Processor Design

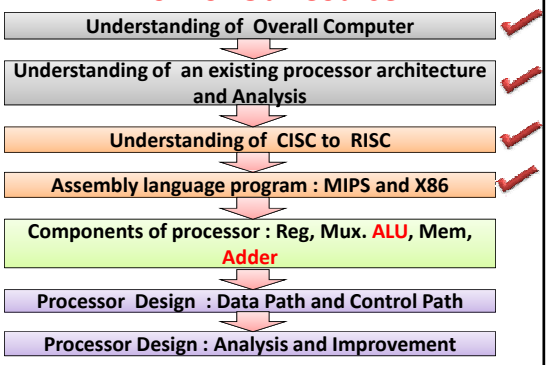
A. Sahu
CSE, IIT Guwahati

Please be updated with
http://jatinga.iitg.ernet.in/~asahu/cs222/

Outline

- **Components of CPU**
  - Register, Multiplexor, Decoder, IM/DM
  - Adder, substractor, **Varity of Adder**
  - **Multiplier : Serial, Parallel**
  - **ALU and Floating Point**
- **Processor Design**
  - Single Cycle (Data Path and Control)
  - Multi-cycle (Data Path)
  - Pipeline (Data Path)

Flow of Our Course



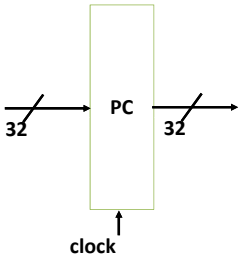
Processor Design & ALU Design

- **Topic ALU Design ← → Processor Design**
- ALU Design
  - Adder/ Substrator, Multiplier
  - Floating point, ALU (Int/float) Design
- Processor Design
  - Single Cycle (Data Path & Control Path)
  - Multi Cycle (Only Data Path)
  - Pipelined (Only Data path)

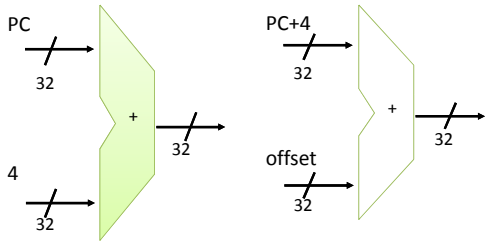
Components for MIPS subset

- Register
- Adder
- ALU
- Multiplexer
- Register file
- Program memory
- Data memory
- Bit manipulation components

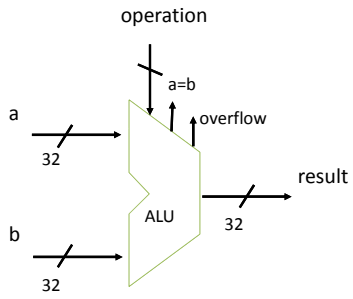
MIPS Components - Register



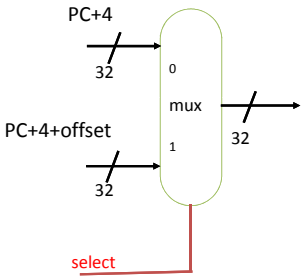
MIPS Components - Adder



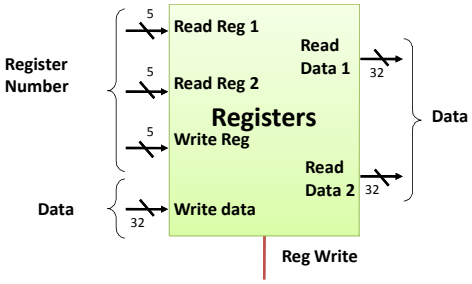
MIPS Components - ALU



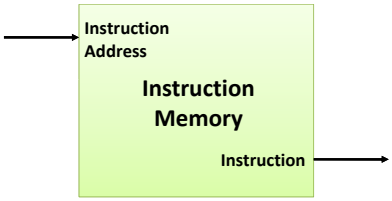
MIPS components - Multiplexers



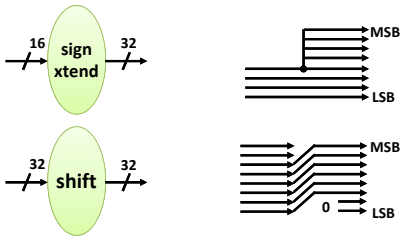
MIPS Components - register file



MIPS Components: Program memory



MIPS Components - Bit manipulation circuits



Processor Design

13

Processor Design

- **A simple implementation: Single Cycle**
  - **Data path and control**
- Performance considerations
- Multi-cycle design
  - Data path and control
- Micro-programmed control
- Exception handling

