COAVL: A Virtual Lab on Computer Organization and Architecture

Chittaranjan Mandal
with
Gargi Roy and Devleena Ghosh
Presenter: Partha De

Dept of Computer Sc & Engg
IIT Kharagpur

COAVL Presentation
February 27, 2015
Virtual lab objective and relevance

- COA is a core course in the curriculum of CSE, EE and ECE
- Laboratory experiments essential to understanding basics
- Most places used bread board based setup

Drawbacks

1. Limits the size and extent of experiments
2. Time consuming
Virtual lab objective and relevance

- COA is a core course in the curriculum of CSE, EE and ECE
- Laboratory experiments essential to understanding basics
- Most places used bread board based setup

Drawbacks
1. Limits the size and extent of experiments
2. Time consuming
Virtual lab objective and relevance

- COA is a core course in the curriculum of CSE, EE and ECE
- Laboratory experiments essential to understanding basics
- Most places used bread board based setup

Drawbacks

1. Limits the size and extent of experiments
2. Time consuming
Virtual lab objective and relevance contd..

- FPGA based lab to overcome this limitation
- Also has some limitations:
  - Learning curve with FPGAs somewhat high
  - Expensive
  - Logistics barrier of running lab
- Hence virtual lab, especially for most institutions
Experiments designed using concept hierarchy

**Computer Arithmetic related**
- Design of a Ripple Carry Adder
- Design of a Carry-Look-Ahead Adder
- Design of Wallace Tree Adder
- Synthesis of flip-flops
- Design of Registers and Counters
- Design of Combinational Multipliers
- Design of Booth’s Multiplier
- Design of an ALU

**Memory related**
- Design of Memory elements
- Design of Associative cache without replacement policy
- Design of Direct Mapped cache without replacement policy

**CPU design related**
Design of single instruction CPU
Web interface of COLDVL

- Pedagogic considerations are reflected in web interface
- Web interface includes set of experiments, manual, others

Web interface of each experiment

- **Theory**
- **Objective**
  - Guideline to check key behavior of the design
  - Test plan
  - Assignments
- **Procedure**
- **Experimentation platform (generic simulation platform)** considered
- **Quizzes**
- **Further reading**
Features of the COAVL simulator

- The simulator supports 5-valued logic [ True(T)(wire color: blue) False(F)(wire color: black) High impedance(Z)(wire color: green) Unknown(X)(wire color: maroon) Invalid(I)(wire color: orange) ]
- Capable of simulating combinational circuits and synchronous sequential circuits
- Control signal generation from a user given ASM chart
- Bus based design with wired AND operation to CPU design
- Includes a single instruction CPU design with built-in controller
- Saving with Identification to check plagiarism
- Circuit analysis through different wire colors
- Minimal server dependency by having client side simulation
COAVL simulator interface

The simulator has the following:

**Graphical editor**
- A canvas to design the circuit
- Click and Drop method to add the components
- Different functional buttons like undo/redo, delete, zoom, save/open, print, showing timing diagram etc.

**Palette**
- Tools like connection, selection, clone
- All types of logic gates and flip-flops
- Inputs including free running clock
- Display units
- Adders, decoders, multiplexers, registers, counters etc.
- Arithmetic logic units, memory elements including cache memory
- Controller
- Other complex components like single instruction CPU, 4 bit address working memory etc.
The simulator has the following:

**Graphical editor**
- A canvas to design the circuit
- Click and Drop method to add the components
- Different functional buttons like undo/redo, delete, zoom, save/open, print, showing timing diagram etc.

**Palette**
- Tools like connection, selection, clone
- All types of logic gates and flip-flops
- Inputs including free running clock
- Display units
- Adders, decoders, multiplexers, registers, counters etc.
- Arithmetic logic units, memory elements including cache memory
- Controller
- Other complex components like single instruction CPU, 4 bit address working memory etc.
COAVL simulator interface

The simulator has the following:

**Graphical editor**
- A canvas to design the circuit
- Click and Drop method to add the components
- Different functional buttons like undo/redo, delete, zoom, save/open, print, showing timing diagram etc.

**Palette**
- Tools like connection, selection, clone
- All types of logic gates and flip-flops
- Inputs including free running clock
- Display units
  - Adders, decoders, multiplexers, registers, counters etc.
  - Arithmetic logic units, memory elements including cache memory
  - Controller
  - Other complex components like single instruction CPU, 4 bit address working memory etc.
COAVL simulator interface

The simulator has the following:

**Graphical editor**
- A canvas to design the circuit
- Click and Drop method to add the components
- Different functional buttons like undo/redo, delete, zoom, save/open, print, showing timing diagram etc.

