Lecture 4
Review on Digital Logic (Part 3)

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese461/
Fixed Point Multiplication

- Two Q15 number multiply
  - \( Q_{15} \times Q_{15} = Q_{30} \)
  - 2.30 format, 32 bits, two sign bits
  - MSB: extended sign bit
  
  \[
  \begin{array}{c}
  \text{Q15} \\
  \hline
  \text{S} \\
  \end{array}
  \quad \times \quad \begin{array}{c}
  \text{Q15} \\
  \hline
  \text{S} \\
  \end{array}
  \quad \begin{array}{c}
  \text{Q30} \\
  \hline
  \text{S} \quad \text{S} \\
  \end{array}
  \]

  - need to truncate back to 1.15 format
  - left shift by one bit, storing upper 16 bits
  - right shift by 15 bits, storing lower 16 bits

  \[
  \begin{array}{c}
  \text{Q30} \\
  \hline
  \text{S} \quad \text{S} \quad \text{r} \\
  \end{array}
  \]

  rounding by addition a '1' here
  \[
  \begin{array}{c}
  \hline
  \text{S} \quad \text{S} \quad \text{r} \\
  \end{array}
  \]
Sequential Circuit Analysis

- Design steps
  - word description
  - state diagram
  - state table
  - select flip-flop types
  - input to FF and output
  - verification

- Reverse engineering
Input Equations

- To flip-flops
  - $D_A = A(t)x(t) + B(t)x(t)$
  - $D_B = \overline{A(t)}x(t)$

- Output $y$
  - $y(t) = \overline{x(t)}(B(t) + A(t))$
State Table

- For the example:

\[ A(t+1) = A(t)x(t) + B(t)x(t) \]
\[ B(t+1) = A(t)x(t) \]
\[ y(t) = x(t)(B(t) + A(t)) \]

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(t) B(t)</td>
<td>x(t)</td>
<td>A(t+1) B(t+1)</td>
<td>y(t)</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1 0</td>
<td>0</td>
</tr>
</tbody>
</table>

Inputs of the table

Outputs of the table

Inputs of the table:

- \( m \): no. of FF
- \( n \): no. of inputs

Outputs of the table:

\( 2^3 \) rows

\( (2^{m+n}) \) rows
State Diagram

- Conventions

**Moore Machine**

Moore type output depends only on state

Example:

\[
\begin{array}{c}
AB \\
y \\
\hline
01 \\
1 \\
\end{array}
\]

**Mealy Machine**

Mealy type output depends on state and input

Example:

\[
\begin{array}{c}
01 \\
\end{array}
\]
Example: Elevator Controller

- **Description of the controller**
  - elevator can be at one of two floors: ground and first
  - one button controls the elevator: up and down
  - two lights floor indicator: red (ground), green (1\textsuperscript{st})

- **State diagram**
- **State table**
- **Boolean expressions**
Outline

Arithmetic Logic

Sequential Logic

Memory Circuit
Memory Devices and Hierarchy

source: copterj.tumblr.com

source: wikipedia
Static RAM

- **Applications**
  - CPU register file, cache, embedded memory, DSP

- **Characteristics**
  - 6 transistor per cell, other topologies
  - no need to refresh
  - access time ~ cycle time
  - no charge to leak
  - faster, more area, more expensive
SRAM Operation

- **Standby**
  - word line de-asserted

- **Read**
  - precharge bit lines
  - assert WL
  - BL rise/drop slightly

- **Write**
  - apply value to BL
  - assert WL
  - input drivers stronger
SRAM Architecture

n-bit word

Bitcell Array

Bitlines for bit n

Column Drivers

 Sense Amplifiers

Source: semiengineering.com
Sense Amplifier:
Differential input

bitline

\begin{align*}
\text{Vdd} \\
\text{Vss}
\end{align*}

bitline

Data Out
Multi-Bank Layout

source: semiengineering.com
DRAM

- **Applications**
  - main memory in desktop, laptop, workstation

- **Characteristics**
  - 1 transistor and 1 capacitor per bit cell
  - need to refresh
  - access time < cycle time
  - slower, less area, cheaper
**DRAM Read**

- disconnect sense amp
- precharge bit lines
- precharge off
- assert wordline
  - transfer charge from cell to bit-line
- detect by sense amp
  - latch output
- read selected column
  - recharge cell
- de-assert word-line
DRAM Write

- select row
- force sense amp to desired value
  - positive feedback
- bit-line to charge cell
- entire row refreshed
Content-Addressable Memory (CAM)

- Associative memory
  - used in high-speed searching application
  - e.g. networking routers

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Address (Binary)</th>
<th>Output Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101XX</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>0110X</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>011XX</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>10011</td>
<td>D</td>
</tr>
</tbody>
</table>

Simplified routing table

- more power, area, circuitry
CAM Operation

- NOR-based CAM architecture
  - precharge matchlines
  - broadcast search data
  - CAM cell compare, matchline pulled down if mismatch
  - generate search address/location
CAM Circuit

• CAM cell

Binary CAM Cell

Ternary CAM Cell

source: pagiamtzis.com/cam/camintro
Summary

Number Representation

Boolean Logic and Gates

Combinational Logic

Arithmetic Logic

Sequential Logic

Memory Circuit
Review Quiz

- 50 minutes
- Closed Book. No references
- No electronic devices (including calculator)
Outline

More Reviews

Linux Basics

VCS Simulator
Linux Lab

- [https://linuxlab.seas.wustl.edu/equeue/](https://linuxlab.seas.wustl.edu/equeue/)

- Remote Linux Desktop
  - virtual network computing (VNC)
  - eQUEUE
  - Javascript, Jave Runtime Environment (JRE)
Linux Basics

- GUI
  - file management
  - folder navigation

- Terminal
  - command-line interface

- Introduction to Linux
  - https://www.edx.org/course/introduction-linux-linuxfoundationx-lfs101x-0
Useful Commands

- **Navigation**
  - `ls`, `cd`, `pwd`, `ln`, `find`, `less`, `more`

- **Manipulation**
  - `cat`, `cp`, `mv`, `grep`, `mkdir`, `touch`
  - `chmod`, `chown`

- **Check status**
  - `df`, `du`, `quota`, `uname`, `history`

- **Set Environment**
  - Unix shell: Bash, C shell (`csh`, `tcsh`), etc.
  - `source .bashrc`
  - `module avail`
Questions?

Comments?

Discussion?