14

Simple Processor Design

- MIPS subset for implementation
- Design overview
- Division into data path and control
- Building blocks - combinational and sequential
- Clock and timings
- Components required for MIPS subset

MIPS subset for implementation

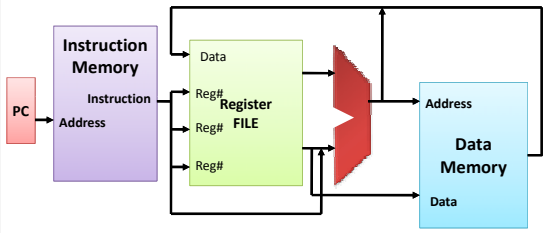
- Arithmetic - logic instructions
  - **–add, sub, and, or, slt**
- Memory reference instructions
  - **–lw, sw**
- Control flow instructions
  - **–beq, j**

Incremental changes in the design to include other instructions will be discussed later

Generic Implementation

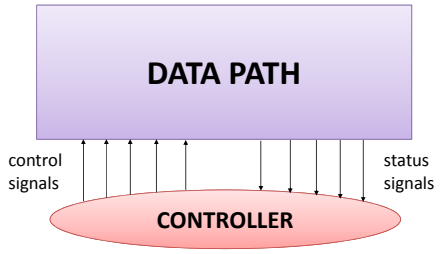
- Use the program counter (PC) to supply instruction address
- Get the instruction from memory
- Read registers
- Use the instruction to decide exactly what to do

Design overview



18

Division into Data path and Control



A Processor Design Method

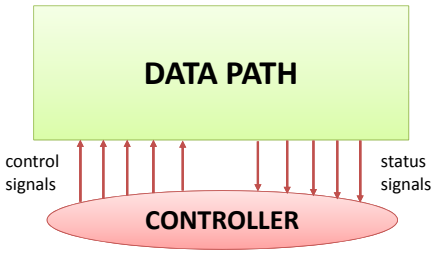
Build the datapath step by step as follows

- Start with R - class instructions
- Include other instructions one by one
- Identify control signals
- Interconnect datapath and controller

MIPS subset for implementation

- Arithmetic - logic instructions
  - **add, sub, and, or, slt**
- Memory reference instructions
  - **lw, sw**
- Control flow instructions
  - **beq, j**

Division into data path and control



Datapath for add,sub,and,or,slt

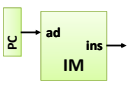
- fetch instruction
- address the register file
- pass operands to ALU
- pass result to register file
- increment PC

} actions required

Format: add $t0, $s1, $s2

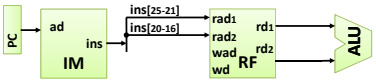
000000	10001	10010	01000	00000	100000
op	r_{dst}	r_{src1}	r_{src2}	shamt	funct

Fetching instruction

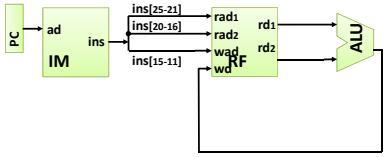


Addressing RF

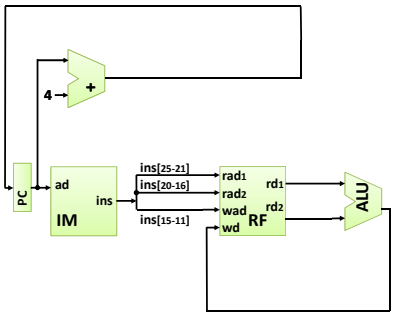
Passing operands to ALU



Passing the result to RF



Incrementing PC

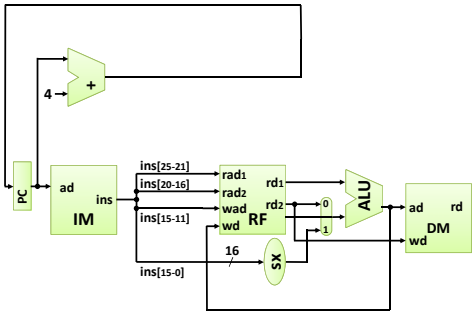


Load and Store instructions

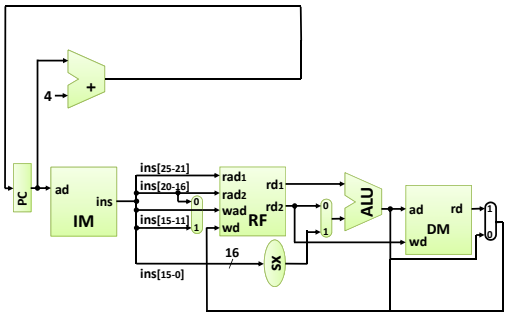
- format : I
- Example: lw $t0, 32($s2)

