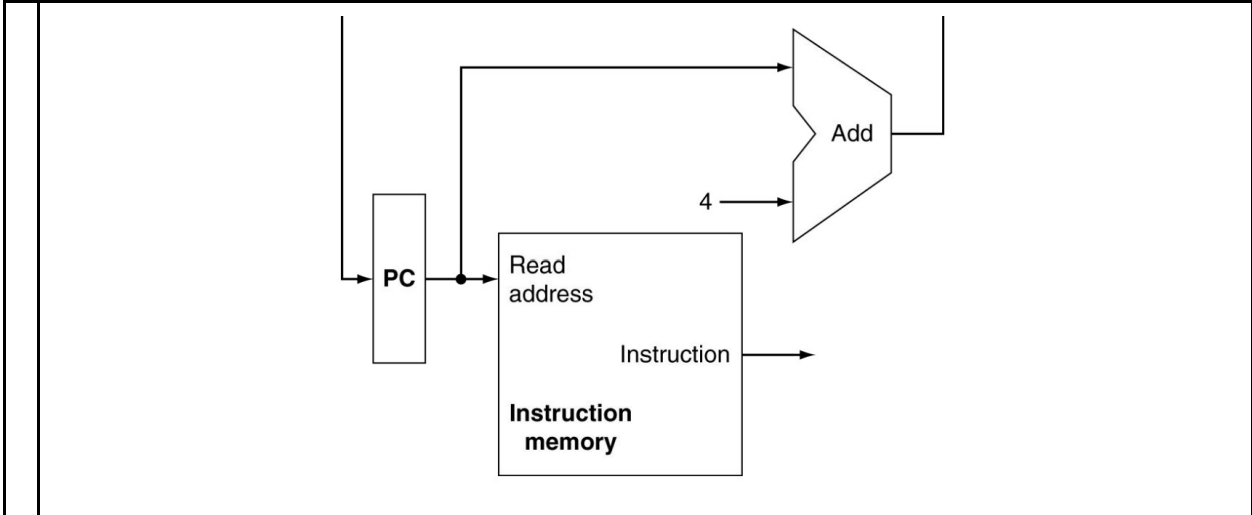


CSCI 341: Computer Organization  
WS 11: Single Cycle Data Path

<b>1</b>	<p>What two types of functional units are present in a data path? What defines when a state can change or not in a data path?</p> <p>Solution:</p> <p>Combinational and state elements. The rising clock edge triggers data write, any other time and writing is not allowed.</p>
<b>2</b>	<p>What assembly instruction is represented by the following lines of register transfer language?</p> <ol style="list-style-type: none"> <li>1. <math>R[rd] \leftarrow R[rs1] + R[rs2]</math></li> <li>2. <math>R[rd] \leftarrow \text{MEM}[R[rs1] + \text{SignExtImm}]</math></li> </ol> <p>Solution:</p> <ol style="list-style-type: none"> <li>1. add rd, rs1, rs2</li> <li>2. lw rd, Imm(rs1)</li> </ol>
<b>3</b>	<p>Why does the register file need both a Write Register and Write Data input?</p> <p>Solution:</p> <p>The register needs to know both where to write the data and what data to write.</p>
<b>4</b>	<p>Draw the datapath for the instruction fetch stage.</p> <p>Solution:</p>



5	What are the five stages in the pipeline, in order?
	<p>Solution:</p> <ol style="list-style-type: none"><li>1. Fetch, Update PC</li><li>2. Decode</li><li>3. Execute</li><li>4. Memory</li><li>5. Write Back</li></ol>

1. Fetch, Update PC
2. Decode
3. Execute
4. Memory
5. Write Back

6 Draw a picture showing the datapath (with all functional units that are used) for add t2, t1, t0  
Explain what is happening in each functional unit.

**Solution:**

The diagram illustrates the MIPS datapath for the instruction `add t2, t1, t0`. The functional units and their roles are as follows:

- PC (Program Counter):** Provides the address for Instruction Memory.
- Instruction Memory:** Provides the instruction based on the PC address.
- Registers:** Read data 1 (t1) and Read data 2 (t0) from Register 1 and Register 2 respectively. The ALU result is written back to Register 2.
- ALU (Arithmetic Logic Unit):** Performs the `Add` operation on the two register values. The ALU result is sent to the Mux.
- Mux (Multiplexer):** Selects the ALU result to be written back to Register 2.
- Data Memory:** Not used for this instruction.
- PCSrc (Program Counter Source):** Not used for this instruction.
- MemWrite (Memory Write):** Not used for this instruction.
- Mux2 (Multiplexer):** Not used for this instruction.
- IntGen (Integer Generation):** Not used for this instruction.

The datapath is annotated with blue lines and text to show the active paths for this instruction:

- ALUSrc:** Selects the register values for the ALU operation.
- Add:** The ALU operation performed.
- MemRead:** Reads the second operand from memory.
- Mux:** Selects the ALU result to be written back to Register 2.

10



(a picture without the X'd out elements shows what is used by the add)

PC is passed to adder (+4) to be updated to next instruction address.

PC is also passed to Instruction Memory to locate the add instruction.

Registers gets the read and write register inputs, and puts the two read registers (t1 and t0) on the outputs to go to the ALU. It has the write register as t2.

The ALU adds the two values, and puts it on its output, which comes around to be the value to write to the write register, which is the final action taken here.