CS4100: Computer System Design

A Simple Implementation of MIPS ISA

Madhu Mutyam
PACE Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Sept 30, 2016

A Simple Implementation of MIPS ISA

A Simple Implementation of MIPS ISA

- Instruction fetch cycle (IF)

```
IR ← Mem[PC];
NPC ← PC + 4;
```

A Simple Implementation of MIPS ISA

- Instruction fetch cycle (IF)

```
IR ← Mem[PC];
NPC ← PC + 4;
```

- Instruction decode/register fetch cycle (ID)

```
A ← Regs[rs];
B ← Regs[rt];
Imm ← Sign-extended immediate field of IR;
```

A Simple Implementation of MIPS ISA

- Instruction fetch cycle (IF)

```
IR ← Mem[PC];
NPC ← PC + 4;
```

- Instruction decode/register fetch cycle (ID)

```
A ← Regs[rs];
B ← Regs[rt];
Imm ← Sign-extended immediate field of IR;
```

- Execution/effective address cycle (EX)
  - Memory reference: `ALUOutput ← A + Imm;`
  - Register-register ALU instruction: `ALUOutput ← A` *Op* `B;`
  - Register-immediate ALU instruction: `ALUOutput ← A` *Op* `Imm;`
  - Branch instruction: `ALUOutput ← NPC + (Imm << 2);`

    ```
    Cond ← (A == 0);
    ```

A Simple Implementation of MIPS (Contd)

- Memory access/branch completion cycle (MEM)
  - `PC ← NPC;`
  - Memory reference: `LMD ← Mem[ALUOutput]` or
    `Mem[ALUOutput] ← B;`
  - Branch instruction: `if(Cond) PC ← ALUOutput;`

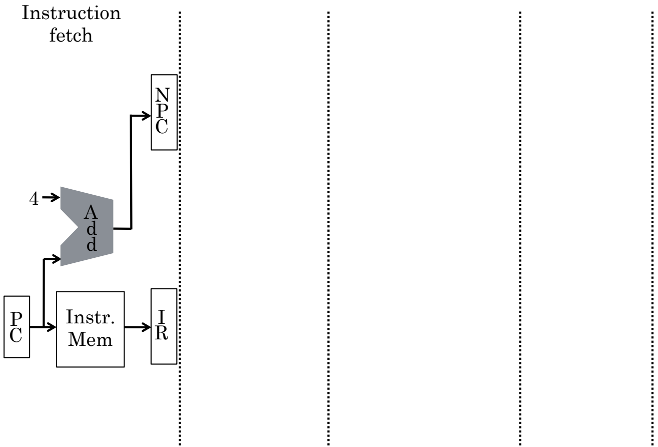
A Simple Implementation of MIPS (Contd)

- Memory access/branch completion cycle (MEM)
  - `PC ← NPC;`
  - Memory reference: `LMD ← Mem[ALUOutput]` or
    `Mem[ALUOutput] ← B;`
  - Branch instruction: `if(Cond) PC ← ALUOutput;`
- Write-back cycle (WB)
  - Register-register ALU instruction: `Regs[rd] ← ALUOutput;`
  - Register-immediate ALU instruction: `Regs[rt] ← ALUOutput;`
  - Load instruction: `Regs[rt] ← LMD;`

