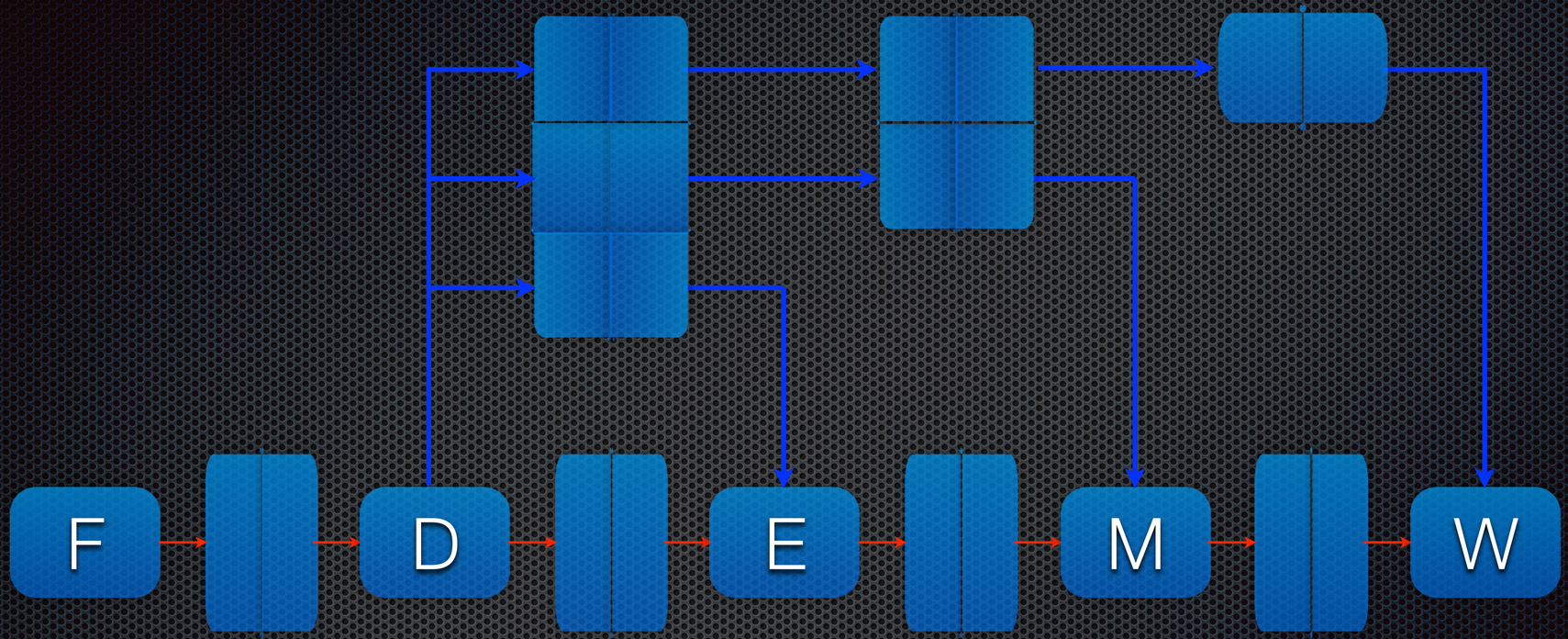
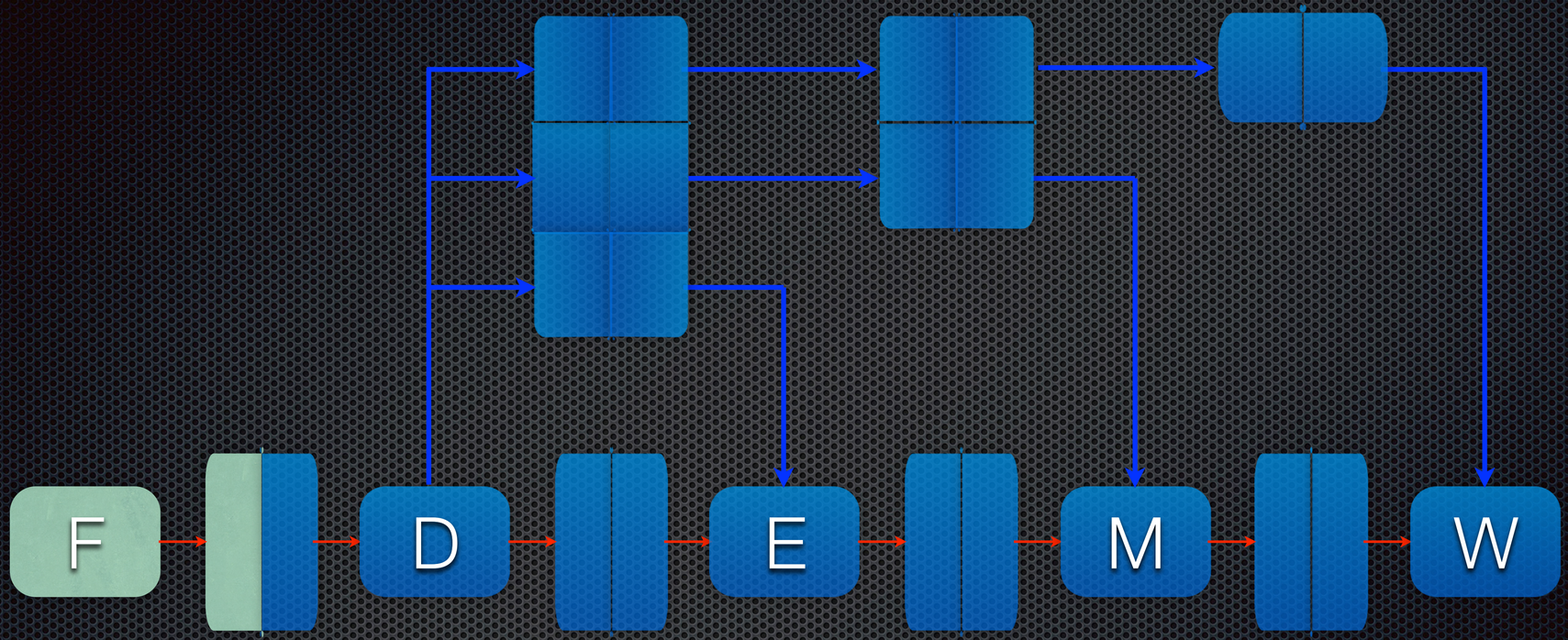
Managing a Pipeline

