Announcements

- GTA: Ed Curley,
  His CEL schedule will be posted next week.
- LDV: Check FAQ page for pointers to installation instructions.
- Homework due Tuesday. Start early...

Last time

- Motivation for HDLs
  - Create the correct design on time
- VHDL:
  - Provides executable specification
  - Supports design decomposition
  - Supports mixed level simulation
  - Supports acceptance testing
- It's not software:
  - Concurrency and discrete event simulation

Introduction

Contemporary System Design
Reading: Ch. 1-2

The Abstraction Hierarchy

- Behavioral Domain
  - defined by input/output behavior
- Structural Domain
  - assembly of primitive components
- Physical Domain
  - defined by electrical and mechanical components

Behavioral Domain

If (S == ‘1’) and (R == ‘0’) Then 
Set Q to ‘1’, set QB to ‘0’
If (S == ‘1’) and (R == ‘1’) Then 
*Hold current state*
If (S == ‘0’) and (R == ‘0’) Then 
Set Q to ‘1’, set QB to ‘1’
If (S == ‘0’) and (R == ‘1’) Then 
Set Q to ‘0’, set QB to ‘1’

Structural Domain
Physical Domain

Levels of Detail

<table>
<thead>
<tr>
<th>Level</th>
<th>Behavioral Representation</th>
<th>Structural Primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Performance Spec/ENGLISH</td>
<td>Disks, racks, computers</td>
</tr>
<tr>
<td>Chip</td>
<td>Algorithmic</td>
<td>Microproc/RAM/ROM</td>
</tr>
<tr>
<td>Register</td>
<td>Data flow</td>
<td>ALU, counter</td>
</tr>
<tr>
<td>Gate</td>
<td>Boolean Eq'ns</td>
<td>AND, OR, NOT</td>
</tr>
<tr>
<td>Circuit</td>
<td>KCL/KVL/Diff.Eq.</td>
<td>NFET, PFET/RLC</td>
</tr>
<tr>
<td>Layout</td>
<td>holes/electrons</td>
<td>Poly, metal</td>
</tr>
</tbody>
</table>

Textual Representations

- Hardware Description Language
  A high-level programming language with specialized constructs for modeling hardware

Behavioral Descriptions

- Algorithmic:
  Procedure defining the I/O response is not meant to imply any particular physical implementation
- Data flow:
  Data dependencies in the description match those in a real implementation

Example: Ones Counter

- Counts the number of 1's in an input vector of length 3
- Interface description:

```vhdl
text
entity ones_counter is
  port (A: in bit_vector (2 downto 0));
  C: out bit_vector (1 downto 0));
end ones_counter;
```

Behavioral Domain

Algorithmic Architecture

```vhdl
text
architecture behavior of ones_counter is
begin
  process (A)
    variable Num: integer range 0 to 3;
  begin
    Num := 0;
    for i in 0 to 2 loop
      if A(i) = '1' then
        Num := Num + 1;
      end if;
    end loop;
    case Num is
    when 0 =>
      C <= "00";
    when 1 =>
      C <= "01";
    when 2 =>
      C <= "10";
    when 3 =>
      C <= "11";
    end case;
  end process;
end
```
Behavioral Domain

Dataflow Architecture

```
architecture dataflow of ones_counter is
begin
    C(1) <= (A(1) and A(0)) or (A(2) and A(0)) or (A(2) and A(1));
    C(0) <= (A(2) and not A(1) and not A(0))
or (not A(2) and not A(1) and A(0))
or (A(2) and A(1) and A(0))
or (not A(2) and A(1) and not A(0));
end dataflow;
```

Structural Domain

Architecture

```
architecture structural of ones_counter is
component MAJ3
    port (X: in bit_vector (2 downto 0); Z: out bit);
end component;
component OPAR3
    port (X: in bit_vector (2 downto 0); Z: out bit);
end component;
begin
    comp1: MAJ3 port map (A, C(1));
    comp2: OPAR3 port map (A, C(0));
end structural;
```

Mixed Modeling

- One can mix modeling styles in a single model
  - Structural
  - Data Flow
  - Algorithmic

Design Process

- Design:
  A series of transformations which leads to a representation that can be fabricated
- Synthesis:
  The process of transforming from one representation into another

Design Cycle

(From Preas and Lorenzetti)
Within each design step

Create a more detailed representation

From upper level

Synthesis

From lower level

Analysis

To upper level

Verification

accept

To lower level

(From Preas and Lorenzetti)

Is it fast enough? Small enough? Cheap enough? And so on.

From upper level

Synthesis

From lower level

Analysis

To upper level

Verification

reject

To lower level

(From Preas and Lorenzetti)

Do you have enough time? Enough resources? Enough people? And so on.

Design Cycle: What really happens....

Repeat n times...

English

Algorithmic

Data flow

Structural

Physical

If n is large, go out of business...

Top-down Design

- Define system behavior
  - Input/output specification
- Partition system into logical modules
  - Create specifications for modules
- Partition modules into primitives
  - Correct by Construction

Bottom-up Design

- Create primitive components
- Test and verify components
- Assemble components into higher-level modules
- Assemble modules into final system
Example: An inverter

- English:
- Algorithmic:
- Data flow:
- Structural:
- Physical:

Design Space

Area

Time

Cost

Summary

- Abstraction hierarchy
  - Behavioral, structural, physical
- Design cycle
  - Ideally a straight line
  - Actually...may be many loops
- Design space
  - Small, fast, cheap: Pick any two.
- Next time: Overview of LDV, Some VHDL Basics
- Reading: Tutorial, pp. 41-51, 72-75.