Instruction Pipelining and Hazards

Daniel Sanchez
Computer Science and Artificial Intelligence Laboratory
M.I.T.
Reminder: Harvard-Style Single-Cycle Datapath for MIPS
Princeton challenge

What problem arises if we use a single memory to hold instructions and data?
What problem arises if we use a single memory to hold instructions and data?

At least the instruction fetch and a Load (or Store) cannot be executed in the same cycle.
Princeton challenge

What problem arises if we use a single memory to hold instructions and data?

At least the instruction fetch and a Load (or Store) cannot be executed in the same cycle

Structural hazard
Princeton Microarchitecture
Datapath & Control

Fetch phase

- IR
- OpCode
- RegDst
- ExtSel
- OpSel
- BSrc
- zero?
- AddrSrc

- PCen
- 0x4
- Add
- RegWrite
- MemWrite
- WBSrc

- ALU
- ALU Control
- Memory
- Data
- wdata
- we
- addr
- rdata
- clk
- rs1
- rs2
- rd1
- ws
- wd
- rd2
- GPRs
- Imm
- Ext
- clk
- Add
- PC
- IR
Princeton Microarchitecture

Datapath & Control

Fetch phase

IRen  OpCode  RegDst  ExtSel  OpSel  BSrc  zero?  AddrSrc = PC

February 25, 2019

MIT 6.823 Spring 2019
Two-State Controller
Princeton Architecture

fetch phase

execute phase

A flip-flop can be used to remember the phase
Control Logic
Princeton Architecture

IR
- opcode
- zero?

old combinational logic (Harvard)
- ExtSel, BSrc, OpSel, WBSrc, RegDest, PCsrc1, PCsrc2
- MemWrite
- RegWrite

1-bit Toggle FF
 Fetch / Execute

new combinational logic
- Wen
- PCen
- IRe
- AddrSrc
The same (mux not shown)

Only one of the phases is active in any cycle
⇒ a lot of datapath is not in use at any given time
Princeton Microarchitecture
Overlapped execution

fetch phase

execute phase
Princeton Microarchitecture
Overlapped execution

Can we overlap instruction fetch and execute?
Can we overlap instruction fetch and execute?

Yes, unless IR contains a Load or Store
Can we overlap instruction fetch and execute?

Yes, unless IR contains a Load or Store

Which action should be prioritized?
Can we overlap instruction fetch and execute?

Yes, unless IR contains a Load or Store

Which action should be prioritized? Execute
Princeton Microarchitecture

Overlapped execution

Can we overlap instruction fetch and execute?

Yes, unless IR contains a Load or Store

Which action should be prioritized? Execute

What do we do with Fetch?
Can we overlap instruction fetch and execute?

Yes, unless IR contains a Load or Store

Which action should be prioritized? Execute

What do we do with Fetch? Stall it
Can we overlap instruction fetch and execute?  
Yes, unless IR contains a Load or Store

Which action should be prioritized?  
Execute

What do we do with Fetch?  
Stall it  
How?
Stalling the instruction fetch

Princeton Microarchitecture

fetch phase

execute phase
When stall condition is indicated
Stalling the instruction fetch

Princeton Microarchitecture

When stall condition is indicated
  - don’t fetch a new instruction and don’t change the PC
Stalling the instruction fetch

*Princeton Microarchitecture*

When stall condition is indicated
- *don’t fetch a new instruction and don’t change the PC*
When stall condition is indicated
- don’t fetch a new instruction and don’t change the PC
- insert a nop in the IR
When stall condition is indicated
- don’t fetch a new instruction and don’t change the PC
- insert a nop in the IR
- set the Memory Address mux to ALU (not shown)
When stall condition is indicated:

- don’t fetch a new instruction and don’t change the PC
- insert a nop in the IR
- set the Memory Address mux to ALU (not shown)

What if IR contains a jump or branch instruction?
Need to stall on branches

Princeton Microarchitecture

fetch phase

execute phase

Jump?
Need to stall on branches

Princeton Microarchitecture

When IR contains a jump or taken branch

• no "structural conflict" for the memory
Need to stall on branches

Princeton Microarchitecture

When IR contains a jump or taken branch

- no "structural conflict" for the memory
- but we do not have the correct PC value in the PC
Need to stall on branches

Princeton Microarchitecture

When IR contains a jump or taken branch

- no "structural conflict" for the memory
- but we do not have the correct PC value in the PC
- memory cannot be used – Address Mux setting is irrelevant
Need to stall on branches

