ECE 2300 Digital Logic and Computer Organization, Fall 2024
Discussion Section 8 – Pipelined Processors

revision: 2024-11-14-23-16

In this discussion section, you will explore three different pipelined processor microarchitectures which support executing just the TinyRV1 arithmetic instructions (i.e., `addi`, `add`, `mul`) and memory instructions (i.e., `lw`, `sw`). The first microarchitecture is the two-stage fully bypassed pipelined processor presented in lecture. The second micorarchitecture is a three-stage pipelined processor which uses a mix of hardware stalling and bypassing, while the final microarchitecture is a three-stage fully bypassed pipelined processor.

We will be using the following assembly program which calculates $Y = aX + Y$ where a is a scalar value and X and Y are vectors (arrays) of the same length. This is called the SAXPY kernel which stands for "Single-Precision A·X Plus Y". SAXPY is one of the most basic linear algebra kernels. Technically SAXPY uses a single-precision floating point number system, but here we are using a two's complement integer number system.

Pseudo-Code

```
Y[0]  = a * X[0]  + Y[0]
Y[1]  = a * X[1]  + Y[1]
Y[2]  = a * X[1]  + Y[2]
...
Y[63] = a * X[63] + Y[63]
```

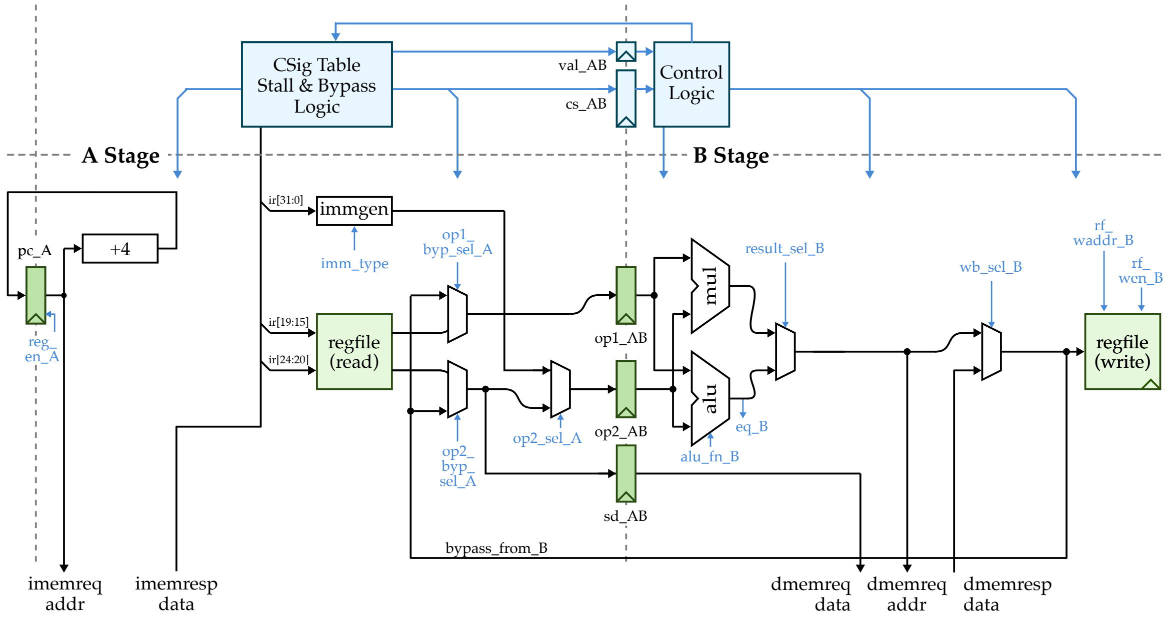
Assembly Code

```
 # a:x1, addr(X[i]):x2, addr(Y[i]):x3

 lw   x4, 0(x3)

 lw   x5, 0(x2)

 mul  x6, x5, x1

 add  x7, x6, x4

 sw   x7, 0(x3)

 addi x2, x2, 4

 addi x3, x3, 4
```

For all parts, assume the length of the vectors (arrays) is 64 (i.e., the above seven instructions are repeated 64 times). **To get started, draw the architectural RAW dependency arrows on the above assembly program.**

Problem 1. Two-Stage Pipelined Processor

Consider the datapath for the two-stage pipelined processor shown below. This datapath does not support the control flow instructions (i.e., it cannot execute `jal`, `jr`, and `bne`). The datapath is "fully bypassed" meaning every reasonable bypass path is included.



