Announcements

• Lab 3a due tomorrow

• In-class bonus questions
  – Bonus point (= 1 extra slip day on any of the remaining assignments)
Review: Critical Path of 32-bit CLA

Key insight: Carry generation and propagation functions (e.g., $G_{4:0}$, $P_{4:0}$) in a 4-bit CLA block do not depend on results from other CLA blocks.
Review: Arithmetic Logic Unit (ALU)

<table>
<thead>
<tr>
<th>OP Name</th>
<th>OP</th>
<th>BSEL</th>
<th>CI</th>
<th>LOP</th>
<th>SOP</th>
<th>OSEL</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>000</td>
<td>00</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>00</td>
<td>Y = A + B + CI</td>
</tr>
<tr>
<td>SUB</td>
<td>001</td>
<td>01</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>00</td>
<td>Y = A + B' + 1</td>
</tr>
<tr>
<td>AND</td>
<td>011</td>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>01</td>
<td>Y = A AND B</td>
</tr>
<tr>
<td>OR</td>
<td>100</td>
<td>00</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>01</td>
<td>Y = A OR B</td>
</tr>
<tr>
<td>SHL</td>
<td>101</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>10</td>
<td>Y = A[6..0],SI</td>
</tr>
<tr>
<td>SHR</td>
<td>110</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>10</td>
<td>Y = SI,A[7..1]</td>
</tr>
<tr>
<td>PASS A</td>
<td>111</td>
<td>10</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>00</td>
<td>Y = A</td>
</tr>
</tbody>
</table>

![ALU Diagram](image)

Lecture 14: 4
Comparison Operations (Byproduct of SUB)

- To compare A and B, perform $A - B$
  - If the result is 0, then $A = B$
  - Z flag set to 1 whenever ALU result is 0
  - Can check for $A \geq B$ and $A < B$ by observing the MSB of the result of $A - B$
Our Microprocessor Needs (Large) Memory
General Memory Organization

- **Two dimensional array of bit cells**
  - Each bit cell stores 1 bit

- **Address selects a word with multiple bit cells**

![Diagram of memory organization with an array of bit cells and an example data arrangement.](image)
General Memory Structure

2:4 decoder

Address

2:4 decoder

Only one wordline is asserted at any given time
Types of Memories

- **Read-Only Memory (ROM)**
  - Truly read-only
    - Written in the factory, and never written after installation
  - Mostly read and rarely written
    - Much faster to read than write
  - Usually non-volatile

- **Random Access Memory (RAM)**
  - Dynamic RAM (DRAM), Static RAM (SRAM)
  - Read and write any location at similar speeds
  - Typically volatile: loses contents when powered off
    - New classes of non-volatile/persistent RAMs are emerging
Read-Only Memory (ROM) Structure

- n address lines
- decoder
- $2^n$ word lines
- memory array ($2^n \times m$)
- m bit lines

*fixed at factory or rarely written*
Example ROM Implementation

- Active-high decoder outputs
- Word line
- Resistor network: Each NMOS "hardcodes" a 1
- Address inputs
- Decoder
- Active-low bit lines, pulled low if transistor present
- Bit line
- Data output
Applications of ROM

• **Data & program storage**
  – e.g., Boot configuration for personal computers or application code for embedded systems

• **Combinational logic functions**
  – Lookup table
    • Address inputs = function inputs
    • Data outputs = function outputs
Using ROMs for Combinational Logic

F0 = A' B' C' + A B' C' + A B' C
F1 = A' B' C' + A' B C' + A B C
F2 = A' B' C' + A' B' C + A B' C'
F3 = A' B C' + A B' C' + A B C'

A B C | F0 F1 F2 F3
0 0 0 | 0 1 0 0
0 0 1 | 1 1 0 0
0 1 0 | 0 1 0 0
0 1 1 | 0 0 0 1
1 0 0 | 1 0 1 1
1 0 1 | 0 0 0 0
1 1 0 | 0 0 0 1
1 1 1 | 0 1 0 0

Decoder

Memory array (8 words by 4 bits)