**Princeton Microarchitecture**

When IR contains a jump or taken branch

- *no “structural conflict” for the memory*
- *but we do not have the correct PC value in the PC*
- *memory cannot be used – Address Mux setting is irrelevant*
- *insert a nop in the IR*
When IR contains a jump or taken branch
- no “structural conflict” for the memory
- but we do not have the correct PC value in the PC
- memory cannot be used – Address Mux setting is irrelevant
- insert a nop in the IR
- insert the nextPC (branch-target) address in the PC
When IR contains a jump or taken branch
  • *no “structural conflict” for the memory*
  • *but we do not have the correct PC value in the PC*
  • *memory cannot be used – Address Mux setting is irrelevant*
  • *insert a nop in the IR*
  • *insert the nextPC (branch-target) address in the PC*
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUiu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ(_z=1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ(_z=0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td></td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUi</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUiu</td>
<td></td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=1&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=0&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAddrSrc = pc/ALU
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ₂=1</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ₂=0</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAddSrc = pc/ALU
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
### Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td></td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=1&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=0&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td></td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td></td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td></td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_z=1</td>
<td></td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_z=0</td>
<td></td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td></td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ (_z=1)</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ (_z=0)</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
### Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=1&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=0&lt;/sub&gt;</td>
<td></td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm    ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAAddr Src = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAdd Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>sE^16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>uE^16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE^16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE^16</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE^16</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_{z=1}</td>
<td>sE^16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_{z=0}</td>
<td>sE^16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAaddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=1&lt;/sub&gt;</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=0&lt;/sub&gt;</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAaddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;_z=1&lt;/sub&gt;</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>* 0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;_z=0&lt;/sub&gt;</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>* 0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAaddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ₂=1</td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ₂=0</td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm    ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAaddrSrc = pc/ALU
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAaddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ_z=1</td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_z=0</td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAaddrSrc = pc/ALU  
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ₁=1</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ₀=0</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAddrSrc = pc/ALU  
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAaddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no    *</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
<td></td>
</tr>
<tr>
<td>BEQZ₂=1</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no    *</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ₂=0</td>
<td></td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no    *</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no    *</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>no    *</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no    *</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
<td></td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAaddrSrc = pc/ALU
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE_16</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE_16</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ_z=1</td>
<td>yes</td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ_z=0</td>
<td>yes</td>
<td>sE_16</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC;  IRSrc = nop/mem;  MAddrSrc = pc/ALU
RegDst = rt / rd / R31;  PCSrc1 = pc+4 / br / rind / jabs;  PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE\textsubscript{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ\textsubscript{z=1}</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQZ\textsubscript{z=0}</td>
<td>sE\textsubscript{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;_z=1&lt;/sub&gt;</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;_z=0&lt;/sub&gt;</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
### Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE\textsubscript{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ\textsubscript{z=1}</td>
<td>yes</td>
<td>sE\textsubscript{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>BEQZ\textsubscript{z=0}</td>
<td>sE\textsubscript{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JALR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>ExtSel</th>
<th>BSrc</th>
<th>OpSel</th>
<th>MemW</th>
<th>RegW</th>
<th>WBSrc</th>
<th>RegDst</th>
<th>PCSrc1</th>
<th>PCSrc2</th>
<th>IRSrc</th>
<th>MAddrSrc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE₁₆</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE₁₆</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ₂=1</td>
<td>yes</td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>BEQZ₂=0</td>
<td>yes</td>
<td>sE₁₆</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>J</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JAL</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JALR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddrSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
## Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE_{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE_{16}</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE_{16}</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE_{16}</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ_{z=1}</td>
<td>yes</td>
<td>sE_{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>BEQZ_{z=0}</td>
<td>no</td>
<td>sE_{16}</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>J</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JAL</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JALR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
### Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=1&lt;/sub&gt;</td>
<td>yes</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>BEQZ&lt;sub&gt;z=0&lt;/sub&gt;</td>
<td>no</td>
<td>sE&lt;sub&gt;16&lt;/sub&gt;</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>ncp</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>J</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td>ncp</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JAL</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td>ncp</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td>ncp</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JALR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td>ncp</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>ncp</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm ; WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC
# Pipelined Princeton: Control Table

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Stall</th>
<th>Ext Sel</th>
<th>B Src</th>
<th>Op Sel</th>
<th>Mem W</th>
<th>Reg W</th>
<th>WB Src</th>
<th>Reg Dst</th>
<th>PC Src1</th>
<th>PC Src2</th>
<th>IR Src</th>
<th>MAddr Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>no</td>
<td>*</td>
<td>Reg</td>
<td>Func</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rd</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUi</td>
<td>no</td>
<td>sE(_{16})</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>ALUiu</td>
<td>no</td>
<td>uE(_{16})</td>
<td>Imm</td>
<td>Op</td>
<td>no</td>
<td>yes</td>
<td>ALU</td>
<td>rt</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>LW</td>
<td>yes</td>
<td>sE(_{16})</td>
<td>Imm</td>
<td>+</td>
<td>no</td>
<td>yes</td>
<td>Mem</td>
<td>rt</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>SW</td>
<td>yes</td>
<td>sE(_{16})</td>
<td>Imm</td>
<td>+</td>
<td>yes</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>pc</td>
<td>nop</td>
<td>ALU</td>
</tr>
<tr>
<td>BEQZ(_{z=1})</td>
<td>yes</td>
<td>sE(_{16})</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>br</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>BEQZ(_{z=0})</td>
<td>no</td>
<td>sE(_{16})</td>
<td>*</td>
<td>0?</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
<tr>
<td>J</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JAL</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>jabs</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>JALR</td>
<td>yes</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>yes</td>
<td>PC</td>
<td>R31</td>
<td>rind</td>
<td>npc</td>
<td>nop</td>
<td>*</td>
</tr>
<tr>
<td>NOP</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>no</td>
<td>*</td>
<td>*</td>
<td>pc+4</td>
<td>npc</td>
<td>mem</td>
<td>pc</td>
</tr>
</tbody>
</table>

BSrc = Reg / Imm  ;  WBSrc = ALU / Mem / PC; IRSrc = nop/mem; MAddSrc = pc/ALU
RegDst = rt / rd / R31; PCSrc1 = pc+4 / br / rind / jabs; PCSrc2 = pc/nPC

stall & IRSrc columns are identical
Pipelined Princeton Architecture

**Clock:** \[ t_{C-Princeton} > t_{RF} + t_{ALU} + t_{M} + t_{WB} \]

**CPI:** \((1 - f) + 2f\) cycles per instruction where \(f\) is the fraction of instructions that cause a stall.
Pipelined Princeton Architecture

**Clock:** \( t_{C-Princeton} > t_{RF} + t_{ALU} + t_M + t_{WB} \)

**CPI:** \( (1 - f) + 2f \) cycles per instruction

where \( f \) is the fraction of instructions that cause a stall

*What is a likely value of \( f \)?*
An Ideal Pipeline

- All objects go through the same stages
- No sharing of resources between any two stages
- Propagation delay through all pipeline stages is equal
- The scheduling of an object entering the pipeline is not affected by the objects in other stages

These conditions generally hold for industrial assembly lines.