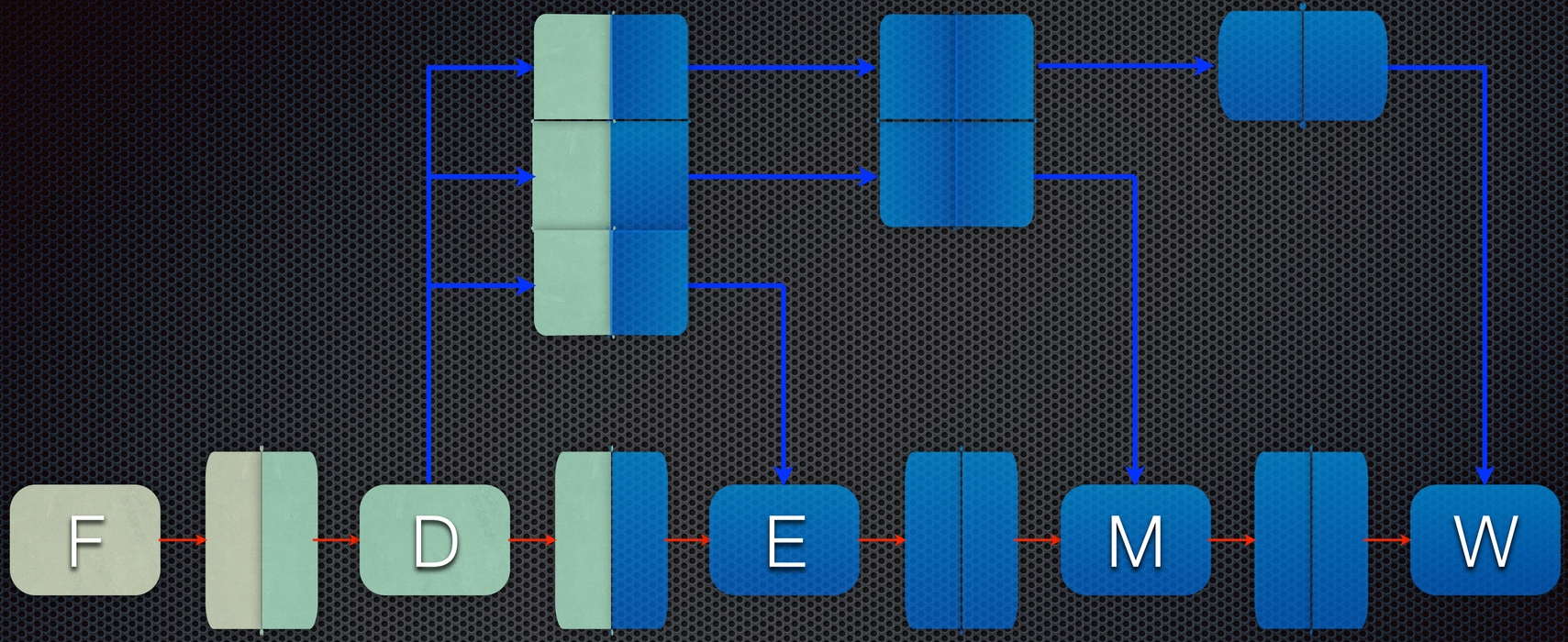
No valves, but plenty of bubbles

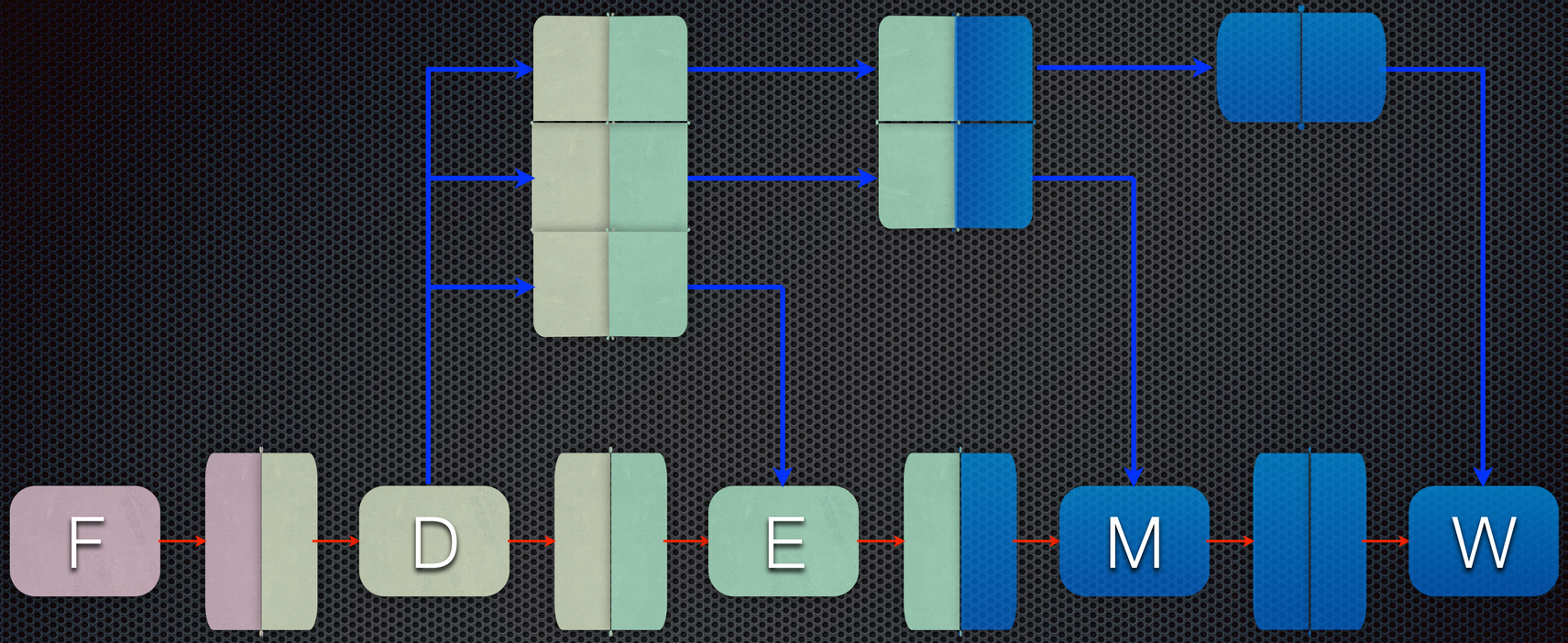
Pipeline Control

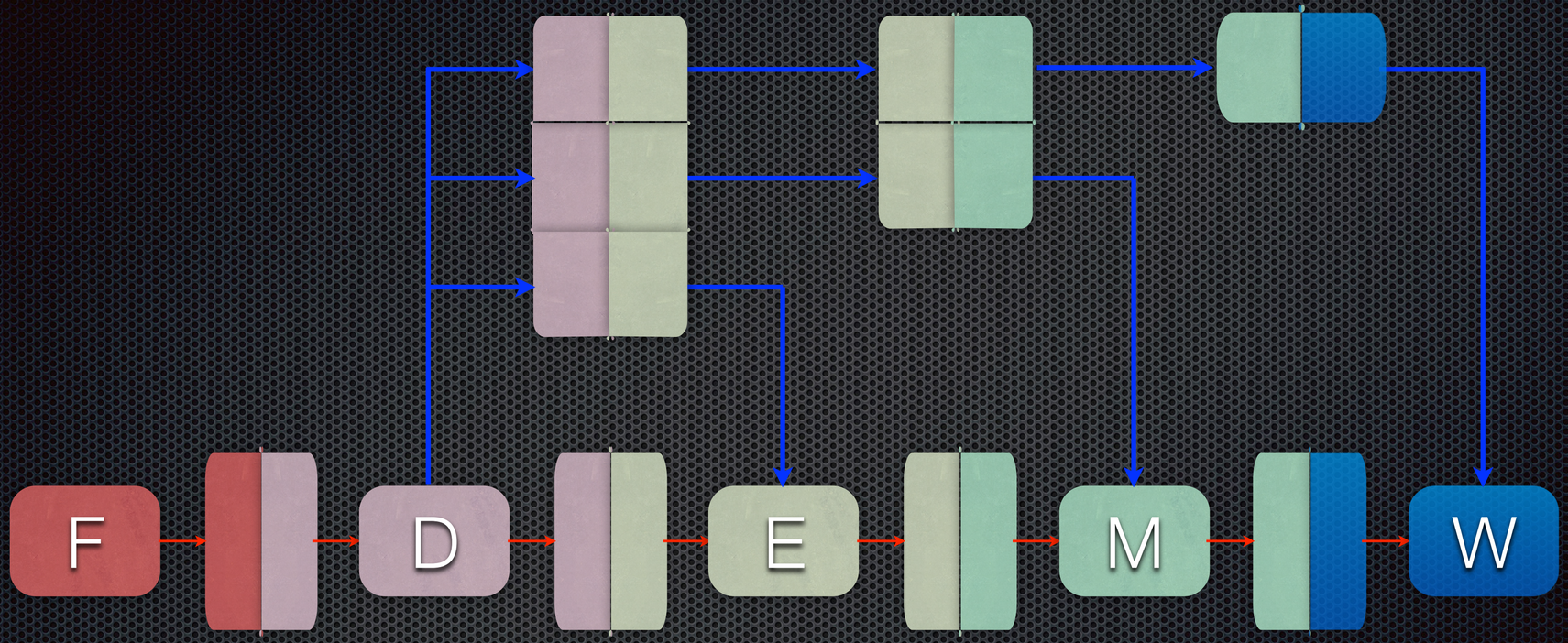
- How does the CU keep track of what each stage should be doing for each instruction in the pipe?
- At decode, it generates all of the control signals for the current instruction, for every stage, and stores them in a set of control registers
- The control registers are also pipelined
- At each stage, one register is “consumed” and the others are passed on to the next stage