Part 1.A Minimum Clock Period

What is the minimum clock period (T_c) in units of τ for the two-stage fully bypassed processor? Assume the critical path does not through the control unit. To justify your answer, show a timing delay inequality that clearly identifies the name of component and also clearly identifies the delay of each component. **Highlight the critical path on the datapath diagram.**

	t_{pd}
32-bit 2-to-1 Mux	4τ
32-bit 4-to-1 Mux	8τ
32-bit Adder	60τ
32-bit ALU	64τ
32-bit Multiplier	100τ
32-bit +4 Unit	30τ
ImmGen Unit	12τ
32-bit Reg Clk-to-Q	9τ
32-bit Reg Setup	10τ
Register File Read	25τ
Register File Setup	20τ
Memory Read	120τ
Memory Setup	120τ

Part 1.B Execution Time

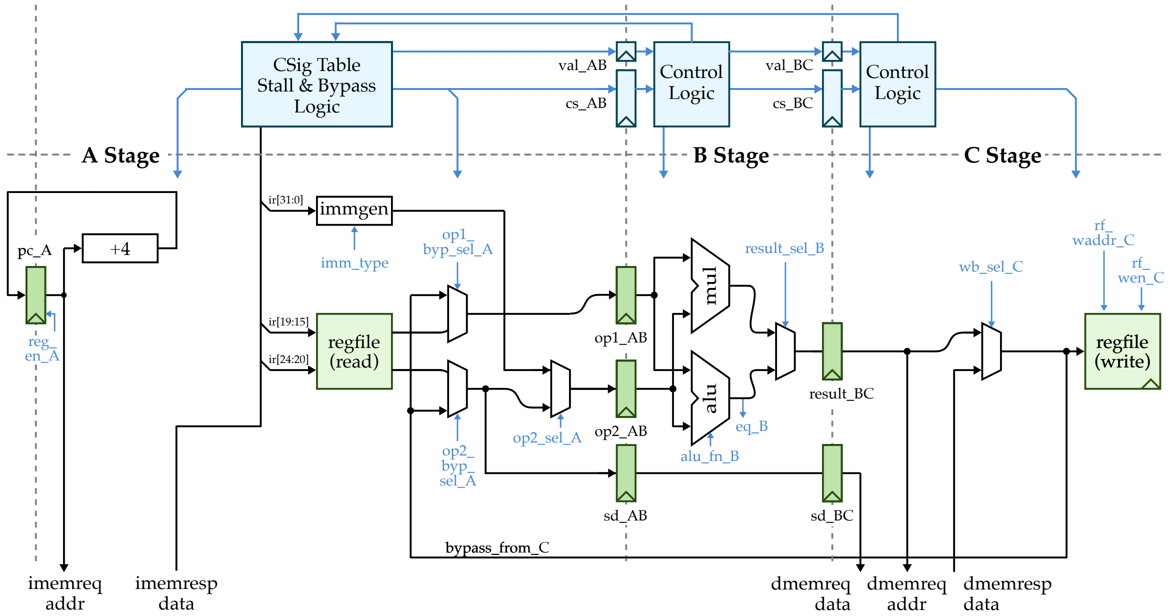
Draw the pipeline diagram for the SAXPY kernel running on the two-stage fully bypassed processor. Include all RAW microarchitectural dependency arrows. You do not need to use every row of the diagram, but add enough instructions to ensure you can analyze the steady-state behavior.

`lw   x4, 0(x3)`																		
`lw   x5, 0(x2)`																		
`mul  x6, x5, x1`																		
`add  x7, x6, x4`																		
`sw   x7, 0(x3)`																		
`addi x2, x2, 4`																		
`addi x3, x3, 4`																		

Calculate the total execution time in units of τ for the SAXPY kernel running on the two-stage fully bypassed processor. Show your work.

Problem 2. Three-Stage Partially Bypassed Pipelined Processor

Consider the datapath for the three-stage pipelined processor shown below. This datapath does not support the control flow instructions (i.e., it cannot execute `jal`, `jr`, and `bne`). The datapath is "partially bypassed" meaning it will stall to resolve some RAW hazards because the corresponding bypass paths are not implemented in the datapath.



Part 2.A Minimum Clock Period

What is the minimum clock period (T_c) in units of τ for the three-stage partially bypassed processor? Assume the critical path does not through the control unit. To justify your answer, show a timing delay inequality that clearly identifies the name of component and also clearly identifies the delay of each component. **Highlight the critical path on the datapath diagram.**

	t_{pd}
32-bit 2-to-1 Mux	4τ
32-bit 4-to-1 Mux	8τ
32-bit Adder	60τ
32-bit ALU	64τ
32-bit Multiplier	100τ
32-bit +4 Unit	30τ
ImmGen Unit	12τ
32-bit Reg Clk-to-Q	9τ
32-bit Reg Setup	10τ
Register File Read	25τ
Register File Setup	20τ
Memory Read	120τ
Memory Setup	120τ

Part 2.B Execution Time

Draw the pipeline diagram for the SAXPY kernel running on the three-stage partially bypassed processor. Include all RAW microarchitectural dependency arrows. You do not need to use every row of the diagram, but add enough instructions to ensure you can analyze the steady-state behavior. *Every vertical arrow in the pipeline diagram must correspond to a bypass path in the datapath. Make sure that you do not assume a bypass path which does not exist in the datapath!*

`lw   x4, 0(x3)`																		
`lw   x5, 0(x2)`																		
`mul  x6, x5, x1`																		
`add  x7, x6, x4`																		
`sw   x7, 0(x3)`																		
`addi x2, x2, 4`																		
`addi x3, x3, 4`																		

