ECE 3055 Test 1
Wednesday, February 13
Open Book & Notes - Copy All Short Answers (1-4) to first page!

1. Part 1. 16 ns.  Part 2. 16 ns.
   Part 3. 5 ns.  Part 4. 9 ns.

2. Complexity of Control Unit
   Total Hardware Used
   1 Single Cycle  2 Single Cycle  1 Single Cycle  3 Single Cycle
   3 Multi Cycle  1 Multi Cycle  3 Multi Cycle  1 Multi Cycle
   2 Pipelined  3 Pipelined  2 Pipelined

3. Instruction = 00200000  Read Data 1 = 00000000
   Read Data 2 = 00000002  ALU Result = 0000000B
   (Data Memory) Read Data = 0 or 1  Write Register (Address) = 4
   Write Data (input at register file after mux) = 00000003
   ALU control input = 2  RegWrite = 1  ALUSrc = 0

4. Part I: Total number of NOPs required 16
   Part II: A total of 11 clock cycles is required for execution.
1. (17 points) Compare the execution time of the program segment below on the three MIPS hardware models studied in class. Assume the branch is taken.

\[
\begin{align*}
\text{lw} & \quad \text{S}3,200(0) \quad 5 \\
\text{sw} & \quad \text{S}2,100(0) \quad 4 \\
\text{or} & \quad \text{S}4,\text{S}3,\text{S}7 \quad 4 \\
\text{beq} & \quad \text{S}8,\text{S}7,\text{Label} \quad 3
\end{align*}
\]

Part 1: The single clock cycle MIPS with a clock frequency of 250 Mhz. would take 
\[4 \times 4 \times ns = 16 \times ns\]

Part 2: The multi clock cycle model of the MIPS with a clock frequency of 1Ghz. would take \[16 \times 1 \times ns = 16 \times ns\]

Part 3: The pipelined MIPS model with data forwarding and hazard detection and a clock frequency of 1 Ghz, would require \[5 \times \text{ns, stall}\] ns to execute the program. Assume a register will write and read correctly during the same clock cycle and the branch decision is made in decode. Include any stalls or flushes, but do not include the time required to initially fill the pipeline.

Part 4: If the pipelined MIPS model started this program immediately after powering up, it would take \[5 + (4 - 1) + \left(\frac{5}{\text{stall}}\right)\]

2. (8 Points) Rank the three MIPS models 1, 2, 3 in the following categories. 1 is least or smallest and 3 is most or largest. For control unit complexity only consider what is included in the one hardware block labeled “control unit” in the text. (don’t include data pipeline registers and forwarding hardware)

<table>
<thead>
<tr>
<th>Complexity of Control Unit</th>
<th>Total Hardware Used</th>
<th>CPI</th>
<th>Clock Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single Cycle</td>
<td>1</td>
<td>Single Cycle</td>
</tr>
<tr>
<td>2</td>
<td>Single Cycle</td>
<td>1</td>
<td>Single Cycle</td>
</tr>
<tr>
<td>3</td>
<td>Multi Cycle</td>
<td>1</td>
<td>Multi Cycle</td>
</tr>
<tr>
<td>2</td>
<td>Multi Cycle</td>
<td>2</td>
<td>Multi Cycle</td>
</tr>
<tr>
<td>3</td>
<td>Pipelined</td>
<td>2</td>
<td>Pipelined</td>
</tr>
<tr>
<td>3</td>
<td>Pipelined</td>
<td>2</td>
<td>Pipelined</td>
</tr>
<tr>
<td>3</td>
<td>Pipelined</td>
<td>2</td>
<td>Pipelined</td>
</tr>
</tbody>
</table>
3. (30 points) The following sequence of MIPS instructions is clocked into the pipeline shown on page 472-476. Examine this figure carefully to see exactly where each signal is located (i.e. before or after pipeline registers). After Clock cycle 5, Indicate the resulting register values in the spaces provided below. All numbers are in hex. Assume all data memory locations contain the word address of the location. Assume that each register contains a value equal to the register number prior to execution of this code.