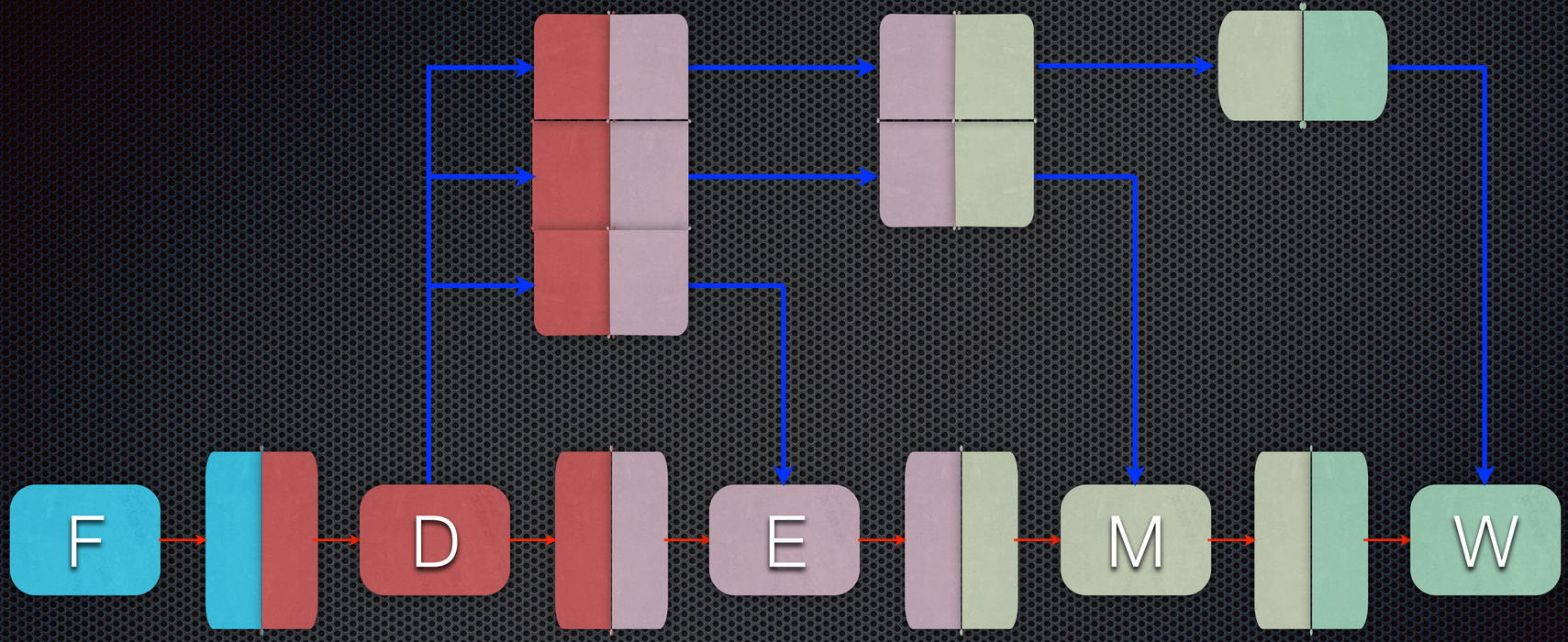

F
D
E
M
W

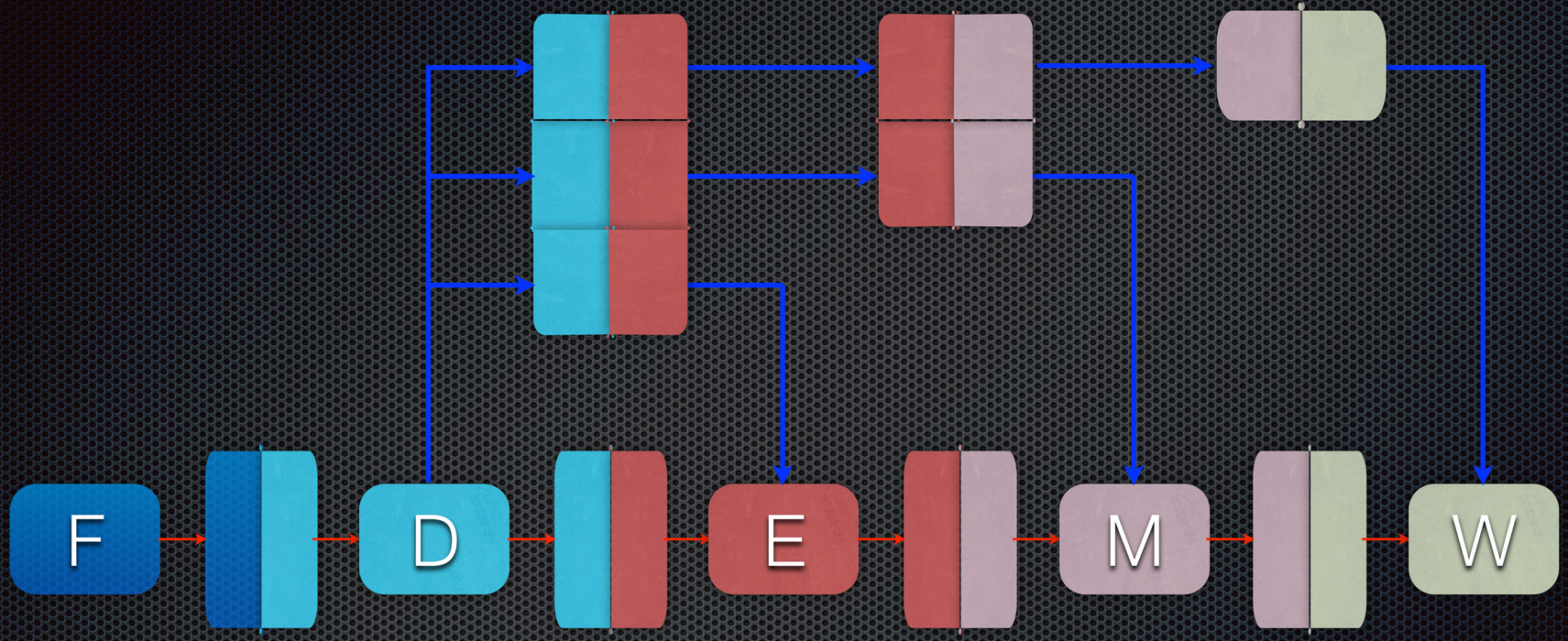

F
D
E
M
W

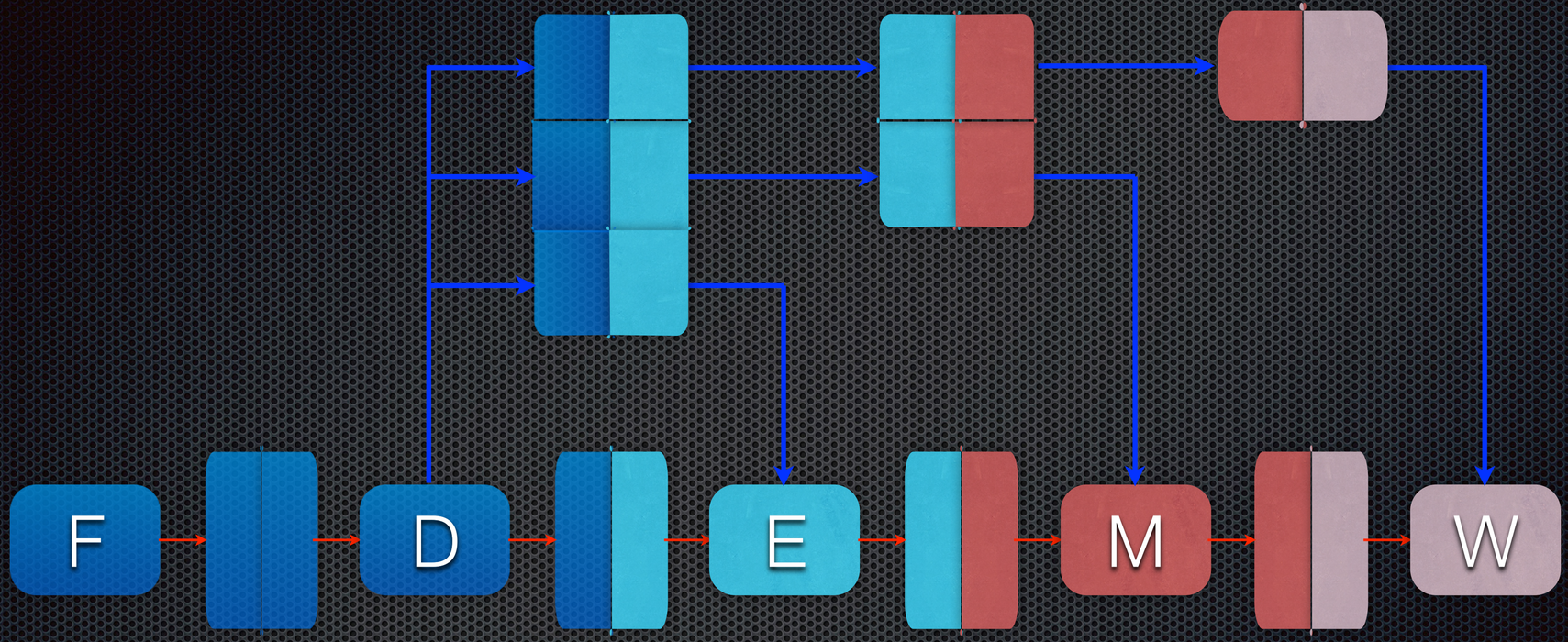

F
D
E
M
W

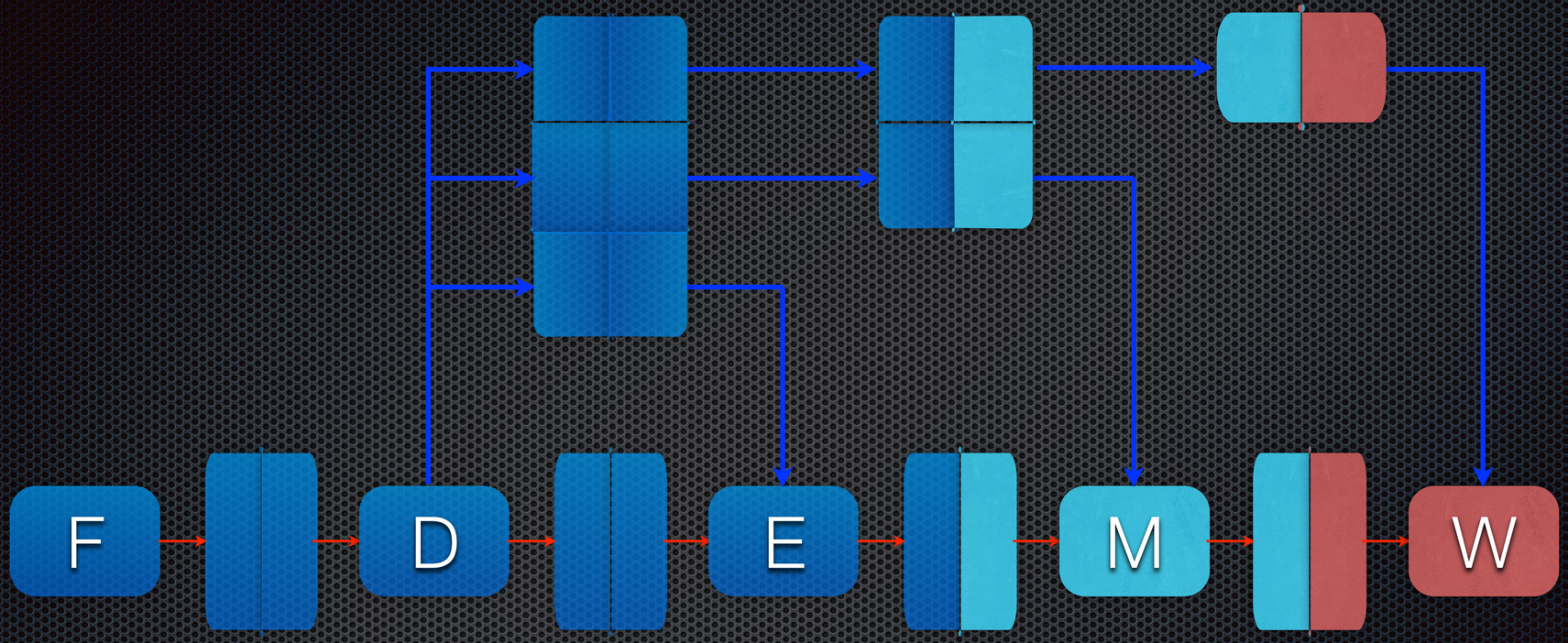

F
D
E
M
W

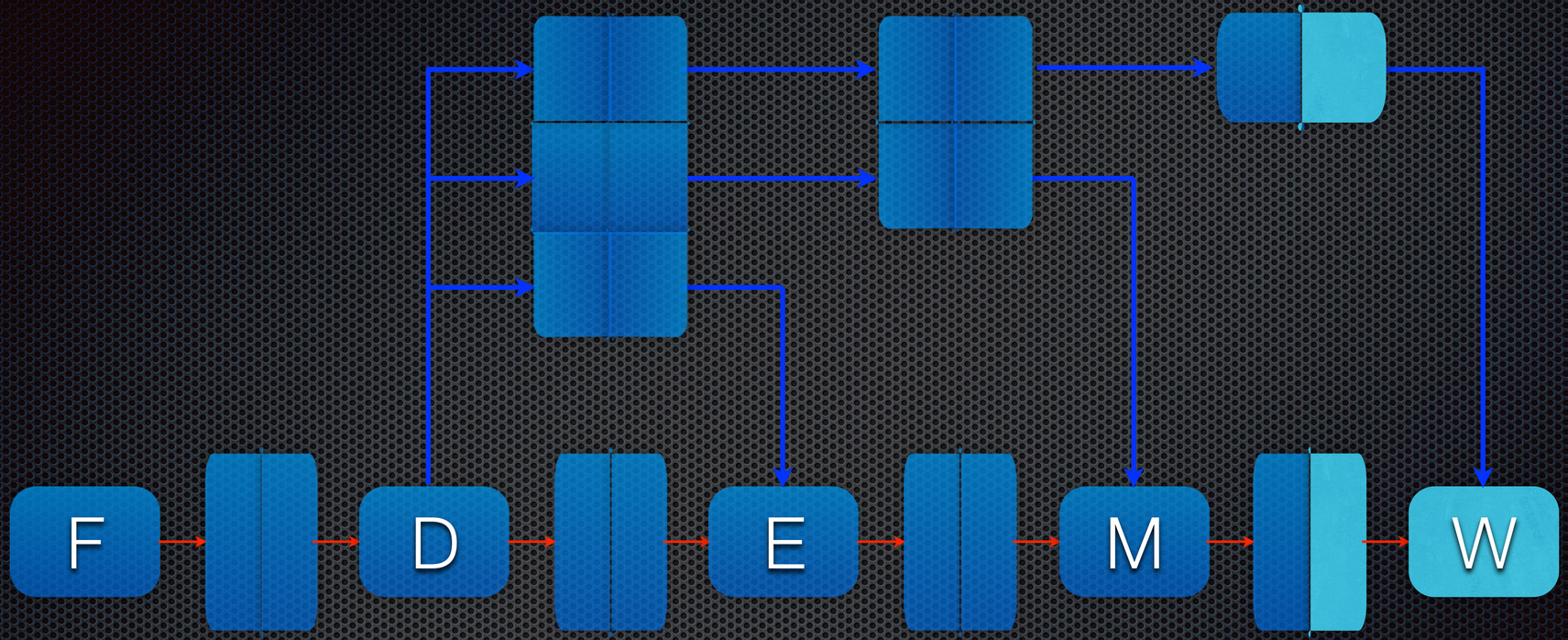

F
