Designing a Pipelined CPU

Peer Instruction Lecture Materials for Computer Architecture by Dr. Leo Porter, adapted by Janet Davis, are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
Review -- Single Cycle CPU
Which of the statements below is true about a pipelined processor?

Choice  Statement
A  Both instruction latency and throughput remain essentially unchanged
B  Instruction latency remains essentially unchanged from single-cycle;
    Instruction throughput increases by a factor of 5
C  Instruction latency improves by a factor of 5 over single-cycle;
    Instruction throughput remains essentially unchanged
D  Both latency and throughput improve by a factor of 5 over single-cycle
E  None of the above
Idea for a Pipelined Datapath
Execution in a Pipelined Datapath
Execution in a Pipelined Datapath
# Pipeline Stages

Should we force every instruction to go through all 5 stages?  
Can R-type taking 4 cycles instead of 5?

<table>
<thead>
<tr>
<th>Selection</th>
<th>Yes/No</th>
<th>Reason (Choose BEST answer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Yes</td>
<td>Decreasing R-type to 4 cycles improves instruction throughput</td>
</tr>
<tr>
<td>B</td>
<td>Yes</td>
<td>Decreasing R-type to 4 cycles improves instruction latency</td>
</tr>
<tr>
<td>C</td>
<td>No</td>
<td>Decreasing R-type to 4 cycles causes hazards</td>
</tr>
<tr>
<td>D</td>
<td>No</td>
<td>Decreasing R-type to 4 cycles doesn’t improve throughput</td>
</tr>
<tr>
<td>E</td>
<td>No</td>
<td>Decreasing R-type to 4 cycles doesn’t improve latency</td>
</tr>
</tbody>
</table>
Mixed Instructions in the Pipeline
Pipeline Principles

• All instructions that share a pipeline must have the same stages in the same order.
  – therefore, &lt;add&gt; does nothing during Mem stage
  – &lt;sw&gt; does nothing during WB stage

• All intermediate values must be latched each cycle.

• There is no functional block reuse
Pipelined Datapath

Instruction Fetch  Instruction Decode/  Execute/  Memory Access  Write Back
Register Fetch  Address Calculation

IF/ID  ID/EX  EX/MEM  MEM/WB

PC  Instruction memory  Address  Instruction

0 Mux 1
Add
4

IF/ID

Instruction

ID/EX

Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2

ID/EX

Shift left 2

EX/MEM

Add

EX/MEM

ALU
Zero
result

ALU
result

MEM/WB

Address
Data memory

Read data
Write data

1 Mux 1

0 Mux 1

16
Sign extend

32

registers!
Which is the best statement?

A. All the pipeline registers are the same size.
B. The ID/EX register is the same size as the EX/MEM register.
C. The IF/ID register is the largest; MEM/WB is the smallest.
D. The ID/EX register is the largest; IF/ID is the smallest.
E. None of the above.
Where does multiplication execute?

A. The ID stage
B. The EX stage
C. The MEM stage
D. The WB stage
E. None of the above
The Pipeline with Control Logic
Execution in a Pipelined Datapath
sub $2, $1, $3

and $12, $2, $5

or $13, $6, $2

add $14, $2, $2

sw $15, 100($2)

What just happened here which is problematic (BEST ANSWER)?
A. The register file is trying to read and write the same register.
B. The ALU and data memory are both active in the same cycle.
C. A value is used before it is produced.
D. Both A and B
E. Both A and C
Data Hazards

- When a result is needed in the pipeline before it is available, a “data hazard” occurs.

```
sub $2, $1, $3
and $12, $2, $5
or $13, $6, $2
add $14, $2, $2
sw $15, 100($2)
```
Hazards continued

• What happens when...
  add $3, $10, $11
  lw $8, 1000($3)
  sub $11, $8, $7