CSE 140 Lecture 14
System Designs
CK Cheng
CSE Dept.
UC San Diego

Note on TED.
Methodology.
System Designs

• Introduction
• Components
• Spec
• Implementation
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 times or more in hardware complexity.
• Data subsystem carries out the data operations and transports.
• Control system sequences the data subsystem and itself.

\[ n = 8, 16, 32, 64, 128 \]
I. Introduction

Data Subsystem

Control Subsystem

Data Inputs

Control Inputs

Data Outputs

Control Outputs

n=64

Conditions

Control Signals

Start/Request

Done/Acknowledgement

+, - overflow

>,
### Introduction

<table>
<thead>
<tr>
<th>Components</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Subsystem</strong></td>
<td><strong>Control Subsystem</strong></td>
</tr>
<tr>
<td>Storage Modules</td>
<td>Sequential machines</td>
</tr>
<tr>
<td>Operators</td>
<td>Control of data operations</td>
</tr>
<tr>
<td>Interconnections</td>
<td>Control of data transports</td>
</tr>
<tr>
<td></td>
<td>Control of the sequential system</td>
</tr>
</tbody>
</table>

- **Components**
  - Storage Modules
  - Operators
  - Interconnections

- **Functions**
  - Data storage
  - Data operations
  - Data transport
Data Subsystem Components

- **Storage:** Register, RAM, FIFO, LIFO, Counter, Shifter
- **Operator:** ALU, Floating Point Operators
- **Interconnect:** Wires, Buses, Crossbars
Components: Storage Modules, Register

LD: Load
CLR: Clear

Q(t+1) = (0, 0, .. , 0) if CLR = 1
= D if LD = 1 and CLR = 0
= Q(t) if LD = 0 and CLR = 0
Storage Component: Registers, Array of Registers

Registers: If C then $R \leftarrow D$

Register Array: If C then $R_{\text{address}} \leftarrow D$

Sharing connections and controls
Storage Components: RAM, FIFO, LIFO

RAM

Decoder

Address

RAM

Random Access Memory

Size of RAM larger than registers

Performance is slower

FIFO (First in first out)

Data goes from left to right

LIFO (Last in first out: Stack)
Modulo-$n$ Counter

\[ Q(t+1) = (0, 0, \ldots, 0) \]
if CLR = 1
\[ = D \]
if LD = 1 and CLR = 0
\[ = (Q(t)+1) \mod n \]
if LD = 0, CNT = 1 and CLR = 0
\[ = Q(t) \]
if LD = 0, CNT = 0 and CLR = 0

\[ TC = \begin{cases} 1 & \text{if } Q(t) = n-1 \text{ and } CNT = 1 \\ 0 & \text{otherwise} \end{cases} \]
Functional Modules

CASE Op-Sel Is
  When F1, Z <= A op1 B
  When F2, Z <= A op2 B
  .
  .
End CASE

Example:
CASE Op-Set Is
  Z <= (A + B)mod 2^n if Op-Sel=addition,
  Z <= (A - B)mod 2^n if Op-Sel=subtraction
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

MUX

DEMUX

1 2 3 .. n

R1  R2  R3  .....  Rn

R1  Rn

S

13
4. Crossbar (Multiple buses running horizontally)
m simultaneous transfers are possible, but more expensive.
Program:
1. Objects (Registers, Outputs of combinational logic)
2. Operation (Logic, Add, Multiplication, DSP, and etc.)
3. Assignment
4. Sequencing

Example:

Signal $S_1, S_2, R[15:0]$:
- FFs, Registers, wires

$Z \leftarrow A + B$:
- Registers, Adder, Interconnect

$R_1 \leftarrow R_2$:
- Registers and Interconnect

Begin, End:
- Control

if ( ) then ( ), ENDIF:
- Control
Ex. If C then R1 $\leftarrow$ S1
Else R2 $\leftarrow$ S2
Endif;

If C1 then X $\leftarrow$ A
Else X $\leftarrow$ B + C
Endif
If C2 then G $\leftarrow$ X
Endif
Implementation: Example

AddModule(X, Y, Z, start, done)
{
    Input X[15:0], Y[15:0] type bit-vector,
    start type boolean;
    Local-Object A[15:0], B[15:0] type bit-vector;
    Output Z[15:0] type bit-vector,
    done type boolean;
    S0: If start’ goto S0 || done 1;
    S1: A  X || B  Y || done 0;
    S2: Z  Add(A, B) || goto S0;
}
Exercise: Go through the handshaking, data subsystem and control subsystem designs.
AddModule(X, Y, start, done)
Hand Shaking

start

done