But what about an instruction pipeline?
Pipelined Datapath

```
0x4 Add

addr rdata
Inst. Memory

V we
rs1 rs2
rd1
ws wd rd2
GPRs

Imm Ext

ALU

V we
addr
Data
Memory
rdata
wdata
```
Pipelined Datapath

- **PC**: Program Counter
- **Addr**: Address
- **Rdata**: Result Data
- **IR**: Instruction Register
- **Mem**: Memory
- **ALU**: Arithmetic Logic Unit
- **GPRs**: General Purpose Registers
- **Imm**: Immediate
- **Ext**: Extended
- **ALU**: ALU output
- **Mem**: Memory output
- **Rdata**: Result Data output
Pipelined Datapath

fetch phase

decode & Reg-fetch phase

execute phase

memory phase

write-back phase
Pipelined Datapath

Clock period can be reduced by dividing the execution of an instruction into multiple cycles

\[ t_C > \max \{ t_{IM}, t_{RF}, t_{ALU}, t_{DM}, t_{RW} \} \ (= t_{DM} \text{ probably}) \]

However, CPI will increase unless instructions are pipelined
Suppose memory is significantly slower than other stages. For example, suppose

\[ t_{IM} = 10 \text{ units} \]
\[ t_{DM} = 10 \text{ units} \]
\[ t_{ALU} = 5 \text{ units} \]
\[ t_{RF} = 1 \text{ unit} \]
\[ t_{RW} = 1 \text{ unit} \]

Since the slowest stage determines the clock, it may be possible to combine some stages without any loss of performance.
Alternative Pipelining

\[ t_C > \max \{ t_{IM}, t_{RF}, t_{ALU}, t_{DM}, t_{RW} \} = t_{DM} \]
Alternative Pipelining

\[ t_C > \max \{ t_{\text{IM}}, t_{\text{RF}}, t_{\text{ALU}}, t_{\text{DM}}, t_{\text{RW}} \} = t_{\text{DM}} \]
Alternative Pipelining

\[ t_C > \max \{ t_{IM}, t_{RF} + t_{ALU}, t_{DM}, t_{RW} \} = t_{DM} \]
Alternative Pipelining

Write-back stage takes much less time than other stages. Suppose we combined it with the memory phase.
Alternative Pipelining

Write-back stage takes much less time than other stages. Suppose we combined it with the memory phase.
Alternative Pipelining

Write-back stage takes much less time than other stages. Suppose we combined it with the memory phase
Alternative Pipelining

\[ t_C > \max \{ t_{IM}, t_{RF} + t_{ALU}, t_{DM} + t_{RW} \} = t_{DM} + t_{RW} \]

⇒ *increase the critical path by 10%*

Write-back stage takes much less time than other stages. Suppose we combined it with the memory phase
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
</table>

February 25, 2019
# Maximum Speedup by Pipelining

## Assumptions

1. $t_{IM} = t_{DM} = 10$,  
   $t_{ALU} = 5$,  
   $t_{RF} = t_{RW} = 1$  
4-stage pipeline

<table>
<thead>
<tr>
<th></th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

February 25, 2019
Maximum Speedup by Pipelining

Assumptions

1. $t_{IM} = t_{DM} = 10,$
   $t_{ALU} = 5,$
   $t_{RF} = t_{RW} = 1$
4-stage pipeline

<table>
<thead>
<tr>
<th>Unpipelined</th>
<th>Pipelined Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_C$</td>
<td>$t_C$</td>
</tr>
</tbody>
</table>

27
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
## Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
</tbody>
</table>
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$ 4-stage pipeline</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$ 4-stage pipeline</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

February 25, 2019

MIT 6.823 Spring 2019
### Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined ( t_C )</th>
<th>Pipelined ( t_C )</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( t_{IM} = t_{DM} = 10, )</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>( t_{ALU} = 5, )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{RF} = t_{RW} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-stage pipeline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. ( t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5 )</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-stage pipeline</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10, \ t_{ALU} = 5, \ t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

4-stage pipeline
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
</tbody>
</table>
# Maximum Speedup by Pipelining

## Assumptions

<table>
<thead>
<tr>
<th></th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$t_{\text{IM}} = t_{\text{DM}} = 10$, $t_{\text{ALU}} = 5$, $t_{\text{RF}} = t_{\text{RW}} = 1$</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>2.</td>
<td>$t_{\text{IM}} = t_{\text{DM}} = t_{\text{ALU}} = t_{\text{RF}} = t_{\text{RW}} = 5$</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>3.</td>
<td>$t_{\text{IM}} = t_{\text{DM}} = t_{\text{ALU}} = t_{\text{RF}} = t_{\text{RW}} = 5$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4-stage pipeline

5-stage pipeline
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined</th>
<th>Pipelined</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>3. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined(t_C)</th>
<th>Pipelined(t_C)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (t_{IM} = t_{DM} = 10, t_{ALU} = 5, t_{RF} = t_{RW} = 1) 4-stage pipeline</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. (t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5) 4-stage pipeline</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>3. (t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5) 5-stage pipeline</td>
<td>25</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined t&lt;sub&gt;C&lt;/sub&gt;</th>
<th>Pipelined t&lt;sub&gt;C&lt;/sub&gt;</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( t_{IM} = t_{DM} = 10, ) ( t_{ALU} = 5, ) ( t_{RF} = t_{RW} = 1 ) 4-stage pipeline</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. ( t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5 ) 4-stage pipeline</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>3. ( t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5 ) 5-stage pipeline</td>
<td>25</td>
<td>5</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10,$ $t_{ALU} = 5,$ $t_{RF} = t_{RW} = 1$ 4-stage pipeline</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$ 4-stage pipeline</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>3. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$ 5-stage pipeline</td>
<td>25</td>
<td>5</td>
<td>5.0</td>
</tr>
</tbody>
</table>

What seems to be the message here?
# Maximum Speedup by Pipelining

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Unpipelined $t_C$</th>
<th>Pipelined $t_C$</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $t_{IM} = t_{DM} = 10$, $t_{ALU} = 5$, $t_{RF} = t_{RW} = 1$</td>
<td>27</td>
<td>10</td>
<td>2.7</td>
</tr>
<tr>
<td>2. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>3. $t_{IM} = t_{DM} = t_{ALU} = t_{RF} = t_{RW} = 5$</td>
<td>25</td>
<td>5</td>
<td>5.0</td>
</tr>
</tbody>
</table>

What seems to be the message here?

