CMPE12 Final Review

MSI Review session
Friday Dec 2nd 5:00-7:00pm
E2 180 (Simularium)
Evaluations

• They are online and are due before the Final
• Please fill them out
  – Not asking for good opinions, but honest ones
  – Attempting to fight self-selection bias
• While your at it (a site very much victim to it)
  – https://www.ratemyprofessors.com

you are
1) nothing
2) wrong path
3) no labels

LDR
ADDI
NOT
BR
Overview

- Tuesday, December 6, 7:30–10:30 p.m.
- Same policy as before: no notes, books or calculators
- Extra Credit
  - If score is over 100% will still help final grade
  - Two Problems
    - One repeat from the midterm exactly
    - One random one
- Show your work
  - No credit if we can’t figure out how you did it
  - Partial credit
Partial List of Topics

- N-type, p-type transistors
- Realization of truth table from transistors and inverse
- Transistors to standard gates and inverse
- Truth table to gates and inverse
- Sum-of-Products and Product-of-Sums
- Boolean Algebra
- Common Logic elements
  - Mux etc including sequential

- Binary representations
  - Unsigned
    - Binary
    - Hex
    - Octal
  - Signed
    - Two’s complement
    - One’s complement
    - Sign magnitude
  - Bias

- Binary Math
  - Overflow indications
Partial List of Topics Continued

• Floating Point
  – Conversions
  – Addition, Subtraction, Multiplication
• LC-3 Architecture
• LC-3 Assembly
  – Opcode translation
  – LC-3 coding and running
  – Subroutine methods
  – Basic data structures
• MIPS
  – Role of registers
  – Theory of Function Calls
• Embedded Uno32
  – I/O
  – Special Purpose Registers (TRIS, etc.)

Theory
Transistor and Gates
Truth Table to Gates

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Diagram of gates corresponding to the truth table.
Transistor and Gates

Truth Table to Transistors

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Transistors and Gates
Transistors to truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
\((A+B) < \)
Logic Elements to Gates

- Draw the gate level diagram of a 2-4 decoder
## Base Conversion Table 8 bits

<table>
<thead>
<tr>
<th>Decimal</th>
<th>1’s Complement</th>
<th>2’s Complement</th>
<th>Signed Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>-35</td>
<td>X</td>
<td>0110 0001</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001 1101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td>10011101</td>
</tr>
</tbody>
</table>
## Base Conversion Table 8 bits

<table>
<thead>
<tr>
<th>Decimal</th>
<th>1’s Complement</th>
<th>2’s Complement</th>
<th>Signed Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>-35</td>
<td>X</td>
<td>0110 0001</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>1001 1101</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>10011101</td>
<td></td>
</tr>
</tbody>
</table>
Binary Arithmetic

- Unsigned
- 2’s Complement
- Signed Magnitude
\[
\begin{align*}
\forall N \in \mathbb{R} & \quad +N + -N \Rightarrow +N - +N \\
& \quad +N - -N \Rightarrow +N + +N \\
& \quad n - N = -(N - n)
\end{align*}
\]
Fractional Representation

4.6 in Base 2?

100.1001

.625 + .5 + .125

.6 + .2 = 1.2
.2 + .2 = 0.4
.4 + .2 = 0.8
.8 + .2 = 1.6
Binary Division

\[ \begin{array}{c|c|c|c|c|c|c} & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline \end{array} \]

- 101 | 101110111
  \[ \begin{array}{c|c|c|c|c|c|c} & 1 & 0 & 1 & 0 & 1 & 1 \\
\hline 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
\hline - & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline \end{array} \]

- 101 | 101110111
  \[ \begin{array}{c|c|c|c|c|c|c} & 1 & 0 & 1 & 0 & 1 & 1 \\
\hline 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
\hline - & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline \end{array} \]
Boolean Algebra

\[ XY + X(Y + Z) + Z(X+X) + (X + X')YZ + Y + Y'Z \]

\[ \begin{align*}
  XY &+ xY + YZ + xZ + YZ + X + Y + Y'Z \\
  XY &+ Z(x + y) + Y + Y + y + Z \sqrt{y + z} \\
  xY &+ Z + Z + Z(x + z) \end{align*} \]

\[ \begin{align*}
  y(x + 1) + 2z(x + z) \\
  \overline{y + z} \end{align*} \]
$3 \times 3$

$7 \times 7$
Floating Point Format

Representation:

```
  31 30  23 22     0
    |     |       |
    S   E     F
```

- $S$ is one bit representing the sign of the number
- $E$ is an 8 bit biased integer representing the exponent
- $F$ is an 23-bit unsigned integer