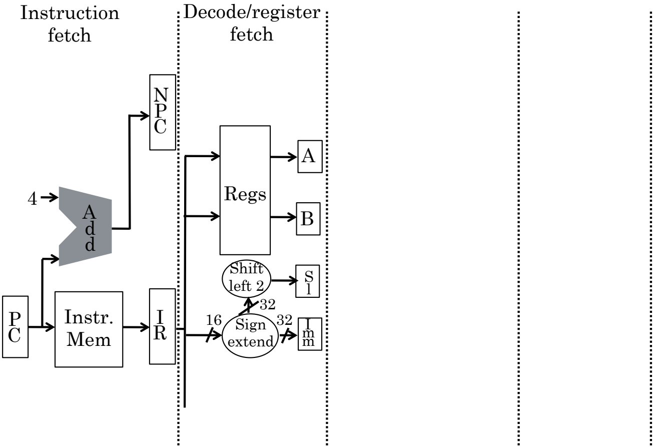
Instruction Flow in the MIPS Data Path

Instruction Flow in the MIPS Data Path

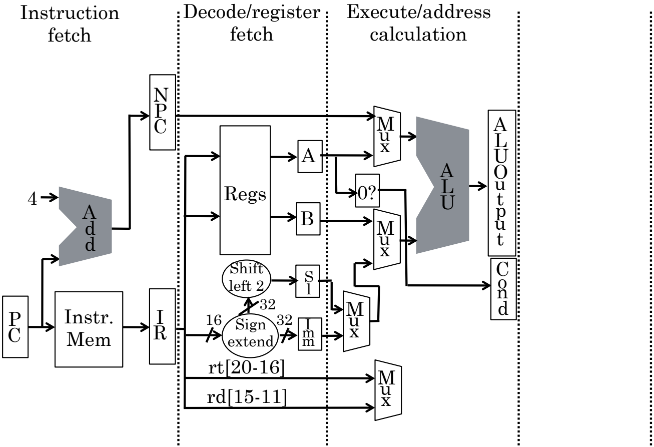
Instruction Flow in the MIPS Data Path



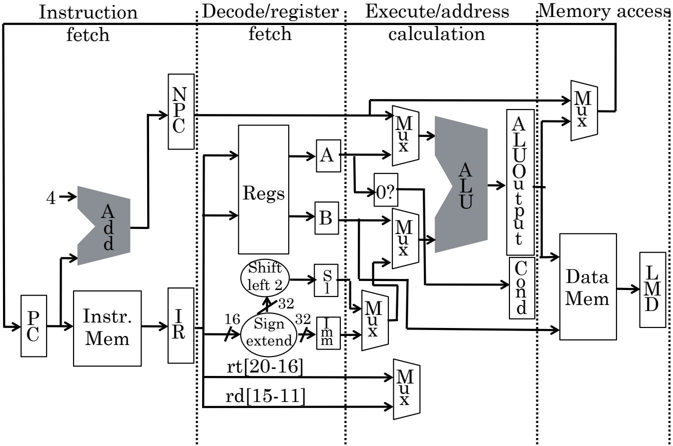
Instruction Flow in the MIPS Data Path



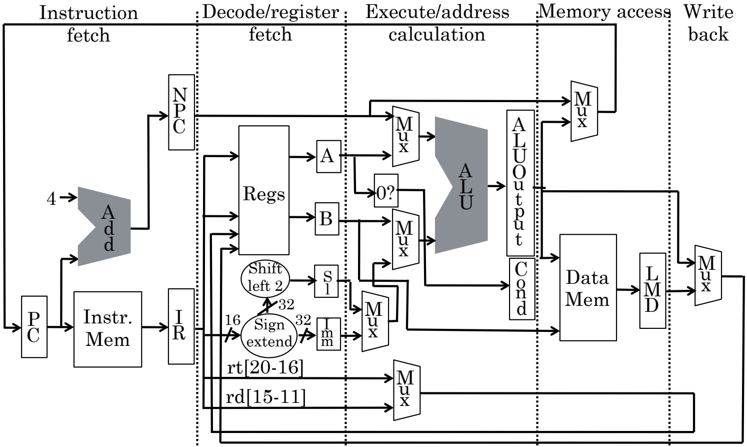