One can achieve higher speedup with more pipeline stages
5-Stage Pipelined Execution

Instruction Flow Diagram

I-Fetch (IF)

Decode, Reg. Fetch (ID)

Execute (EX)

Memory (MA)

Write-Back (WB)
5-Stage Pipelined Execution

Instruction Flow Diagram
5-Stage Pipelined Execution

Instruction Flow Diagram

- **I-Fetch (IF)**
  - time: instruction1

- **Decode, Reg. Fetch (ID)**
  - t0: IF1
  - t1: ID1

- **Execute (EX)**
  - t2: EX1
  - t3: MA1
  - t4: WB1

- **Memory (MA)**
  - t5

- **Write-Back (WB)**
  - t6
  - t7
  - . . .
5-Stage Pipelined Execution

Instruction Flow Diagram

I-Fetch (IF)

Decode, Reg. Fetch (ID)

Execute (EX)

Memory (MA)

Write-Back (WB)

\[ \text{time} \quad t0 \quad t1 \quad t2 \quad t3 \quad t4 \quad t5 \quad t6 \quad t7 \quad \ldots \]

Instruction1

Instruction2

\[ \text{IF}_1 \quad \text{ID}_1 \quad \text{EX}_1 \quad \text{MA}_1 \quad \text{WB}_1 \]

\[ \text{IF}_2 \quad \text{ID}_2 \quad \text{EX}_2 \quad \text{MA}_2 \quad \text{WB}_2 \]
5-Stage Pipelined Execution

Instruction Flow Diagram

I-Fetch (IF)

Decode, Reg. Fetch (ID)

Execute (EX)

Memory (MA)

Write Back (WB)

Time

Instruction 1

Instruction 2

Instruction 3

\( t_0 \)

\( t_1 \)

\( t_2 \)

\( t_3 \)

\( t_4 \)

\( t_5 \)

\( t_6 \)

\( t_7 \)

\ldots

IR

Addr

Rdata

GPRs

ALU

Addr

Rdata

Wdata

IR

PC

Add

Inst. Memory

Memory

Write Back

Add

0x4
5-Stage Pipelined Execution

Instruction Flow Diagram

- I-Fetch (IF)
- Decode, Reg. Fetch (ID)
- Execute (EX)
- Memory (MA)
- Write-Back (WB)

<table>
<thead>
<tr>
<th>Time</th>
<th>Instruction 1</th>
<th>Instruction 2</th>
<th>Instruction 3</th>
<th>Instruction 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0</td>
<td>IF&lt;sub&gt;1&lt;/sub&gt;</td>
<td>IF&lt;sub&gt;2&lt;/sub&gt;</td>
<td>IF&lt;sub&gt;3&lt;/sub&gt;</td>
<td>IF&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td>t1</td>
<td>ID&lt;sub&gt;1&lt;/sub&gt;</td>
<td>ID&lt;sub&gt;2&lt;/sub&gt;</td>
<td>ID&lt;sub&gt;3&lt;/sub&gt;</td>
<td>ID&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td>t2</td>
<td>EX&lt;sub&gt;1&lt;/sub&gt;</td>
<td>EX&lt;sub&gt;2&lt;/sub&gt;</td>
<td>EX&lt;sub&gt;3&lt;/sub&gt;</td>
<td>EX&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td>t3</td>
<td>MA&lt;sub&gt;1&lt;/sub&gt;</td>
<td>MA&lt;sub&gt;2&lt;/sub&gt;</td>
<td>MA&lt;sub&gt;3&lt;/sub&gt;</td>
<td>MA&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td>t4</td>
<td>WB&lt;sub&gt;1&lt;/sub&gt;</td>
<td>WB&lt;sub&gt;2&lt;/sub&gt;</td>
<td>WB&lt;sub&gt;3&lt;/sub&gt;</td>
<td>WB&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td>t5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5-Stage Pipelined Execution

Instruction Flow Diagram

- **I-Fetch (IF)**: Fetches the next instruction from memory.
- **Decode, Reg. Fetch (ID)**: Decodes the instruction and fetches registers.
- **Execute (EX)**: Performs the execution of the instruction.
- **Memory (MA)**: Performs memory operations.
- **Write-Back (WB)**: Writes back the results to memory.

**Time and Instructions**:
- **t0**: IF1 (instruction1)
- **t1**: ID1 (instruction1)
- **t2**: EX1 (instruction1)
- **t3**: MA1 (instruction1)
- **t4**: WB1 (instruction1)
- **t5**: IF2 (instruction2)
- **t6**: ID2 (instruction2)
- **t7**: EX2 (instruction2)
- **t8**: MA2 (instruction2)
- **t9**: WB2 (instruction2)
- **t10**: IF3 (instruction3)
- **t11**: ID3 (instruction3)
- **t12**: EX3 (instruction3)
- **t13**: MA3 (instruction3)
- **t14**: WB3 (instruction3)
- **t15**: IF4 (instruction4)
- **t16**: ID4 (instruction4)
- **t17**: EX4 (instruction4)
- **t18**: MA4 (instruction4)
- **t19**: WB4 (instruction4)
- **t20**: IF5 (instruction5)
- **t21**: ID5 (instruction5)
- **t22**: EX5 (instruction5)
- **t23**: MA5 (instruction5)
- **t24**: WB5 (instruction5)

**Operations**:
- **Addr**: Addressing
- **Write-Back (WB)**: Writes back results to memory
- **Add**: Addition
- **Imm**: Immediate
- **Ext**: Extension
- **Inst.**: Instruction
- **GPRs**: General Purpose Registers
- **ALU**: Arithmetic Logic Unit
- **IR**: Instruction Register
- **PC**: Program Counter
5-Stage Pipelined Execution

Instruction Flow Diagram

- **I-Fetch (IF)**
- **Decode, Reg. Fetch (ID)**
- **Execute (EX)**
- **Memory (MA)**
- **Write-Back (WB)**

*Time Points:*
- **t0**
- **t1**
- **t2**
- **t3**
- **t4**
- **t5**
- **t6**
- **t7**

*Instructions:*
- Instruction 1
- Instruction 2
- Instruction 3
- Instruction 4
- Instruction 5