D
E
M
W

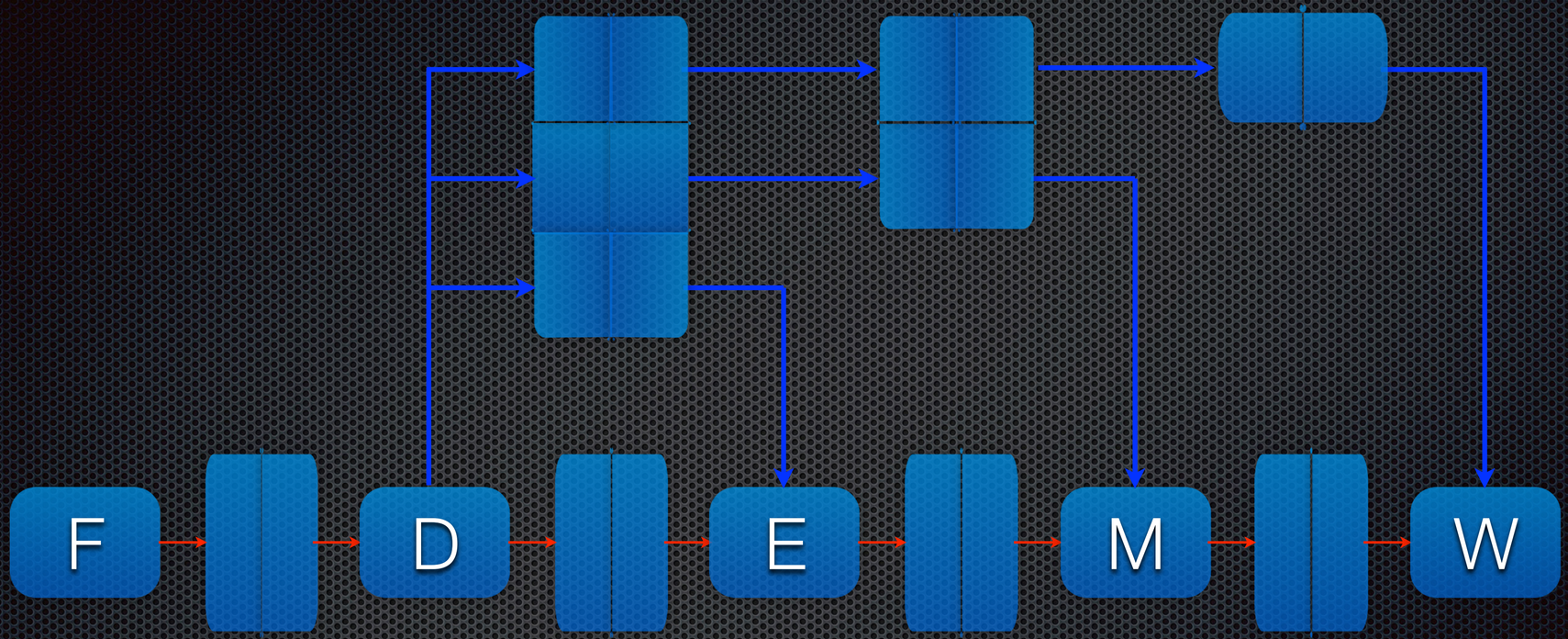

F
D
E
M
W


F
D
E
M
W


F
D
E
M
W

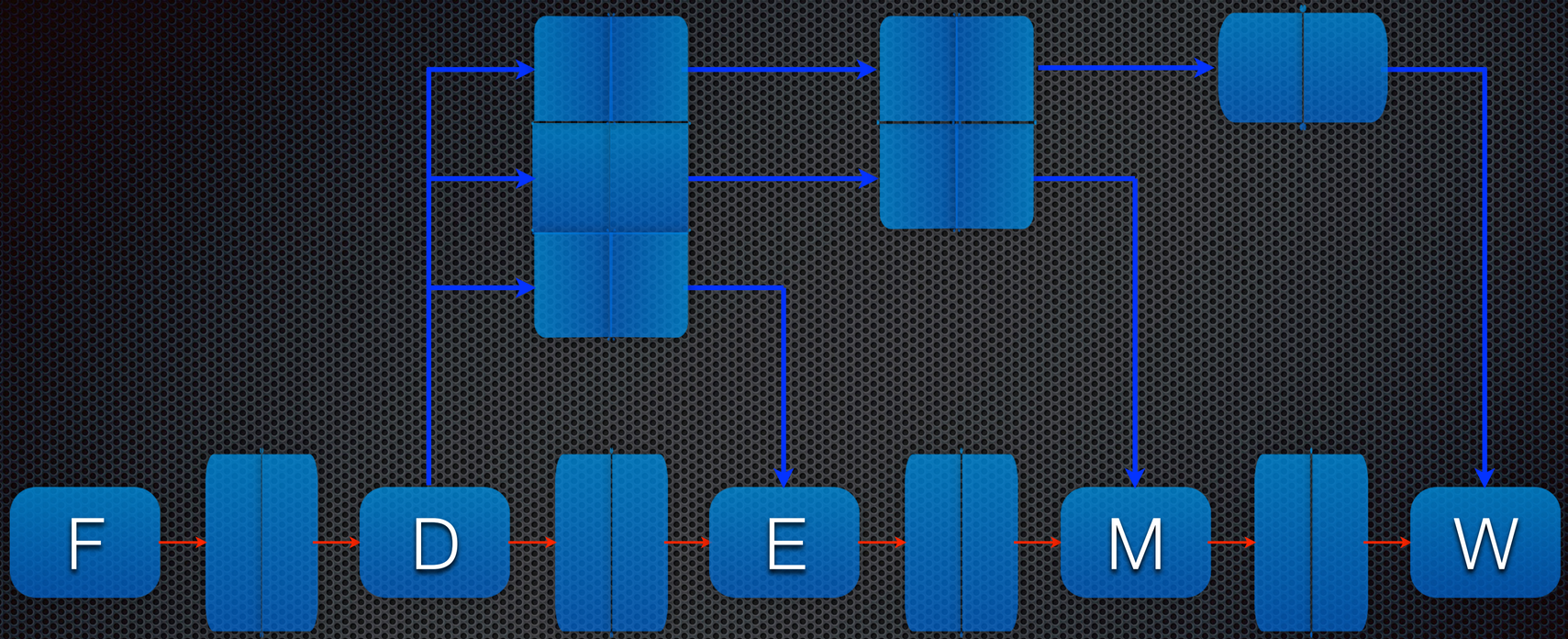

F
D
E
M
W

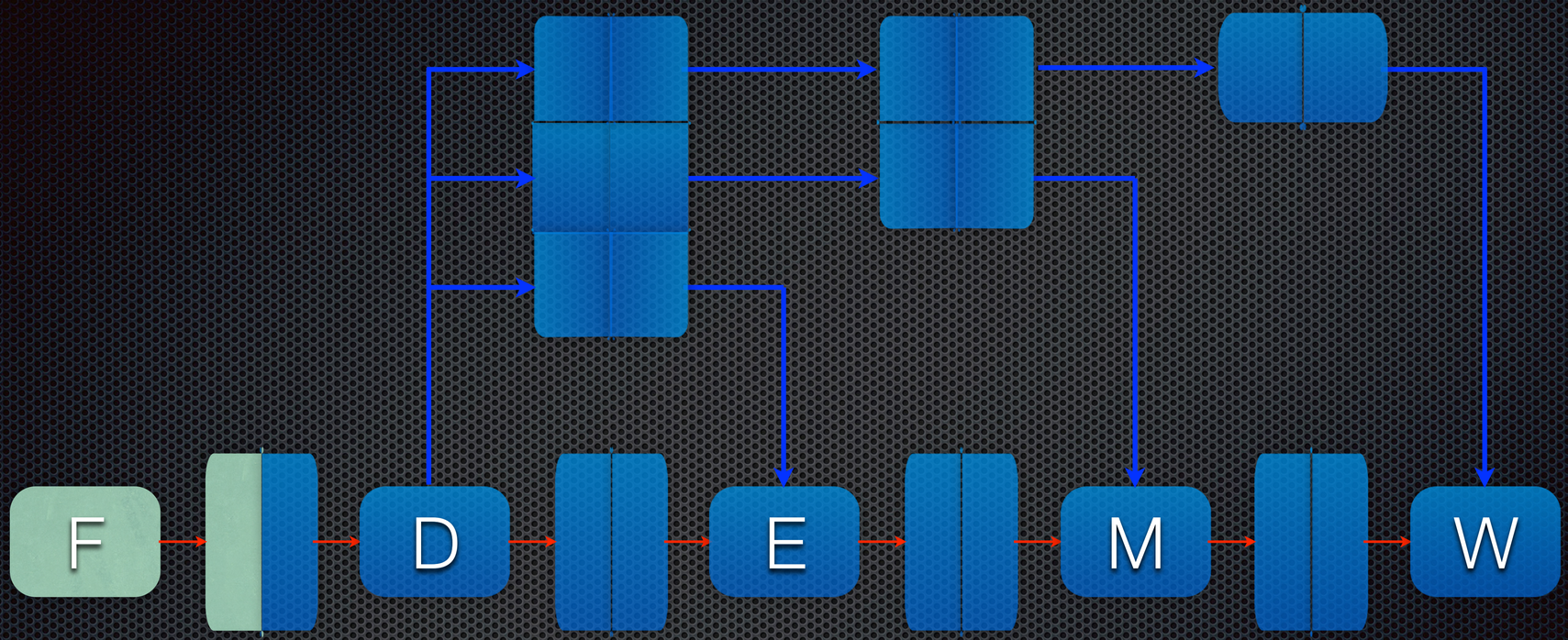

F
D
E
M
W

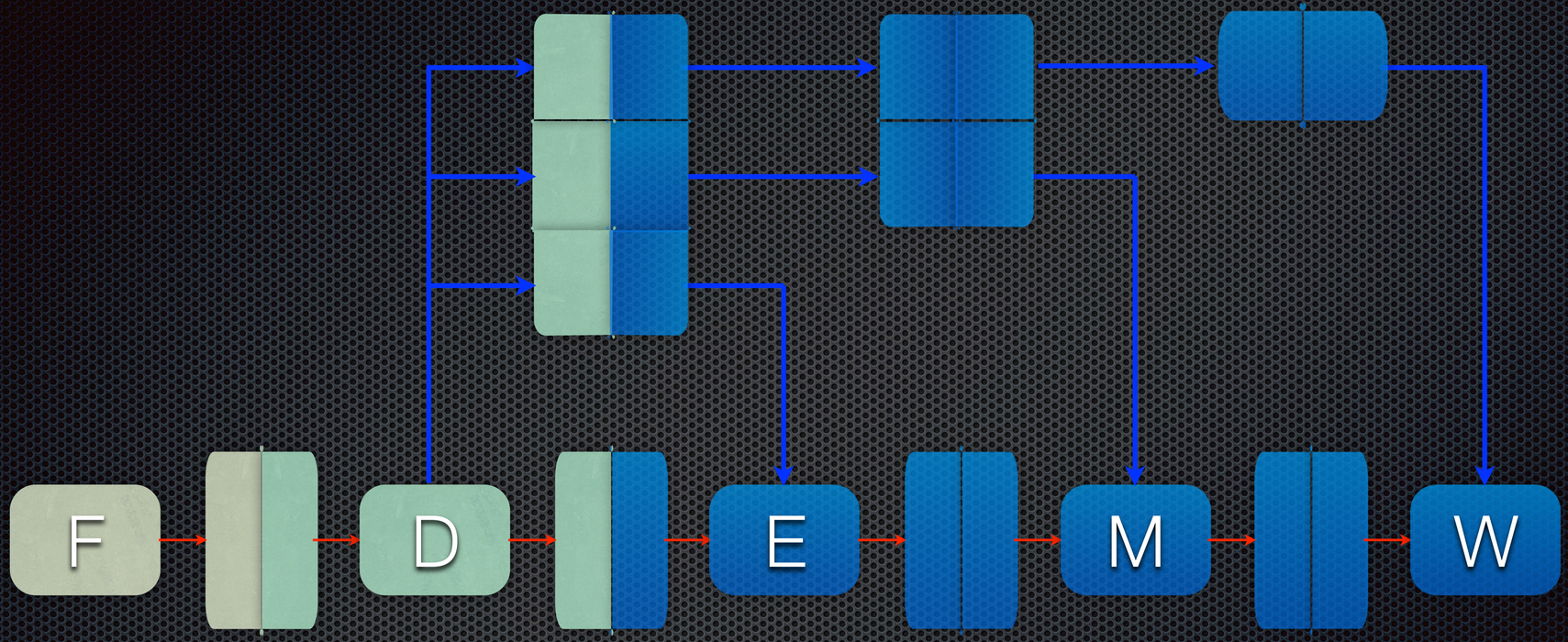

F
D
E
M
W