Instruction Flow in the MIPS Data Path



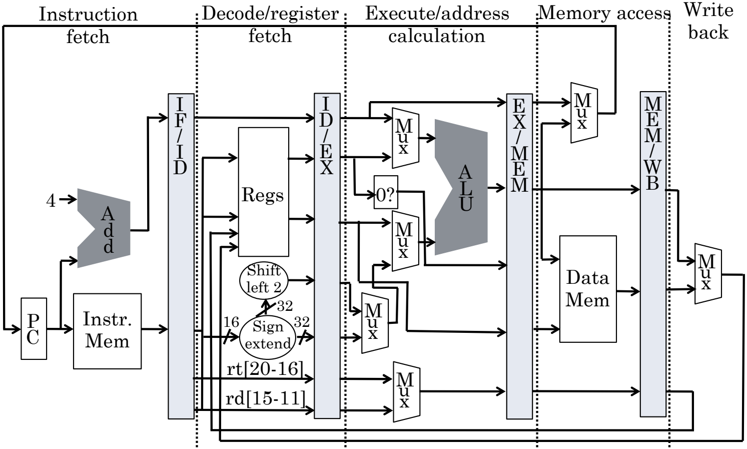
Instruction Flow in the MIPS Data Path



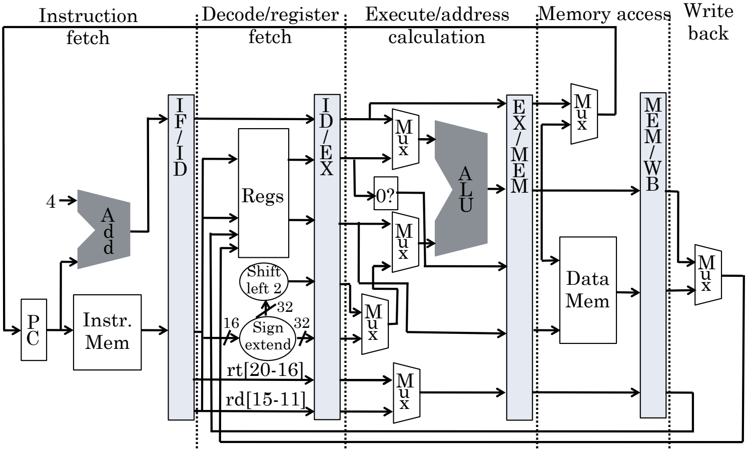
Instruction Flow in the MIPS Data Path



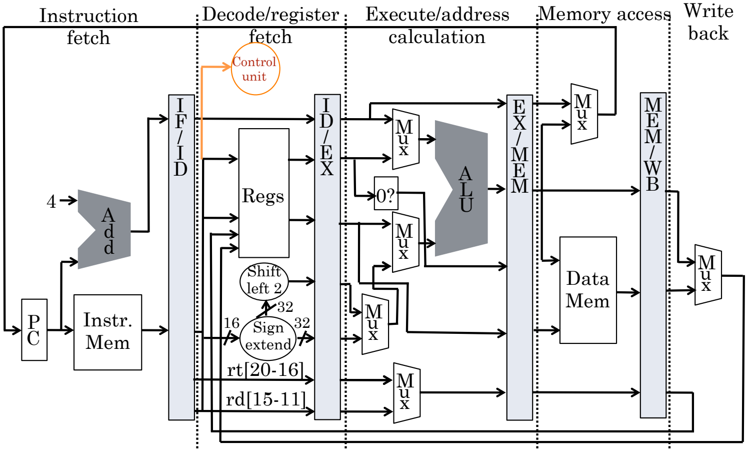
The MIPS Pipelined Data Path



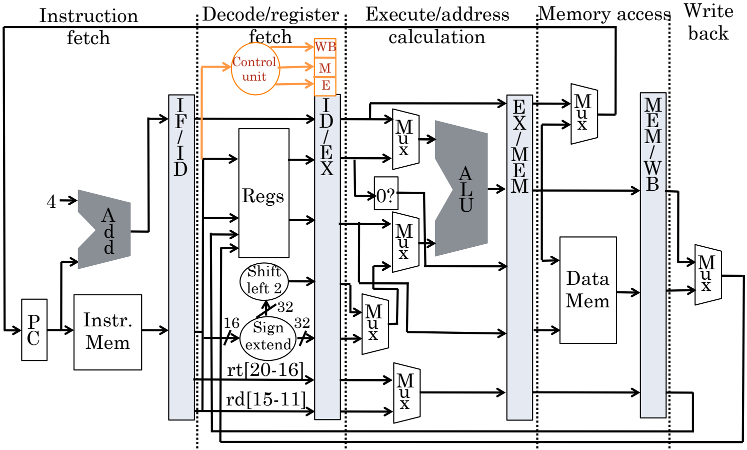
The MIPS Pipelined Data Path + Control Unit