*Critical Paths:*
- Addr, rdata
- Instruction
- Memory
- Write-back

*Cardinalities:*
- **PC**: 0x4
- **Add**: 1

*Data Paths:*
- Address
- Write
- Read
5-Stage Pipelined Execution

Resource Usage Diagram

I-Fetch (IF)  Decode, Reg. Fetch (ID)  Execute (EX)  Memory (MA)  Write-Back (WB)
5-Stage Pipelined Execution

Resource Usage Diagram

- **I-Fetch (IF)**: PC, addr, rdata, Inst., Memory
- **Decode, Reg. Fetch (ID)**: rs1, rs2, rd1, we, ws, wdrd2, GPRs, Imm Ext
- **Execute (EX)**: ALU, addr, rdata, Data, Memory, wdata
- **Memory (MA)**: addr
- **Write-Back (WB)**: Write

Resources:

- time: t0, t1, t2, t3, t4, t5, t6, t7, ....
5-Stage Pipelined Execution

Resource Usage Diagram

I-Fetch (IF)

Decode, Reg. Fetch (ID)

Execute (EX)

Memory (MA)

Write-Back (WB)

Resources

IF

\( t_0 \) \quad \( t_1 \) \quad \( t_2 \) \quad \( t_3 \) \quad \( t_4 \) \quad \( t_5 \) \quad \( t_6 \) \quad \( t_7 \) \ldots \)
5-Stage Pipelined Execution

Resource Usage Diagram

**I-Fetch (IF)**

**Decode, Reg. Fetch (ID)**

**Execute (EX)**

**Memory (MA)**

Write Back (WB)

Resources:

- **IF**
  - Time: t0, t1, t2, t3, t4, t5
  - Resources: \( I_1, I_2, I_3, I_4, I_5 \)

- **ID**
  - Time: t6, t7
  - Resources: \( I_1, I_2, I_3, I_4, I_5 \)

**Resources Used:***

- **Write Back (WB):**
  - Time: t6, t7
  - Resources: \( I_1, I_2, I_3, I_4, I_5 \)
5-Stage Pipelined Execution

Resource Usage Diagram

```
<table>
<thead>
<tr>
<th>Resources</th>
<th>IF</th>
<th>ID</th>
<th>EX</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>t0</td>
<td>t1</td>
<td>t2</td>
</tr>
<tr>
<td></td>
<td>t3</td>
<td>t4</td>
<td>t5</td>
</tr>
<tr>
<td></td>
<td>t6</td>
<td>t7</td>
<td>......</td>
</tr>
<tr>
<td>IF</td>
<td>I₁</td>
<td>I₂</td>
<td>I₃</td>
</tr>
<tr>
<td>ID</td>
<td>I₄</td>
<td>I₅</td>
<td>I₆</td>
</tr>
<tr>
<td>EX</td>
<td>I₇</td>
<td>I₈</td>
<td>I₉</td>
</tr>
</tbody>
</table>
```
5-Stage Pipelined Execution
Resource Usage Diagram

Resources

<table>
<thead>
<tr>
<th>Resources</th>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>, I_1</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_2</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_3</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_4</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_5</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_6</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>, I_7</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>time</th>
<th>t0</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>t6</th>
<th>t7</th>
<th>. . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>I_1</td>
<td>I_2</td>
<td>I_3</td>
<td>I_4</td>
<td>I_5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td>I_1</td>
<td>I_2</td>
<td>I_3</td>
<td>I_4</td>
<td>I_5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX</td>
<td></td>
<td></td>
<td>I_1</td>
<td>I_2</td>
<td>I_3</td>
<td>I_4</td>
<td>I_5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA</td>
<td></td>
<td></td>
<td></td>
<td>I_1</td>
<td>I_2</td>
<td>I_3</td>
<td>I_4</td>
<td>I_5</td>
<td></td>
</tr>
</tbody>
</table>

February 25, 2019

MIT 6.823 Spring 2019

L06-21
The diagram illustrates the 5-stage pipelined execution of a typical microprocessor, focusing on resource usage and timing. Each stage is labeled with its primary function:

- **I-Fetch (IF)**: This stage fetches the instruction from memory. It uses resources for address generation and instruction decoding.
- **Decode, Reg. Fetch (ID)**: Here, the instruction is decoded, and the required registers are fetched. Resources are used for instruction processing and register access.
- **Execute (EX)**: In this stage, the instruction is executed, and data is processed. Resources include the ALU for arithmetic and logic operations, and data memory access.
- **Memory (MA)**: This stage handles memory access, crucial for data retrieval or storage. Resources include memory access and data transfer.
- **Write-Back (WB)**: Finally, the results are written back to the appropriate registers. Resources are used for data write-back and update.

The diagram also highlights the timing sequence with time labels: t0, t1, t2, t3, t4, t5, t6, t7, indicating the progression through the pipeline stages. The resources used are marked for each stage, showing the dynamic allocation of resources across the pipeline.

In summary, the 5-stage pipelined execution effectively manages the flow of instructions and data, optimizing the processing efficiency of the microprocessor.
5-Stage Pipelined Execution

Resource Usage Diagram

- **I-Fetch (IF)**
- **Decode, Reg. Fetch (ID)**
- **Execute (EX)**
- **Memory (MA)**
- **Write-Back (WB)**

**Resources**
- **IF**
- **ID**
- **EX**
- **MA**
- **WB**

**Time**
- t0
- t1
- t2
- t3
- t4
- t5
- t6
- t7

**Instructions**
- I1
- I2
- I3
- I4
- I5

**Resources**
- PC
- Addr
- Rdata
- Inst.
- Memory
- Addr
- Rdata
- Wdata
- GPRs
- PC
- Imm
- Ext
- ALU
- Memory
- Addr
- Rdata
- Wdata

**Adders**
- 0x4

**PC**
- Add
Pipelined Execution

ALU Instructions

![Diagram of pipelined ALU execution](image)
Pipelined Execution

ALU Instructions

Not quite correct!
Pipelined Execution

ALU Instructions

Not quite correct!

We need an Instruction Reg (IR) for each stage
Not quite correct!

We need an Instruction Reg (IR) for each stage
Not quite correct!

We need an Instruction Reg (IR) for each stage
Pipelined Execution

ALU Instructions

Not quite correct!