35	18	9	32
op	r_{dst}	r_{src1}	16 bit number

Adding "sw" instruction



Adding "lw" instruction



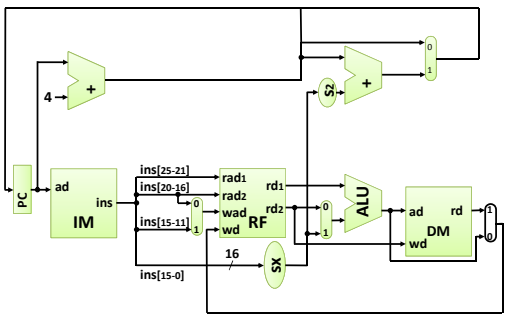
Format of beq instruction

- beq I – format

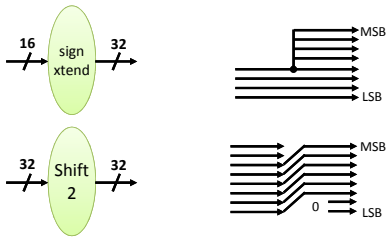
op	rs	rt	16 bit number

PC = (PC+4) + SignXtend16to32 (level*2)

Adding "beq" instruction



MIPS components - bit manipulation circuits



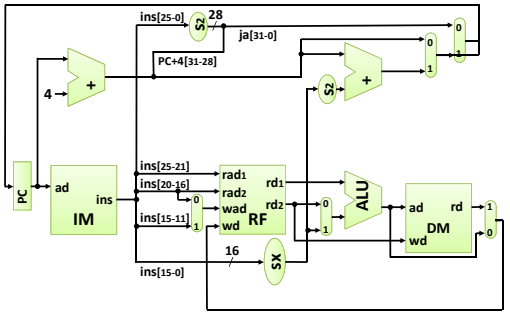
Format of jump instruction

- j J - format

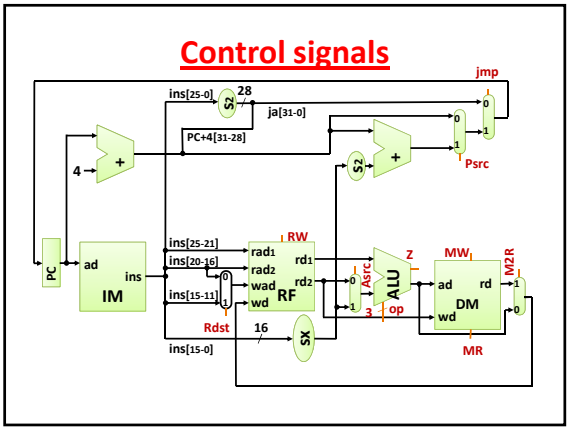
op	26 bit number

PC = Higher 4bit PC Old PC + Level <<2
= 4 bit from PC + 16 bit Level + 2 bit 00
= 32 bits

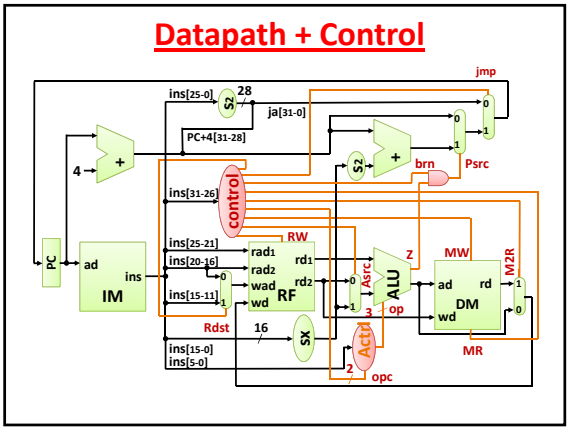
Adding "j" instruction



Control signals



Datapath + Control



Summary

Processor designed for {add, sub, and, or, slt, lw, sw, beq, j}

- Step by step approach
- Started with {add, sub, and, or, slt}
- Added {sw, lw}, then added {beq, j}
- Identified control signals and connected to a controller (black box).