CSE 140 Lecture 15
System Designs

Professor CK Cheng
CSE Dept.
UC San Diego
System Designs

- Introduction
- Components
- Spec
- Implementation
Digital Designs vs Computer Architectures

• Instruction Set (H.Chapter 6, CSE141)
  – Bottleneck: Silicon Area, Power
• Data Path (H.Chapter 7.1-7.3)
• Control Subsystem (H.Chapter 7.1-7.3)
• Memory Management (Chapter 8, CSE141)
  – Bottleneck: IO, Memory Latency
Introduction

• Methodology
  • Approach with success stories
  • Hierarchical designs with interface between the levels

• Data Subsystem and Control Subsystem
  • For n-bit data, each operation takes n or more in complexity
  • Data subsystem carries out the data operations and transports
  • Control system sequences the data subsystem and itself.
I. Introduction
## Introduction

<table>
<thead>
<tr>
<th>Components</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Subsystem</td>
<td>Storage Modules</td>
</tr>
<tr>
<td></td>
<td>Operators</td>
</tr>
<tr>
<td></td>
<td>Interconnections</td>
</tr>
<tr>
<td>Control Subsystem</td>
<td>Sequential machines</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Subsystem Components

• Storage
• Operator
• Interconnect
Components: Storage Modules, Register

\[ Q(t+1) = \begin{cases} 
(0, 0, \ldots, 0) & \text{if CLR} = 1 \\
D & \text{if LD} = 1 \text{ and } \text{CLR} = 0 \\
Q(t) & \text{if LD} = 0 \text{ and } \text{CLR} = 0
\end{cases} \]
Storage Component: Registers, Array of Registers

Registers: If C then R ← D

Array of Registers: Sharing connections and controls
Storage Components: RAM, FIFO, LIFO

RAM

Decoder

Address

RAM

FIFO (First in first out)

LIFO (Stack)

Size of RAM larger than registers
Performance is slower
CASE Op-Sel Is
  When F1, Z <= A op1 B
  When F2, Z <= A op2 B
  .
  .
End CASE
Interconnect Modules (Wires and Switches)

- Single Lines
- Band of Wires
- Shared Buses
- Crossbar

1. Single line (shifting, time sharing)
2. Band of Wires (BUS)

3. Shared Bus

Switches
4. Crossbar (Multiple buses running horizontally) m simultaneous transfers are possible, but more expensive.
Program:
1. Objects (Registers, Outputs of combinational logic)
2. Operation
3. Assignment
4. Sequencing

Example:  Signal R1, R2, Bit Vector V[15:0];
          Z  A + B ( A, B, Z need to be defined)
          R1  R2
          Begin
          End
          if ( ) then ( ), ENDIF;
Ex. If C then R1 \(\leftarrow\) S1
Else R2 \(\leftarrow\) S2
Endif;

\[
\begin{align*}
&\text{LD R1} \\
&C \\
&S1 \\
&S2 \\
&\text{R2}
\end{align*}
\]

If C1 then X \(\leftarrow\) A
Else X \(\leftarrow\) B + C
Endif

If C2 then G \(\leftarrow\) X
Endif

\[
\begin{align*}
&\text{Adder} \\
&A \\
&B \\
&C \\
&C1 \\
&C2 \\
&MUX \\
&G \\
&\text{CLK}
\end{align*}
\]