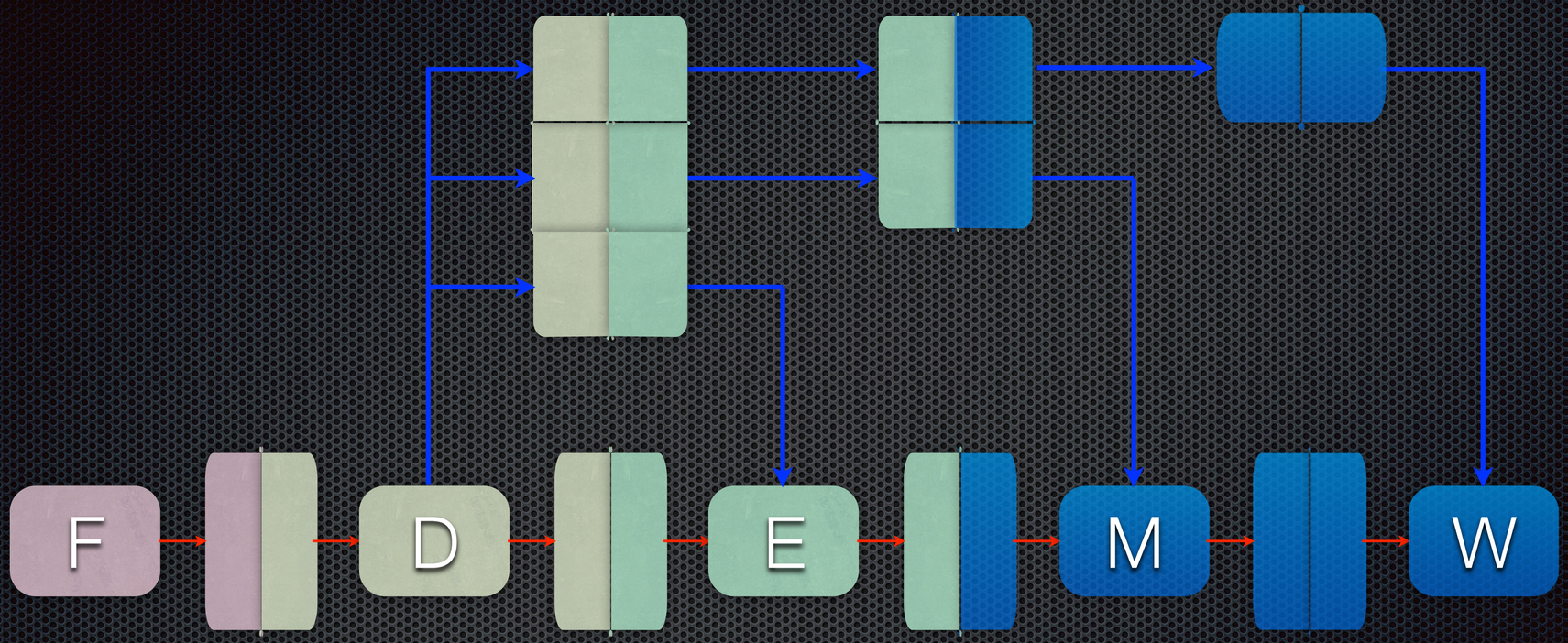
Simulation

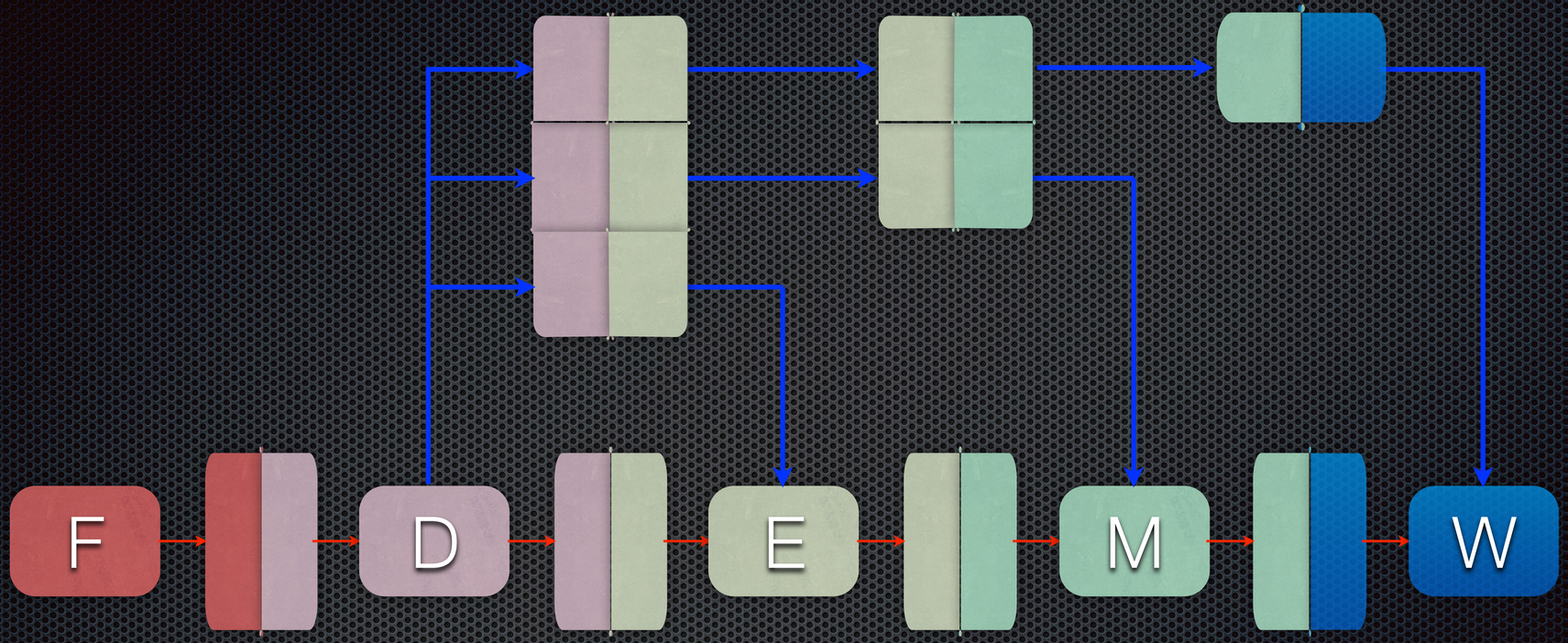
- On each cycle, evaluation proceeds back to front
- When a stage advances, its opens a space for the one behind it to advance (like a queue)
- Stages that take multiple cycles don’t advance, and a “bubble” occurs
- A stage is blocked from advancing if the next stage is still full
- A multi-cycle operation may be able to do work while blocked from advancing (e.g., FP DIV in Execute while a cache miss is serviced in Memory)