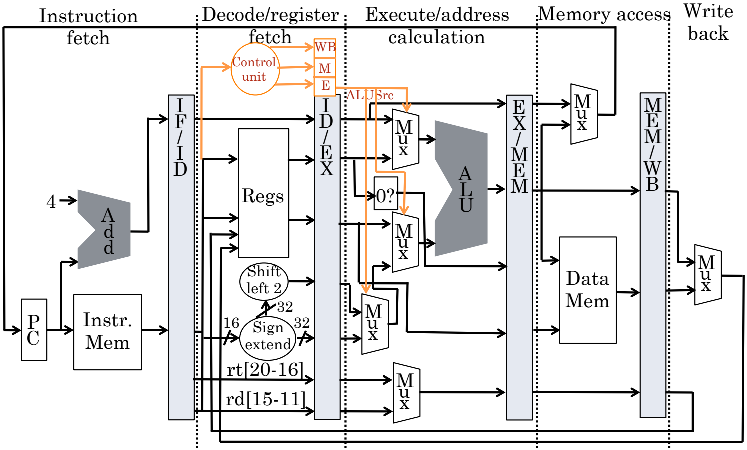
The MIPS Pipelined Data Path + Control Unit



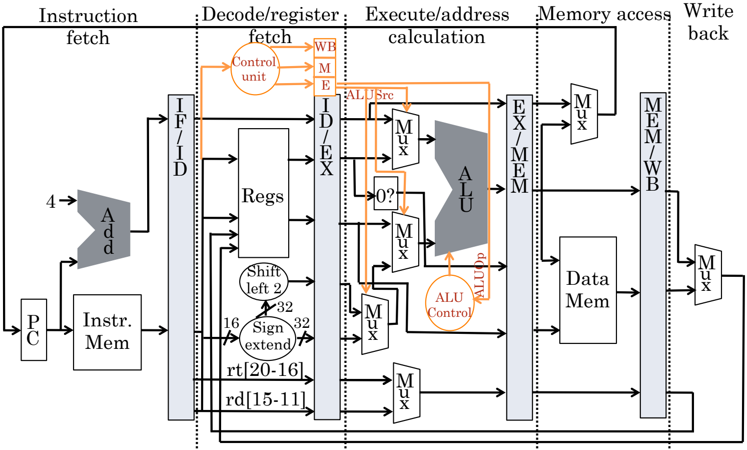
The MIPS Pipelined Data Path + Control Unit



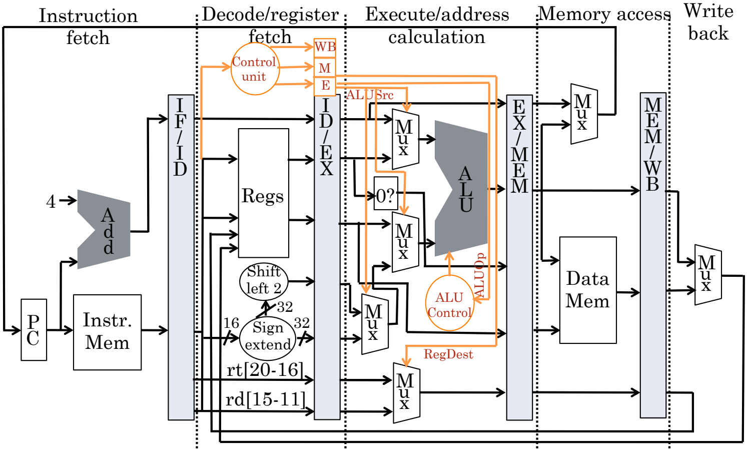
The MIPS Pipelined Data Path + Control Unit



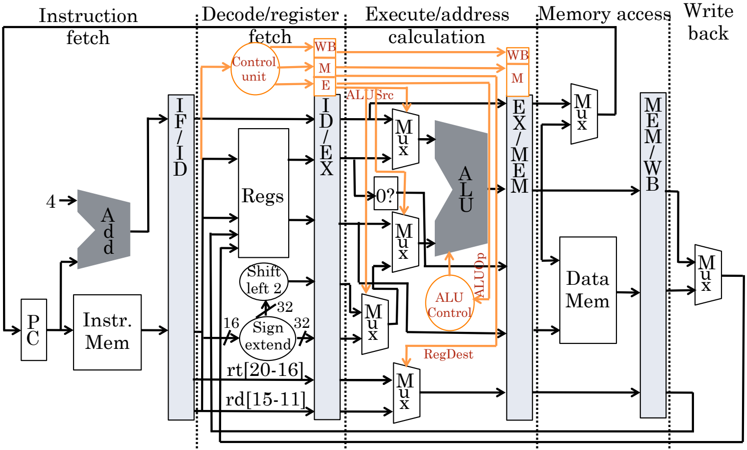
The MIPS Pipelined Data Path + Control Unit