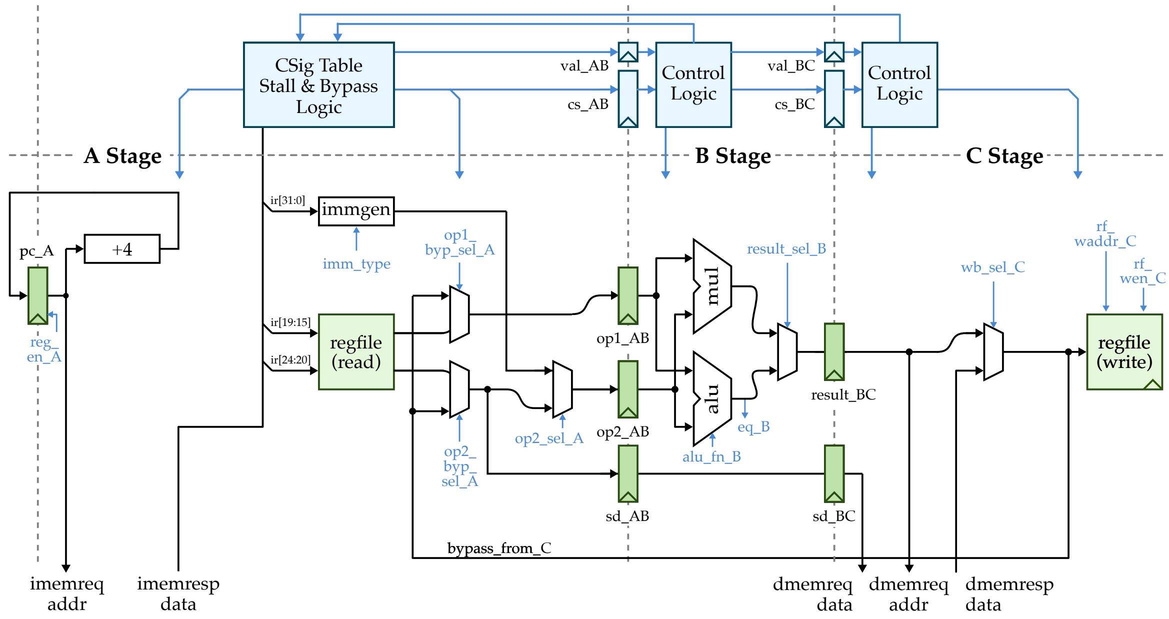
Calculate the total execution time in units of τ for the SAXPY kernel running on the three-stage partially bypassed processor. Show your work.

Part 2.C Stall Signal

Derive the stall signal for the three-stage partially bypassed processor.

Problem 3. Three-Stage Fully Bypassed Pipelined Processor

Consider the datapath for the three-stage pipelined processor shown below. This datapath does not support the control flow instructions (i.e., it cannot execute `jal`, `jr`, and `bne`).



Part 3.A Adding Bypass Paths

Add bypass paths to the datapath diagram to ensure it is "fully bypassed" meaning every reasonable bypass path is included.

Part 3.B Minimum Clock Period

What is the minimum clock period (T_c) in units of τ for the three-stage fully bypassed processor? Assume the critical path does not through the control unit. To justify your answer, show a timing delay inequality that clearly identifies the name of component and also clearly identifies the delay of each component. **Highlight the critical path on the datapath diagram.**

	t_{pd}
32-bit 2-to-1 Mux	4τ
32-bit 4-to-1 Mux	8τ
32-bit Adder	60τ
32-bit ALU	64τ
32-bit Multiplier	100τ
32-bit +4 Unit	30τ
ImmGen Unit	12τ
32-bit Reg Clk-to-Q	9τ
32-bit Reg Setup	10τ
Register File Read	25τ
Register File Setup	20τ
Memory Read	120τ
Memory Setup	120τ

Part 3.C Execution Time

Draw the pipeline diagram for the SAXPY kernel running on the three-stage fully bypassed processor. Include all RAW microarchitectural dependency arrows. You do not need to use every row of the diagram, but add enough instructions to ensure you can analyze the steady-state behavior. *Every vertical arrow in the pipeline diagram must correspond to a bypass path in the datapath. Make sure that you do not assume a bypass path which does not exist in the datapath!*

`lw   x4, 0(x3)`																		
`lw   x5, 0(x2)`																		
`mul  x6, x5, x1`																		
`add  x7, x6, x4`																		
`sw   x7, 0(x3)`																		
`addi x2, x2, 4`																		
`addi x3, x3, 4`																		

Calculate the total execution time in units of τ for the SAXPY kernel running on the three-stage fully bypassed processor. Show your work.

Part 3.D Bypass Signals

Derive the new bypass signals for the three-stage fully bypassed processor. Are there still any stall signals?

Problem 4. Comparative Analysis

Use your data from the previous problems to fill in the table below. The time/cycle and time/program should be in units of τ or kilo-τ (kτ).

	Inst/Prog	Cycles/Inst	Time/Cycle	Time/Prog
Fully Bypassed 2-Stage				
Partially Bypassed 3-Stage				
Fully Bypassed 3-Stage				

Compare and contrast the three microarchitectures in terms of delay (quantitative) and area (qualititatve).