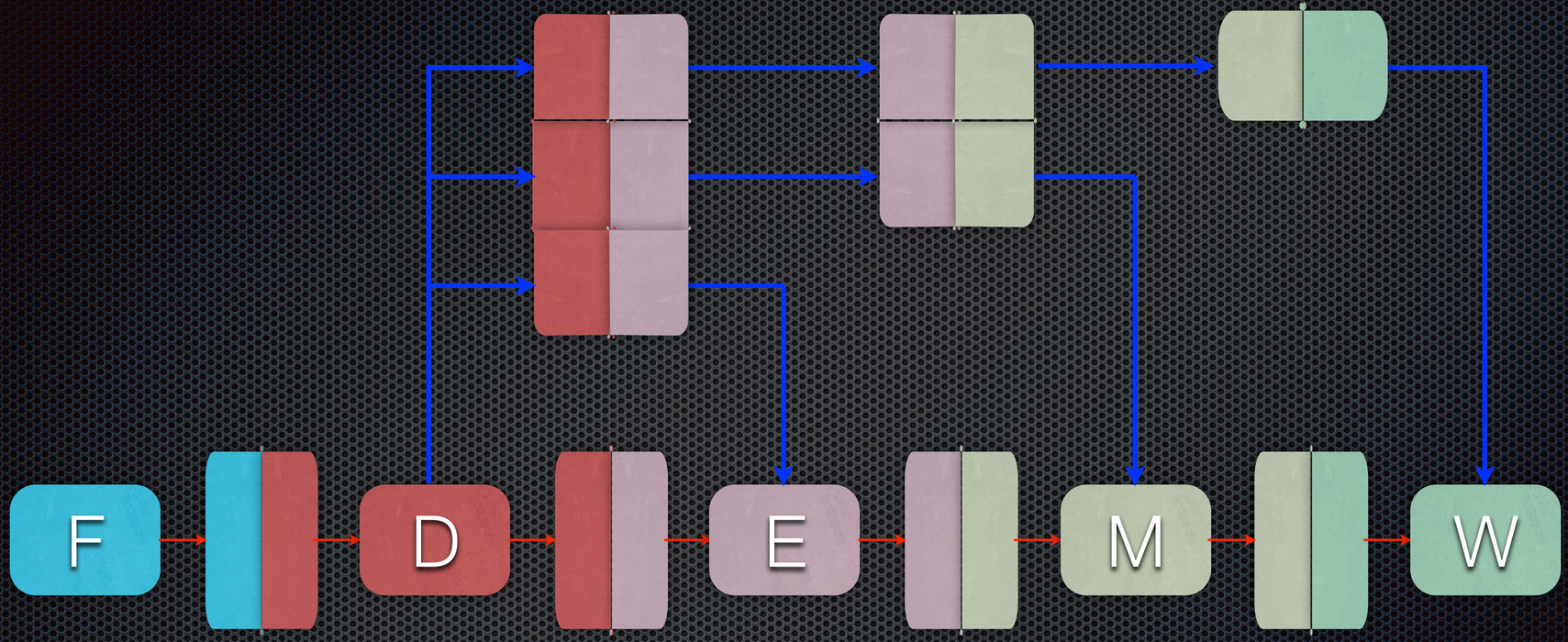

F
D
E
M
W


F
D
E
M
W

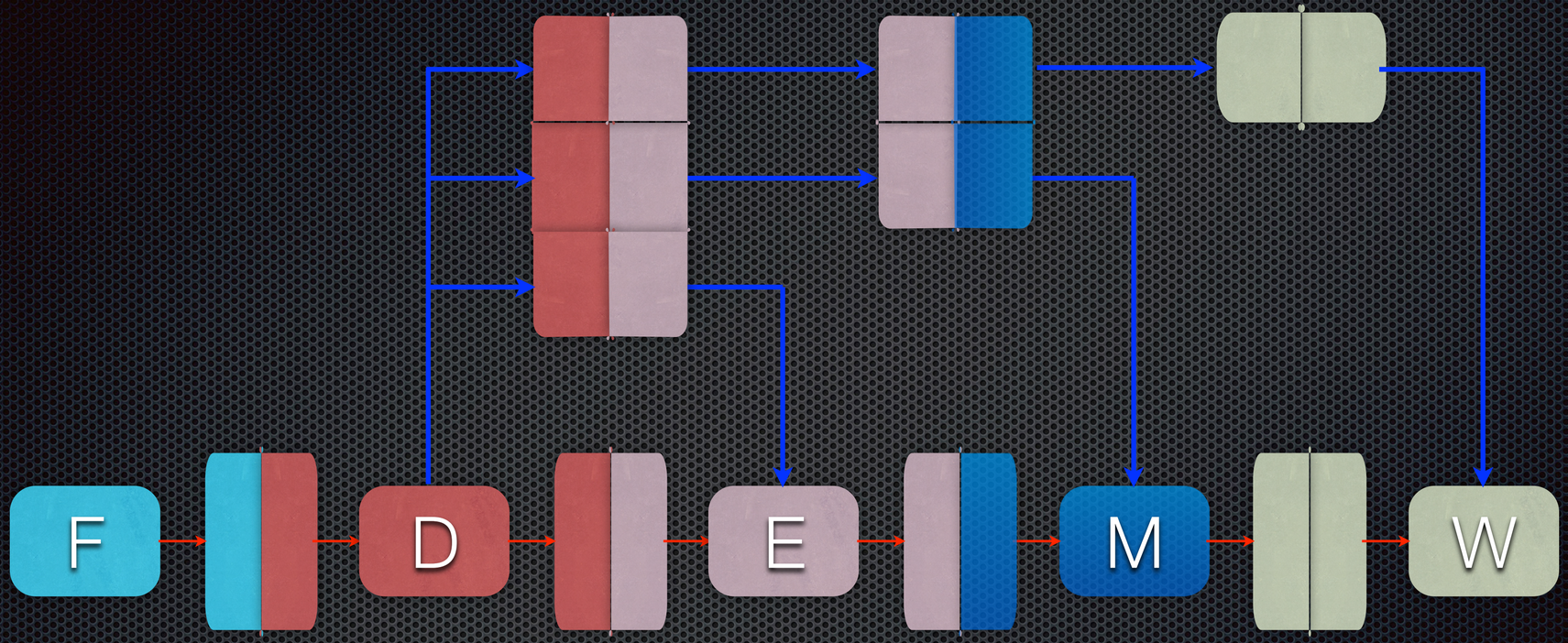

F
D
E
M
W


F
D
E
M
W

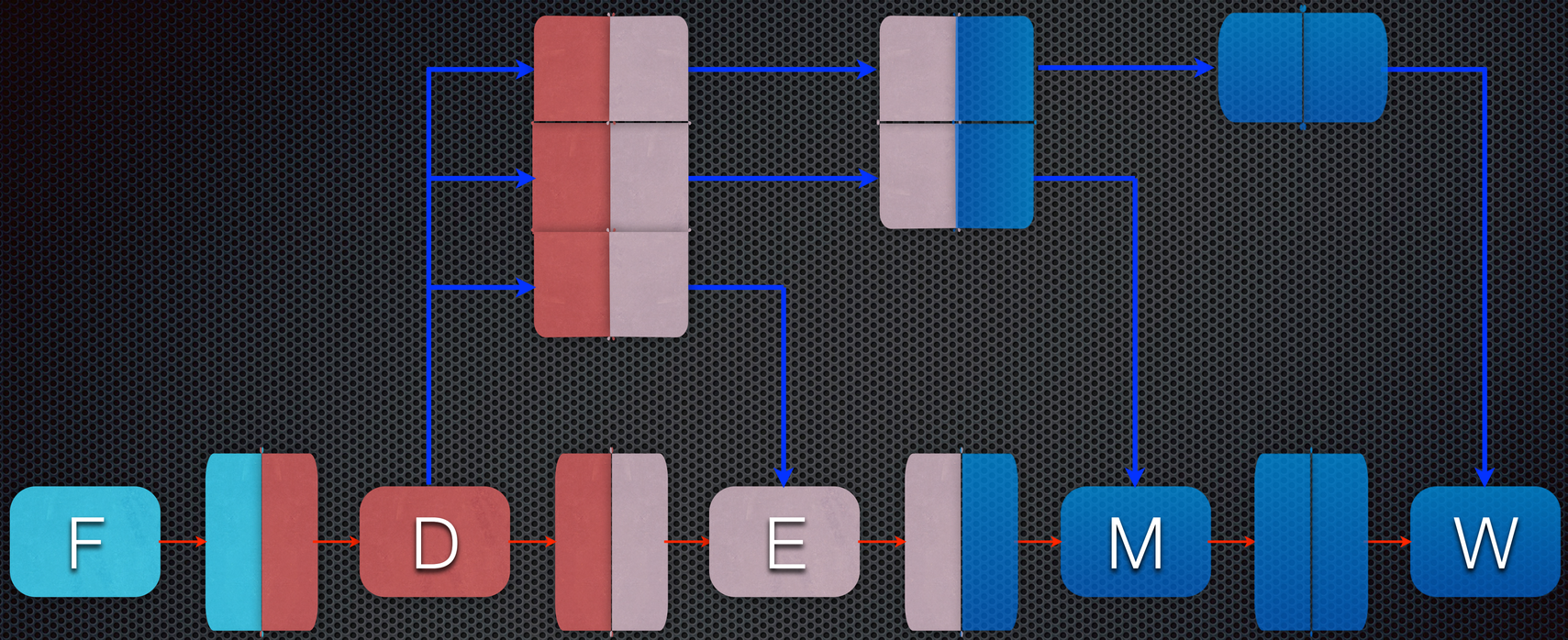

F
D
E
M
W



