorf
• Market Research on a Shoestring by Naeem Zafar
• Steve Blank's online videos (steveblank.com)
• Contact me for more
What do they teach?

• How to design, sequence, and deliver questions to minimize bias (both yours and theirs).
• How to talk with strangers.
• This is not natural. Most must learn it.
Done.