\[\begin{align*}
\text{add} & \quad \text{S4}, \text{S3}, \text{S0} \\
\text{andi} & \quad \text{S6}, \text{S7}, \text{S8} \\
\text{add} & \quad \text{S2}, \text{S5}, \text{S6} \\
\text{lw} & \quad \text{S2}, \text{1} \\
\text{sub} & \quad \text{S3}, \text{S1}, \text{S4}
\end{align*}\]

\[\text{Instruction} = \text{8C020001}\]

\[\text{Read Data 1} = \text{0000000000}\]

\[\text{Read Data 2} = \text{00000002}\]

\[\text{ALU Result} = \text{00000000B}\]

\[\text{(Data Memory) Read Data} = \text{00000000}\]

\[\text{Write Register (Address)} = 4\]

\[\text{Write Data (input at register file after mux)} = \text{00000003}\]

\[p^{35.5}_{-5} \text{ add } = 010\]

\[\text{ALU control input} = 2\]

\[\text{RegWrite} = 1\]

\[\text{ALUSrc} = 0\]
4. (25 points) The program below is executed on the 5 stage pipelined MIPS described in chapter 6. Answer the following questions about this program.

loop:  
sw $s10($s0)  
sub $s2,$s5,$s3  
lw $s7,$200($s2)  
and $s8,$s3,$s4  
andi $s6,$s7,$s8  
bcq $s6,$s8,then  
add $s5,$s5,$s8  
then:  
or $s8,$s3,$s8  
sw $s5,$100($s6)  
beq $s8,$s0,loop

Part I (10 points) Assume the control unit does not have any hazard detection, forwarding, a new branch compare circuit, or automatic branch flushing. That register file will not write and then read a new register value in one clock cycle. Rewrite the code sequence by adding the minimum number of NOP instructions to eliminate all potential data and branch hazards—do not change the order of the instructions. Assume other non-NOP instructions follow the last branch in the original code sequence above.

Total number of NOPs required

Part II (15 points) Assume the control unit is improved by adding the hazard and forwarding unit as outlined in the text, adding a branch compare unit to the decode stage, and the register file writes then reads a new value in a single clock cycle. Determine the number of clock cycles required to complete the first loop execution (i.e., executes code in loop and branches back to top of loop and is just ready to fetch sw again) of the original code sequence. Assume the inner branch is taken.

If there were no hazards or branch flushing, the original program would require clock cycles for execution.

But the program stalls and/or flushes the pipeline clock cycles so a total of clock cycles is required for execution (do not include time to fill pipeline).
5. (20 points) Write a complete VHDL synthesis model for the digital hardware shown in the block diagram. Use a positive edge clock with a synchronous reset. Put all VHDL code inside a single Process block. The signal, Bin may or may not be required in your code.

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_SIGNED.ALL;
ENTITY testlc IS
PORT (Datain,Count : IN STD_LOGIC_VECTOR (31 DOWNTO 0));
ALU_Op : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
Control,clk,reset : IN STD_LOGIC;
OUT : OUT STD_LOGIC_VECTOR (31 DOWNTO 0));
END testlc;
ARCHITECTURE behavior OF testlc IS
SIGNAL Bin, OUT_int : STD_LOGIC_VECTOR (31 DOWNTO 0);
BEGIN
Bin <= OUT_int WHEN Control='0' ELSE Count;
OUT <= OUT_int;
PROCESS
BEGIN
WAIT UNTIL clk'EVENT AND clk='1';
IF reset='1' THEN Outp_int <= "0000000000000000000000000000000000";
ELSE
CASE ALU_op IS
WHEN "00" => Out_int <= Datain + Bin;
WHEN "01" => Out_int <= Datain - Bin;
WHEN "10" => Out_int <= Datain OR Bin;
WHEN "11" => Out_int <= Datain(30 Downto 0) & "0";
WHEN OTHERS => Out_int <= "0000000000000000000000000000000000";
END CASE;
END IF;
END PROCESS;
END behavior;