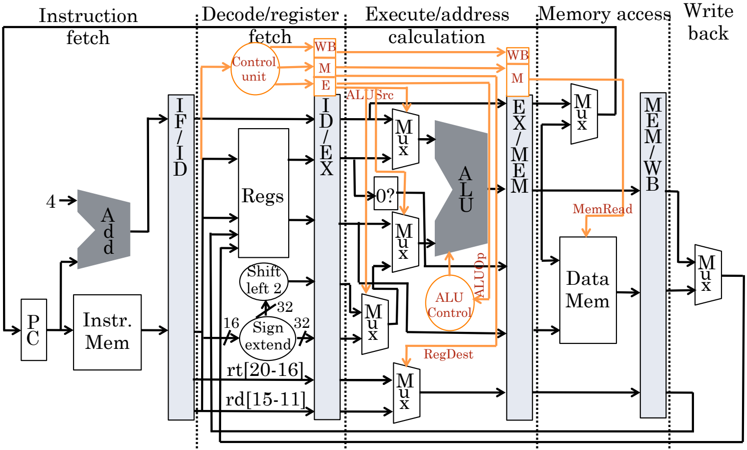
The MIPS Pipelined Data Path + Control Unit



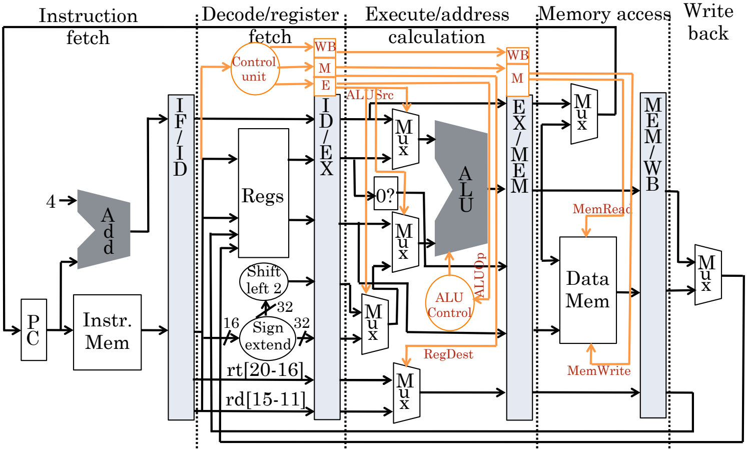
The MIPS Pipelined Data Path + Control Unit



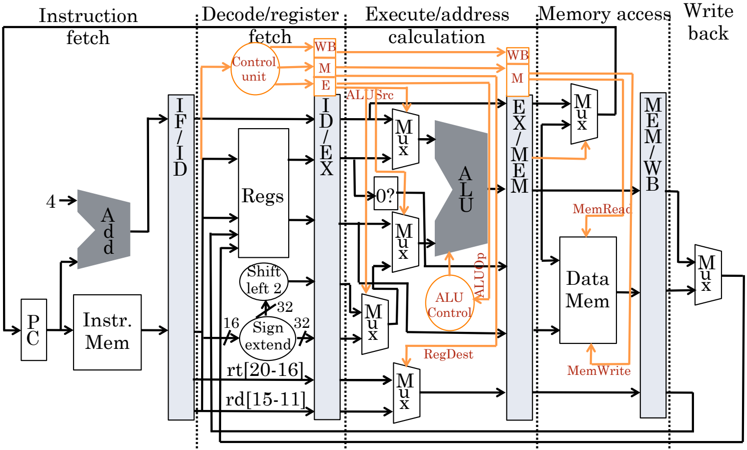
The MIPS Pipelined Data Path + Control Unit



