# CS365 Project #2 (25 points)

**Project Goal:** Use PathSim to observe data flow for the execution of MIPS instructions.

**Software Information:** PathSim is a simulator of the simple data path discussed in sections 4.1-4.4 of the textbook. The simulator presents a graphical depiction of the architecture shown in Figure 4.17. You need to download the software from <http://www.cs.sbu.edu/PathSim4/>. You can find the information/guidelines for using the simulator in PathSim4/SuggestedLabExercises/LabA/LabA.html. Read this file first before you start to test the following program.

**Test Program & Data**

Assembly code: (note: offset/immediate data are in hex format.)

1. lw $t0, 10($zero) // lw $r8, 10($0)
2. beq $t0, $t1, 2 //
3. add $t0, $t1, $t0 // r-type
4. sub $t1, $t1, $t0 //
5. lw $t2,48($zero) //
6. sw $t2, 80($zero) //
7. addi $t3, $zero, 80 //
8. sw $t3,3($t0) //

Data Memory Initial contents (note: both address and contents are hex numbers.)

Location Contents

00000010 00000025

00000048 00000220

00000080 00000046

Register Initial values:

R8: 00000445

R9: 00000025

**Step-by-step Instruction**

1. Load the above program to instruction memory
2. Load the above register data to register file
3. Load the above memory data to data memory
4. Step through the instruction execution cycle by cycle *until the end of execution.*
5. During each cycle of execution, view the data displayed on selected points (P1, P2, …, P10 as explained at the end of this assignment) of the datapath. If the data is on the critical path of this instruction’s execution, record the data into a worksheet (worksheet 1 below shows a sample format.) If the data isn’t on the critical path, put ‘x’ or ‘N/A’ in the worksheet.
6. Record the values of each control signals during each cycle of execution.
7. Record the values in registers (i.e. list the register contents for those registers with values at the end of execution) and values in memory locations (i.e. list memory contents for those locations with values at the end of execution). Note: no need cycle by cycle, just the final values.
8. Make sure you understand that the data displayed on the datapath as well as control signals are what you expect to have for the execution of that instruction.

Worksheet 1: cycle by cycle datapath tracing

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cycles | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
| 1 | 00000000 | 00 | 08 | 08 | 0000000A | 00000000 | 0000000A | X | 00000000 | X |
| 2 | 00000004 | 09 | 08 | X | 00000000 | 00000025 | X | X | X | 0000000010 |
| 3 | 00000008 | 09 | 08 | 08 | 00000000 | 00000025 | 00000025 | X | 00000025 | X |
| 4 | 0000000C | 09 | 08 | 09 | 00000025 | 00000025 | 00000000 | X | 00000000 | X |
| 5 | 00000010 | 00 | 0A | 0A | 00000030 | 00000000 | 00000030 | X | 00000000 | X |
| 6 | 00000014 | 00 | 0A | X | 00000050 | 00000000 | 00000050 | 00000000 | X | X |
| 7 | 00000018 | 00 | 08 | 0B | 00000050 | 00000000 | 00000050 | X | 00000050 | X |
| 8 | 0000001C | 08 | 0B | X | 00000003 | 00000025 | 00000028 | 00000050 | X | X |

Worksheet 2: Control signal values for each cycle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cycles | RegDst | MemRead | MemWrite | RegWrite | MemtoReg | ALUSrc | ALUOp | Branch |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 00 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 1 |
| 3 | 1 | 0 | 0 | 1 | 0 | 0 | 2(10) | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 0 | 2(10) | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 | 1 | 00 | 0 |
| 6 | 0 | 0 | 1 | 0 | 0 | 1 | 00 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 1 | 3(11) | 0 |
| 8 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

//do these after but don’t clear the textboxes

Worksheet 3: Register file (list register contents shown at the end of execution)

|  |  |
| --- | --- |
| Register number | Final value |
| 8 | 00000025 |
| 11 | 00000050 |
| All others | 00000000 |

Worksheet 4: Memory locations (list locations with contents shown at the end of execution)

|  |  |
| --- | --- |
| Memory address | Final value |
| 010 | 00000025 |
| 048 | 00000220 |
| 080 | 00000046 |
| 028 | 00000050 |
| 050 | 00000000 |

Note to the worksheets: You may reformat the worksheets to fit the data in. Also, the “… “ in the worksheets means that additional rows should be added in.

**Turn-in:** Printout of the above worksheets and the observation writing. No need to submit on blackboard.

**Points on datapath**

P1: input to IM’s Read address;

P2: Input to RegFile’s Read Register 1;

P3: Input to RegFile’s Read Register 2;

P4: Input to RegFile’s Write Register;

P5: First input to ALU;

P6: Second input to ALU;

P7: ALU output;

P8: Input to Data Memory’s Write Data;

P9: Input to RegFile’s Write Data;

P10: ALU result of the branch offset Adder.

(Points on datapath will be shown at lecture meeting when project the is assigned.)