We need an Instruction Reg (IR) for each stage
Pipelined MIPS Datapath
without jumps

What else is needed?
Pipelined MIPS Datapath

without jumps

Control Points Need to Be Connected

What else is needed?
Pipelined MIPS Datapath

without jumps

What else is needed?

Control Points Need to Be Connected
Pipelined MIPS Datapath
without jumps

What else is needed?

Control Points Need to Be Connected
Pipelined MIPS Datapath
without jumps

What else is needed?
Pipelined MIPS Datapath

without jumps

What else is needed?

Control Points Need to Be Connected
Pipelined MIPS Datapath without jumps

What else is needed?
How instructions can interact with each other in a pipeline
How instructions can interact with each other in a pipeline

• An instruction in the pipeline may need a resource being used by another instruction in the pipeline → structural hazard
How instructions can interact with each other in a pipeline

• An instruction in the pipeline may need a resource being used by another instruction in the pipeline → structural hazard

• An instruction may depend on a value produced by an earlier instruction
  
  – Dependence may be for a data calculation → data hazard
  
  – Dependence may be for calculating the next PC → control hazard (branches, interrupts)
Data Hazards

... 
\( r1 \leftarrow r0 + 10 \) 
\( r4 \leftarrow r1 + 17 \) 
...
Data Hazards

... 

\[ r1 \leftarrow r0 + 10 \]

\[ r4 \leftarrow r1 + 17 \]

...
Data Hazards

... 
\[ r1 \leftarrow r0 + 10 \]
\[ r4 \leftarrow r1 + 17 \]
...
Data Hazards

... r1 ← r0 + 10
r4 ← r1 + 17
...

r1 is stale. Oops!
Resolving Data Hazards

Strategy 1: *Wait for the result to be available by freezing earlier pipeline stages* \(\rightarrow\) *stall*

Strategy 2: *Route data as soon as possible after it is calculated to the earlier pipeline stage* \(\rightarrow\) *bypass*

Strategy 3: *Speculate on the dependence*

Two cases:
Resolving Data Hazards

Strategy 1: *Wait for the result to be available by freezing earlier pipeline stages* → *stall*

Strategy 2: *Route data as soon as possible after it is calculated to the earlier pipeline stage* → *bypass*

Strategy 3: *Speculate on the dependence*

  *Two cases:*

  *Guessed correctly* → *do nothing*
Resolving Data Hazards

Strategy 1: *Wait for the result to be available by freezing earlier pipeline stages → stall*

Strategy 2: *Route data as soon as possible after it is calculated to the earlier pipeline stage → bypass*

Strategy 3: *Speculate on the dependence*
   
   Two cases:
   
   - Guessed correctly → do nothing
   - Guessed incorrectly → kill and restart
Strategy 1:

Wait for the result to be available by freezing earlier pipeline stages \(\rightarrow\) stall
Feedback to Resolve Hazards

• Later stages provide dependence information to earlier stages which can *stall (or kill) instructions*
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can *stall (or kill) instructions*
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can stall (or kill) instructions
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can *stall (or kill) instructions*
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can *stall (or kill) instructions*
Feedback to Resolve Hazards

Later stages provide dependence information to earlier stages which can stall (or kill) instructions.
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can stall (or kill) instructions
Later stages provide dependence information to earlier stages which can *stall* (or *kill*) instructions.
Feedback to Resolve Hazards

- Later stages provide dependence information to earlier stages which can *stall (or kill) instructions*.

- Controlling a pipeline in this manner works provided the instruction at stage $i+1$ can complete without any interference from instructions in stages 1 to $i$ (otherwise deadlocks may occur).
Resolving Data Hazards by Stalling

... $r1 \leftarrow r0 + 10$

... $r4 \leftarrow r1 + 17$

...
Resolving Data Hazards by Stalling

... r1 ← r0 + 10
r4 ← r1 + 17
...

February 25, 2019
Resolving Data Hazards by Stalling

... 

r1 ← r0 + 10
r4 ← r1 + 17
...

February 25, 2019
Resolving Data Hazards by Stalling

---

Stall Condition

... 
\[ r_1 \leftarrow r_0 + 10 \] 
\[ r_4 \leftarrow r_1 + 17 \] 
...

---
Stalled Stages and Pipeline Bubbles
Stalled Stages and Pipeline Bubbles

\[ \text{time} \]
\[ t_0 \ t_1 \ t_2 \ t_3 \ t_4 \ t_5 \ t_6 \ t_7 \ \ldots \]
Stalled Stages and Pipeline Bubbles

\[ \begin{align*}
\text{time} \\
t_0 & \quad t_1 & \quad t_2 & \quad t_3 & \quad t_4 & \quad t_5 & \quad t_6 & \quad t_7 & \quad \ldots \ldots \\
(I_1) & \quad r_1 \leftarrow (r_0) + 10 & \quad IF_1 & \quad ID_1 & \quad EX_1 & \quad MA_1 & \quad WB_1
\end{align*} \]
Stalled Stages and Pipeline Bubbles

<table>
<thead>
<tr>
<th>time</th>
<th>t0</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>t6</th>
<th>t7</th>
<th>....</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_1) r1 ← (r0) + 10</td>
<td>IF_1</td>
<td>ID_1</td>
<td>EX_1</td>
<td>MA_1</td>
<td>WB_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_2) r4 ← (r1) + 17</td>
<td>IF_2</td>
<td>ID_2</td>
<td>ID_2</td>
<td>ID_2</td>
<td>ID_2</td>
<td>EX_2</td>
<td>MA_2</td>
<td>WB_2</td>
<td></td>
</tr>
</tbody>
</table>
Stalled Stages and Pipeline Bubbles

\[ \begin{array}{cccccccccc}
\text{time} & t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & \ldots \\
(I_1) & r_1 & \leftarrow (r_0) & + & 10 & \text{IF}_1 & \text{ID}_1 & \text{EX}_1 & \text{MA}_1 & \text{WB}_1 \\
(I_2) & r_4 & \leftarrow (r_1) & + & 17 & \text{IF}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{EX}_2 & \text{MA}_2 & \text{WB}_2 \\
(I_3) & & & & & \text{IF}_3 & \text{IF}_3 & \text{IF}_3 & \text{IF}_3 & \text{ID}_3 & \text{EX}_3 & \text{MA}_3 & \text{WB}_3 \\
\end{array} \]
Stalled Stages and Pipeline Bubbles

\[
\begin{array}{c}
\text{time} \\
t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 & \ldots
\end{array}
\]