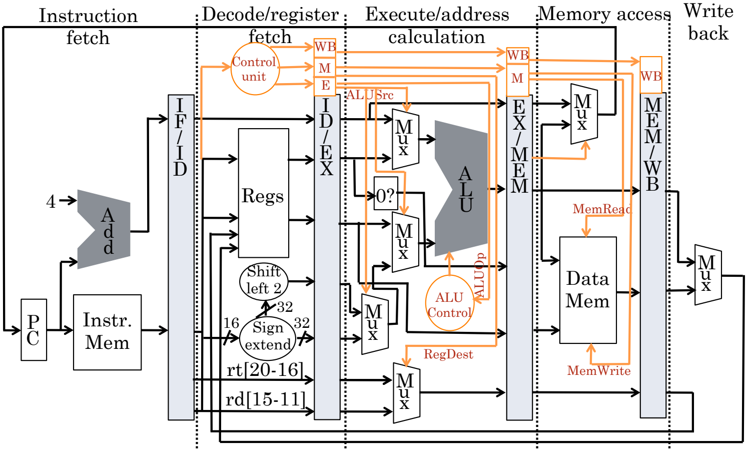
The MIPS Pipelined Data Path + Control Unit



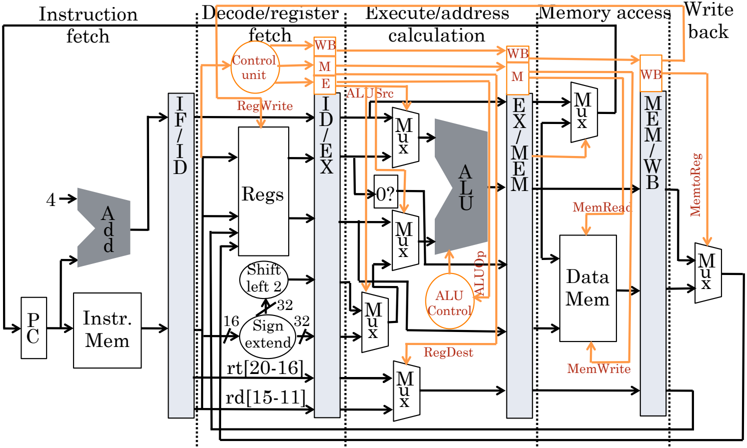
The MIPS Pipelined Data Path + Control Unit



The MIPS Pipelined Data Path + Control Unit

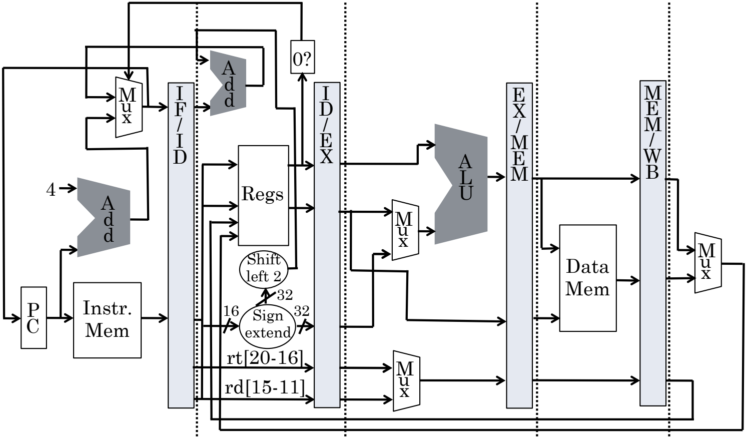


The MIPS Pipelined Data Path + Control Unit



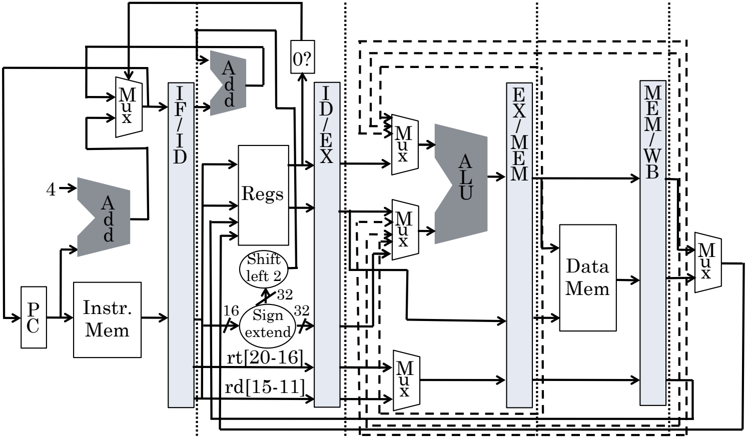
Dealing with Branches in the MIPS Pipeline

Dealing with Branches in the MIPS Pipeline



Implementing Operand Forwarding in the MIPS Pipeline

Implementing Operand Forwarding in the MIPS Pipeline



Thank You