The true value represented is: $(-1)^S \times f \times 2^e$

- $S =$ sign bit
- $e = E - \text{bias}$
- $f = F/2^n + 1$
- for single precision numbers $n=23$, $\text{bias}=127$
Floating Point Conversion

• What is the decimal value for this SP FP number 0x421A 0000?

\[
\begin{align*}
\text{0100 00100 001 1010 00} & \rightarrow 0 \\
128 + 4 - 127 & = 5 \\
1.0011101 & = 38.5 \\
2^{-1} = \frac{1}{2} &
\end{align*}
\]
$s = 1$

Floating Point Conversion

$s_{25} - s_{12} = 13$

- What is the SP FP value of the decimal value -525.5?

1. $110000001101.1$
2. $9 + 127 = 136 - 128 = 8$
3. $10001000$
4. $11001000000000110110 \rightarrow 0$
5. $0x<4036000$
Floating Point Math

\[ \begin{array}{c}
0.1 \times 10^2 \\
0.1 \times 10^3 \\
\end{array} \]

\[ \begin{array}{c}
0100 \\
0101 \\
\end{array} \quad 1111 \\
\begin{array}{c}
0100 \\
0101 \\
\end{array} \quad 0110 \\
\begin{array}{c}
0100 \\
0101 \\
\end{array} \quad 1011 \\
\begin{array}{c}
0100 \\
0101 \\
\end{array} \quad 0000 \\
\end{array} \]

\[ \begin{array}{c}
0 \rightarrow 0 \\
0 \rightarrow 0 \\
\end{array} \]

\[ \begin{array}{c}
1.11010111 \\
1.11111111 \\
\end{array} + \begin{array}{c}
1.11101011 \\
10.11101010 \\
\end{array} \]

\[ \begin{array}{c}
10001100 \\
10001100 \\
\end{array} \]

\[ \begin{array}{c}
130's \end{array} \]

\[ \begin{array}{c}
0100011000 \\
01111010101 \\
\end{array} \]
Floating Point Math part 2

0x45FFC000
0x456B0000

1) Add exponents
2) Multiply mantissas (Hβ)
3) Re-normalize
Binary Arithmetic

- Unsigned
  
  \[
  \begin{array}{c}
  10011111 \\
  00110101 \\
  \hline
  11110101
  \end{array}
  \]

- 2’s Complement
  
  \[
  \begin{array}{c}
  01110101 \\
  11110001 \\
  \hline
  01110100
  \end{array}
  \]

- Signed Magnitude
  
  \[
  \begin{array}{c}
  01100101 \\
  + 00011111 \\
  \hline
  10000001
  \end{array}
  \]
Arbitrary Base Conversion

- $1210_3$ in base 10
LC-3 Assembly Coding

• Write LC-3 assembly code that will OR the values in R1 with R3 and store the result in R0.
LC-3 Sub-Routine Coding

- Write the Load Function from Lab 4
  - Assume R0, R1, R2 are used as arguments
  - Be sure to save off registers used
  - Label Base has first address of array and Label Size holds the column length
LC-3 Data Structures
Array, Stacks and Queues

- Basic theory of each
  - Understanding of how to write basic routines
1. (10pts) LC-3 ISA

After the following LC-3 code executes what are the ending contents of the registers and memory? Assume some registers/memories have starting values as indicated. If blank, the content is unknown. Remember that both registers and memory locations are 16-bits wide. The memory portion starts at address 0x3200.

```
LEA      R1, label0
LDR      R2, R1, #0
STR      R0, R1, #4
LEA      R6, label12
ADD      R5, R0, R1
LEA      R0, label1
AND      R7, R2, R5
NOT      R3, R0
STR      R7, R6, # -2
STR      R2, R1, #1
```

<table>
<thead>
<tr>
<th>Register</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>0x1234</td>
</tr>
<tr>
<td>R1</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td></td>
</tr>
</tbody>
</table>

Memory

<table>
<thead>
<tr>
<th>Label</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>label0</td>
<td>0xDEAD</td>
</tr>
<tr>
<td>label1</td>
<td></td>
</tr>
<tr>
<td>label2</td>
<td></td>
</tr>
</tbody>
</table>
PIC32 Architecture

- Word size, register count, similar to questions asked about the lc-3
- Standard registers available and their uses
PIC32 Ports

• What do TRIS LAT and PORT do?
• Pseudocode on how to use them
PIC32 Function Calls

Stack

1) a ...
2) save/restore registers
   - save: push
     restore: pop
$<sp>