(I_1) \ r1 \leftarrow (r0) + 10 \\
\quad \begin{array}{c}
\text{IF}_1 \\
\text{ID}_1 \\
\text{EX}_1 \\
\text{MA}_1 \\
\text{WB}_1
\end{array}

(I_2) \ r4 \leftarrow (r1) + 17 \\
\quad \begin{array}{c}
\text{IF}_2 \\
\text{ID}_2 \\
\text{ID}_2 \\
\text{ID}_2 \\
\text{ID}_2 \\
\text{EX}_2 \\
\text{MA}_2 \\
\text{WB}_2
\end{array}

(I_3) \\
\quad \begin{array}{c}
\text{IF}_3 \\
\text{ID}_3 \\
\text{IF}_3 \\
\text{IF}_3 \\
\text{IF}_3 \\
\text{ID}_3 \\
\text{EX}_3 \\
\text{MA}_3 \\
\text{WB}_3
\end{array}

(I_4) \\
\quad \begin{array}{c}
\text{IF}_4 \\
\text{ID}_4 \\
\text{EX}_4 \\
\text{MA}_4 \\
\text{WB}_4
\end{array}

February 25, 2019
Stalled Stages and Pipeline Bubbles

\begin{align*}
\text{time} & \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad \ldots \\
(I_1) & \quad r_1 \leftarrow (r_0) + 10 & \text{IF}_1 & \text{ID}_1 & \text{EX}_1 & \text{MA}_1 & \text{WB}_1 \\
(I_2) & \quad r_4 \leftarrow (r_1) + 17 & \text{IF}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{EX}_2 & \text{MA}_2 & \text{WB}_2 \\
(I_3) & \quad & \text{IF}_3 & \text{ID}_3 & \text{IF}_3 & \text{IF}_3 & \text{IF}_3 & \text{EX}_3 & \text{MA}_3 & \text{WB}_3 \\
(I_4) & \quad & \text{IF}_4 & \text{ID}_4 & \text{EX}_4 & \text{MA}_4 & \text{WB}_4 \\
(I_5) & \quad & \text{IF}_5 & \text{ID}_5 & \text{EX}_5 & \text{MA}_5 & \text{WB}_5
\end{align*}
Stalled Stages and Pipeline Bubbles

\[
\begin{array}{ccccccccc}
\text{time} & t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & \ldots \\
(I_1) & r_1 & \leftarrow & (r_0) + 10 & \text{IF}_1 & \text{ID}_1 & \text{EX}_1 & \text{MA}_1 & \text{WB}_1 & \\
(I_2) & r_4 & \leftarrow & (r_1) + 17 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{EX}_2 & \text{MA}_2 & \text{WB}_2 \\
(I_3) & & & & & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{EX}_3 & \text{MA}_3 & \text{WB}_3 \\
(I_4) & & & & & & \text{ID}_4 & \text{ID}_4 & \text{EX}_4 & \text{MA}_4 & \text{WB}_4 \\
(I_5) & & & & & & & \text{ID}_5 & \text{EX}_5 & \text{MA}_5 & \text{WB}_5 \\
\end{array}
\]
Stalled Stages and Pipeline Bubbles

\[ \text{time} \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \ldots \]

1. \( r_1 \leftarrow (r_0) + 10 \) IF
2. \( r_4 \leftarrow (r_1) + 17 \) IF

\( \ldots \text{stalled stages} \ldots \)
### Stalled Stages and Pipeline Bubbles

**Resource Usage**
Stalled Stages and Pipeline Bubbles

\[ \text{time} \]
\[
\begin{array}{cccccccc}
  t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 \\
  \underline{IF} & \underline{ID} & \underline{EX} & \underline{MA} & \underline{WB} & \underline{IF} & \underline{ID} & \underline{EX} \\
\end{array}
\]

\[ (I_1) \ r1 \leftarrow (r0) + 10 \]
\[ (I_2) \ r4 \leftarrow (r1) + 17 \]

stalled stages

\[ \text{Resource Usage} \]
Stalled Stages and Pipeline Bubbles

\[
\begin{align*}
\text{time} & \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad \ldots \ldots \\
(I_1) \quad r_1 & \leftarrow (r_0) + 10 \quad \text{IF}_1 \\
(I_2) \quad r_4 & \leftarrow (r_1) + 17 \quad \text{IF}_2 \\
(I_3) & \quad \text{IF}_3 \\
(I_4) & \quad \text{IF}_4 \\
(I_5) & \quad \text{IF}_5 \\
\end{align*}
\]

stalled stages

\[
\begin{align*}
\text{Resource Usage} & \\
\end{align*}
\]
Stalled Stages and Pipeline Bubbles

\[
\begin{align*}
(I_1) & \quad r_1 \leftarrow (r_0) + 10 \\
(I_2) & \quad r_4 \leftarrow (r_1) + 17 \\
(I_3) & \\
(I_4) & \\
(I_5) &
\end{align*}
\]

Resource Usage

\[
\begin{array}{ccccccccccc}
& \text{time} & t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & \ldots \\
\text{IF} & I_1 & I_2 & I_3 & I_3 & I_3 & I_3 & I_4 & I_5 & & \\
\text{ID} & I_1 & I_2 & I_2 & I_2 & I_2 & I_3 & I_4 & I_5 & & \\
\end{array}
\]

\[
\text{stalled stages}
\]
Stalled Stages and Pipeline Bubbles

\[
\begin{align*}
\text{time} & \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad \ldots \\
(I_1) & \quad r_1 \leftarrow (r_0) + 10 \\
(I_2) & \quad r_4 \leftarrow (r_1) + 17 \\
(I_3) & \\
(I_4) & \\
(I_5) & \\
\end{align*}
\]