**Palette**
- Tools like connection, selection, clone
- All types of logic gates and flip-flops
- Inputs including free running clock
- Display units
- Adders, decoders, multiplexers, registers, counters etc.
- Arithmetic logic units, memory elements including cache memory
- Controller
- Other complex components like single instruction CPU, 4 bit address working memory etc.
The simulator has the following:

**Graphical editor**
- A canvas to design the circuit
- Click and Drop method to add the components
- Different functional buttons like undo/redo, delete, zoom, save/open, print, showing timing diagram etc.

**Palette**
- Tools like connection, selection, clone
- All types of logic gates and flip-flops
- Inputs including free running clock
- Display units
- Adders, decoders, multiplexers, registers, counters etc.
- Arithmetic logic units, memory elements including cache memory
- Controller
- Other complex components like single instruction CPU, 4 bit address working memory etc.
COAVL simulator interface
Component drawers in COAVL simulator
Component drawers in COAVL simulator
Component drawers in COAVL simulator
Component drawers in COAVL simulator
Component drawers contd..
Component drawers contd..
Component drawers contd..

Palette
- Selection Tools
- Connection
- Clone Tool
- Bus Connector
- Marquee Tool

C Mandal, G Roy, D Ghosh (IIT Kharagpur)
COA VL: A Virtual Lab on Computer Organization and Architecture
Building a circuit

Components

D1  D2  D3  D4  D5

Simulate

Components

D1  D2  D3  D4  D5

C Mandal, G Roy, D Ghosh (IIT Kharagpur) COAVL: A Virtual Lab on Computer Organization
Building a circuit

Components

I1

D1

D2

D3

D4

D5

I2

Simulate

Components

I1

D1

D2

D3

D4

D5

I2
Building a circuit

C Mandal, G Roy, D Ghosh (IIT Kharagpur)
Building a circuit

Components

Simulate

Components
Building a half adder circuit

To instantiate a component left click on the component icon.
Instantiating a component contd..

Drop the component at desired position
Connecting components

left click on the connection tool
left click on the output terminal, move the mouse to the desired input terminal
Haff adder circuit
Cloning components

left click on the selection tool
Clonning components contd..

Select the desired components to be cloneed.

![Diagram showing clonning components](image)
Cloning components contd..

left click on the clone tool
Cloning components contd..

Drag from any selected component
Simulating circuits

Click on the *Simulate* button in the top toolbar
Clock waveform

1. Click the *Show waveform* button to see the waveform.
2. Red arrow indicates the button to start the clock.
3. Violet arrow indicates the button to see the component name.
Simulating sequential circuits

Red arrow indicates the button to see the pin configuration of a component.
Simulating sequential circuits contd..

Dynamic Port Circuit
In:  Out:  Set Port

Set Labels and Names
first clock  Label  Name

Memory
Load  Show

Controller
ASM Chart

Circuit Pin Config
Anticlockwise
I/P: Clk:16, S1:15, S0:14
S1 S0: 00 for no change
S1 S0: 01 for arithmetic right shift
S1 S0: 10 for parallel load
S1 S0: 11 for clear
I0: 9, I1: 10, I2: 11, I3: 12
O/P: F0: 8, F1: 7, F2: 6, F3:

Arithmetic right shift after first clock

0  S1  S0  1  0  1
1  1  0  1

F3  F2  F1  F0
Change connection type

Dynamic Port Circuit

Change connection type

C1

C3

C2
Change connection type
Change connection type
Generating control signals from a given ASM chart
Generating control signals from a given ASM chart contd..

Dynamic Port Circuit
In:   Out:   Set Port

Set Labels and Names
Label   Name

Memory
Load   Show

Controller
ASM Chart

Circuit Pin Config

Enter the name of the inputs
input1  input2

Enter the name of the outputs
out1   out2
Generating control signals from a given ASM chart contd..
Generating control signals from a given ASM chart contd..

Red arrow indicates the button to reset the controller.
Loading working memory to examine the behavior of the single instruction CPU component
Showing working memory content at any point of time

<table>
<thead>
<tr>
<th>Address</th>
<th>MSB</th>
<th>LSB</th>
<th>Address</th>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>1010001010001</td>
<td>1010001010001</td>
<td>1000</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0001</td>
<td>10000000000000</td>
<td>10000000000000</td>
<td>1001</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0010</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1010</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0011</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1011</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0100</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1100</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0101</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1101</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0110</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1110</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>0111</td>
<td>00000000000000</td>
<td>00000000000000</td>
<td>1111</td>
<td>00000000000000</td>
<td>00000000000000</td>
</tr>
</tbody>
</table>
Connecting the CPU with working memory
Saving components with identification

Click the save button

[Image of a circuit diagram with a save button being clicked and a pop-up window asking for name and roll number]
Saving components with identification contd..
Saving components with identification contd..

Click the *Show User Id* button to see identification in a saved file.
Thank you!