### VHDL Examples

Figures in this lecture are from:
*Rapid Prototyping of Digital Systems, Second Edition*

http://www.cs.kent.edu/~walker

---

### Processor

- **Memory Input/Ouput**
- **Data Bus**
- **Address Bus**
- **PC**
- **IR**
- **AC**
- **MDR**
- **MAR**
- **ALU Control Unit**

**Processor**

**Memory**

**Input/Output**

---

### 

**Architecture of Typical Computer System**

---

### 

**μP1 Instruction Format & Instructions**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction Mnemonic</th>
<th>Operation Performed</th>
<th>Opcode Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD address</td>
<td>AC &lt;= AC + contents of memory address</td>
<td>00</td>
</tr>
<tr>
<td>STORE address</td>
<td>contents of memory address &lt;= AC</td>
<td>01</td>
</tr>
<tr>
<td>LOAD address</td>
<td>AC &lt;= contents of memory address</td>
<td>02</td>
</tr>
<tr>
<td>JUMP address</td>
<td>PC &lt;= address</td>
<td>03</td>
</tr>
<tr>
<td>JNEG address</td>
<td>If AC &lt; 0 Then PC &lt;= address</td>
<td>04</td>
</tr>
</tbody>
</table>

**μP1 Program for “A = B + C”**

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Machine Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD B</td>
<td>0211</td>
</tr>
<tr>
<td>ADD C</td>
<td>0012</td>
</tr>
<tr>
<td>STORE A</td>
<td>0110</td>
</tr>
</tbody>
</table>

---

### MIF (Memory) File of μP1 Program

---
**μP1 Fetch, Decode, Execute Cycle**

**FETCH**
- MAR = PC
- IR = MDR
- PC = PC + 1
- Read Memory

**DECODE**
- MAR = IR
- Read Memory

**EXECUTE**
- Opcode = ADD
- Opcode = LOAD
- Opcode = STORE
- AC = AC + MDR
- AC = MDR
- MDR = AC
- Write Memory

**μP1 Datapath (Values After Reset)**

**Register Transfers in ADD’s Fetch State**
μP1 Fetch, Decode, Execute Cycle

Fetch

MAR = PC
IR = MDR
PC = PC + 1
Read Memory

Decode

MAR = IR
Read Memory

Execute

Opcode = ADD
Opcode = LOAD
Opcode = STORE

AC = AC + MDR
AC = MDR
... MDR = AC
Write Memory

Register Transfers in ADD’s Decode State

Register Transfers in ADD’s Execute State
LIBRARY ieee;
USE  ieee.std_logic_1164.ALL;
USE  ieee.std_logic_arith.ALL;

ARCHITECTURE a OF warp IS

TYPE STATE, TYPE IS ( reset, pc, fetch, decode, execute_add, execute_load, execute_store, execute_jump );

BEGIN

ONEกระบวนที่มี ของผลิตภัณฑ์สินค้าที่มีความแตกต่างในลักษณะภาพ

END a;