\textbf{Resource Usage}

\[
\begin{align*}
\text{IF} & \quad I_1 \quad I_2 \quad I_3 \quad I_3 \quad I_3 \quad I_4 \quad I_5 \\
\text{ID} & \quad I_1 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \quad I_3 \quad I_4 \quad I_5 \\
\text{EX} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad I_2 \quad I_3 \quad I_4 \quad I_5
\end{align*}
\]
Stalled Stages and Pipeline Bubbles

\[
\begin{array}{cccccccc}
\text{time} & t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 & \ldots \\
(I_1) & r1 \leftarrow (r0) + 10 & \text{IF}_1 & \text{ID}_1 & \text{EX}_1 & \text{MA}_1 & \text{WB}_1 \\
(I_2) & r4 \leftarrow (r1) + 17 & \text{IF}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{EX}_2 & \text{MA}_2 & \text{WB}_2 \\
(I_3) & \text{IF}_3 & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{EX}_3 & \text{MA}_3 & \text{WB}_3 \\
(I_4) & \text{IF}_4 & \text{ID}_4 & \text{ID}_4 & \text{EX}_4 & \text{MA}_4 & \text{WB}_4 \\
(I_5) & \text{IF}_5 & \text{ID}_5 & \text{EX}_5 & \text{MA}_5 & \text{WB}_5 \\
\end{array}
\]

Resource Usage

\[
\begin{array}{cccccccc}
\text{time} & t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 & \ldots \\
\text{IF} & I_1 & I_2 & I_3 & I_3 & I_3 & I_4 & I_5 \\
\text{ID} & I_1 & I_2 & I_2 & I_2 & I_3 & I_4 & I_5 \\
\text{EX} & I_1 & \text{nop} & \text{nop} & \text{nop} & \text{nop} & I_2 & I_3 & I_4 & I_5 \\
\text{MA} & I_1 & \text{nop} & \text{nop} & \text{nop} & \text{nop} & I_2 & I_3 & I_4 & I_5 \\
\end{array}
\]
Stalled Stages and Pipeline Bubbles

\begin{align*}
\text{(I}_1\text{)} & \ r_1 \leftarrow (r_0) + 10 \\
\text{(I}_2\text{)} & \ r_4 \leftarrow (r_1) + 17 \\
\text{(I}_3\text{)} & \\
\text{(I}_4\text{)} & \\
\text{(I}_5\text{)} & \\
\end{align*}

\begin{align*}
\text{time} & \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad \ldots \\
\text{IF} & \quad I_1 \quad I_2 \quad I_3 \quad I_3 \quad I_3 \quad I_4 \quad I_5 \\
\text{ID} & \quad I_1 \quad I_2 \quad I_2 \quad I_2 \quad I_3 \quad I_4 \quad I_5 \\
\text{EX} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad I_2 \quad I_3 \\
\text{MA} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad I_2 \quad I_3 \\
\text{WB} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad I_2 \quad I_3 \\
\end{align*}
Stalled Stages and Pipeline Bubbles

\[
\begin{array}{cccccccc}
\text{time} & t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 & \ldots \\
(I_1) & r1 & \leftarrow (r0) + 10 & \text{IF}_1 & \text{ID}_1 & \text{EX}_1 & \text{MA}_1 & \text{WB}_1 & \\
(I_2) & r4 & \leftarrow (r1) + 17 & \text{IF}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \text{ID}_2 & \\
(I_3) & \text{IF}_3 & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{ID}_3 & \text{EX}_2 & \text{MA}_2 & \text{WB}_2 & \\
(I_4) & \text{IF}_4 & \text{ID}_4 & \text{ID}_4 & \text{ID}_4 & \text{ID}_4 & \text{EX}_3 & \text{MA}_3 & \text{WB}_3 & \\
(I_5) & \text{IF}_5 & \text{ID}_5 & \text{ID}_5 & \text{ID}_5 & \text{ID}_5 & \text{EX}_4 & \text{MA}_4 & \text{WB}_4 & \\
\end{array}
\]

\text{stalled stages}

Resource Usage

\[
\begin{array}{cccccccc}
\text{time} & t0 & t1 & t2 & t3 & t4 & t5 & t6 & t7 & \ldots \\
\text{IF} & I_1 & I_2 & I_3 & I_3 & I_3 & I_3 & \text{I}_4 & \text{I}_5 & \\
\text{ID} & I_1 & I_2 & I_2 & I_2 & I_2 & I_2 & \text{I}_4 & \text{I}_5 & \\
\text{EX} & I_1 & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{I}_4 & \text{I}_5 & \\
\text{MA} & I_1 & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{I}_4 & \text{I}_5 & \\
\text{WB} & I_1 & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{nop} & \text{I}_4 & \text{I}_5 & \\
\end{array}
\]
Stalled Stages and Pipeline Bubbles

\( (I_1) r_1 \leftarrow (r_0) + 10 \)
\( (I_2) r_4 \leftarrow (r_1) + 17 \)
\( (I_3) \)
\( (I_4) \)
\( (I_5) \)

\begin{align*}
\text{time} & \quad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad \ldots \\
\text{IF} & \quad I_1 \quad I_2 \quad I_3 \quad I_3 \quad I_3 \quad I_3 \quad I_3 \quad I_3 \quad I_3 \\
\text{ID} & \quad I_1 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \quad I_2 \\
\text{EX} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \\
\text{MA} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \\
\text{WB} & \quad I_1 \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \quad \text{nop} \\
\end{align*}

\( \text{nop} \Rightarrow \text{pipeline bubble} \)
Compare the *source registers* of the instruction in the decode stage with the *destination register* of the *uncommitted instructions*. 
Stall Control Logic

Compare the source registers of the instruction in the decode stage with the destination register of the uncommitted instructions.
Compare the *source registers* of the instruction in the decode stage with the *destination register* of the *uncommitted instructions*. 
Compare the source registers of the instruction in the decode stage with the destination register of the uncommitted instructions.
Compare the *source registers* of the instruction in the decode stage with the *destination register* of the *uncommitted instructions*. 
Thank you!

Next lecture: Control Hazards, Bypassing and Speculation