8 word lines

F0 F1 F2 F3
ROM As a “Random-Logic” Circuit

- **ROM advantages**
  - Design time is short
  - Can implement any function of $n$ inputs

- **ROM problems**
  - Size doubles for each additional input
  - Cannot exploit logic minimization (e.g., don't cares)
Types of Memories

• **Read-Only Memory (ROM)**
  – Truly read-only
    • Written in the factory, and never written after installation
  – Mostly read and rarely written
    • Much faster to read than write
  – Usually non-volatile

• **Random Access Memory (RAM)**
  – **Dynamic RAM (DRAM), Static RAM (SRAM)**
  – Read and write any location at similar speeds
  – Typically volatile: loses contents when powered off
    • Note: new emerging classes of non-volatile/persistent RAMs are emerging
DRAM Modules

- DRAM modules are typically off-chip and mounted on a separate printed circuit board
DRAM Bit Cell

- Capacitor accessed through a transistor
- Capacitor is charged or discharged to produce a 1 or a 0
- DRAM cells lose their state over time and must be refreshed periodically, hence the name *dynamic*
DRAM Write

• **Word line is asserted**

• **Bit line is driven with the desired value**
  – Capacitor is charged by the bit line to store a 1
  – Or discharged by the bit line to store a 0
DRAM Read

• Word line is asserted

• Bit line *precharged halfway* between 0 and 1
  – Capacitor voltage pulls the bit line slightly higher or lower
  – *Sense amplifier* detects this small change (1 or 0)

• Read destroys the stored value
  – Need to rewrite the value afterwards
Common Structure of a DRAM Array*

• **Multiplexed address inputs**
  (log\(_2\)N bits for an NxN bit cell array)
  - **Row & column addresses share the same pins**
    - Row address (addr) bits arrive first; followed by column addr

- **Row-address strobe (RAS)**
  - RAS loads Address into row addr latch to select 1 row

- **Column-address strobe (CAS)**
  - CAS loads Address into col addr latch to select 1 column bit

Row and column address bits combined form the complete address
Read Access to a DRAM Array

DRAM bit cell array with 1024 rows x 1024 columns (1Mb)

1. 10 row address bits arrive first; RAS asserted
2. 1024 bits are read out to row data latch
3. 10 column address bits arrive later; CAS asserted
4. 1 bit returned
DRAM Refresh

• Capacitors discharge over time
  – Bit cells must be periodically refreshed (e.g., every 64ms, technology dependent)

• Refresh cycles recharge each memory bit
  – An entire row is refreshed at a time – each row periodically accessed using RAS (i.e., a read access), which restores the charge

![Diagram showing voltage changes over time with capacitors discharging and refreshing.]
Static RAM (SRAM) Bit Cell

• **One cell requires six transistors**
  – The core is two cross-coupled inverters

• **Maintaining the state of the cell requires a constant power**
  – The cell is stable; no refresh cycles needed

6-Transistor SRAM Cell

BL = bit line
WL = word line
SRAM Write

- Drive one bit line high, the other low (depending on the desired value)
- Then turn on word line
- Bit lines over power cell with new value
SRAM Read

- **Precharge both bit lines high**
- Then turn on word line
- **One of the two bit lines will be pulled down by the cell**
  - Change detected by sensor amplifier
SRAM Architecture*

- Address Decoder
- Write Enable
- Column Driver
- Sense Amp
- Din_n-1
- Din_n-2
- Dout_n-1
- Dout_n-2
- ...
SRAM vs. DRAM

• **SRAM advantages and disadvantages**
  Usually on the same chip with microprocessor
  (+) Fast access
  (−) High power
  (−) Relatively high area & cost per bit

• **DRAM: main memory is stored in DRAM cells**
  Typically off-chip
  (+) Single transistor storage cell
    • Higher density lower cost/bit
    • Lower power/bit
  (−) Slow
    • Need periodic refresh to retain data
Before Next Class

• H&H 7.3.2-7.3.4

Next Time

Single Cycle Microprocessor