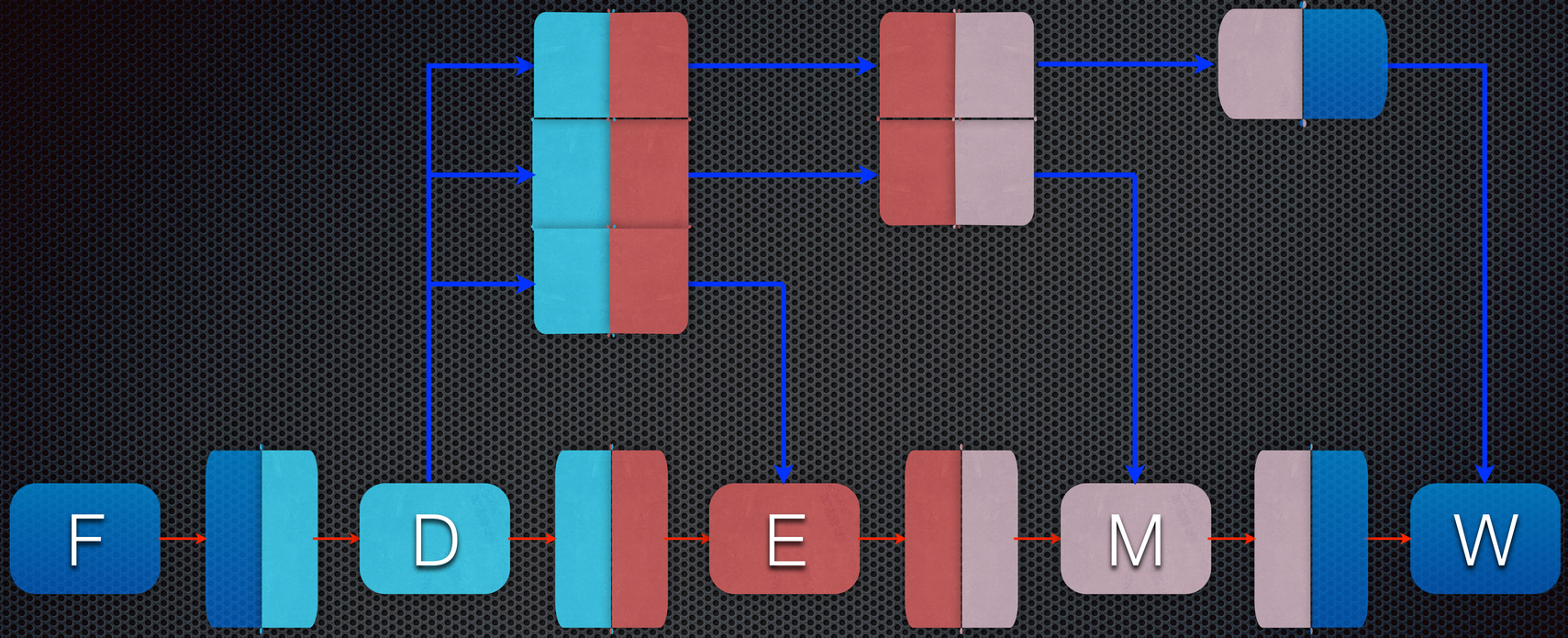
E instruction takes 3 cycles



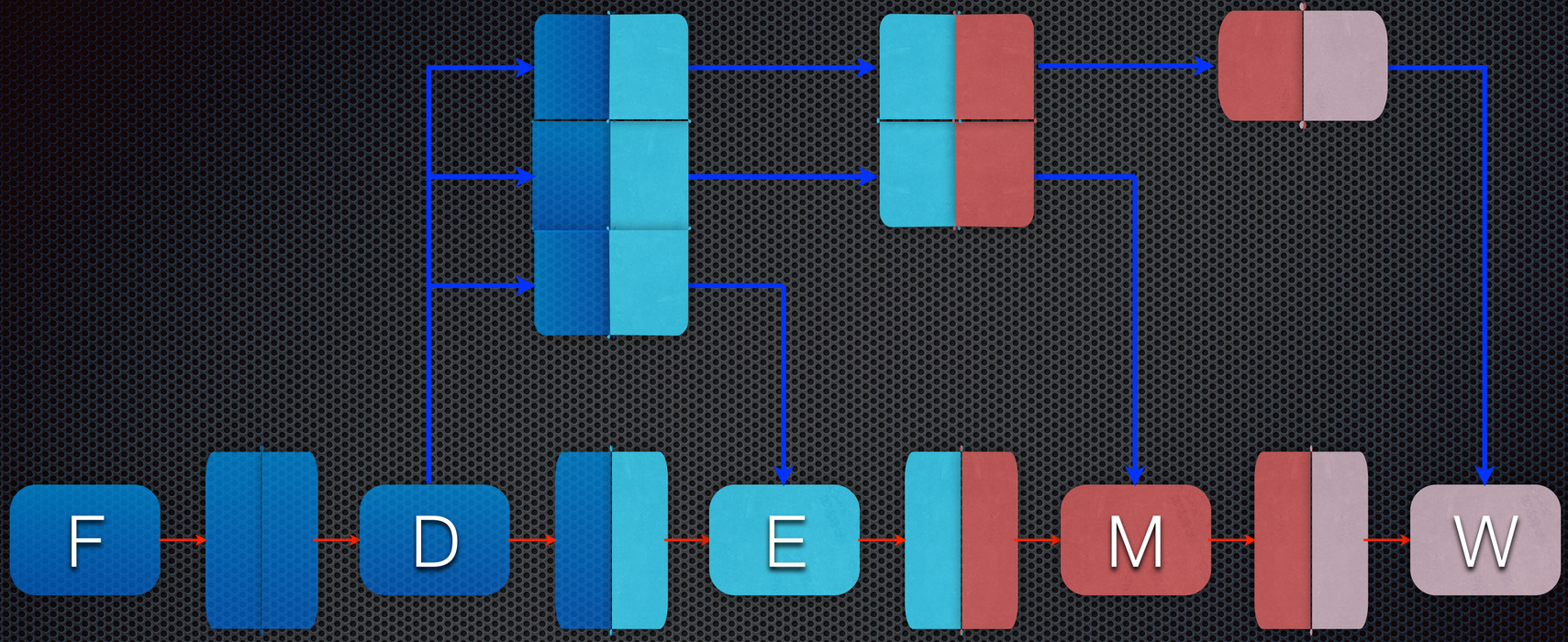
Bubble (NOP) is inserted, and pipe stalls behind E
(structural hazard)

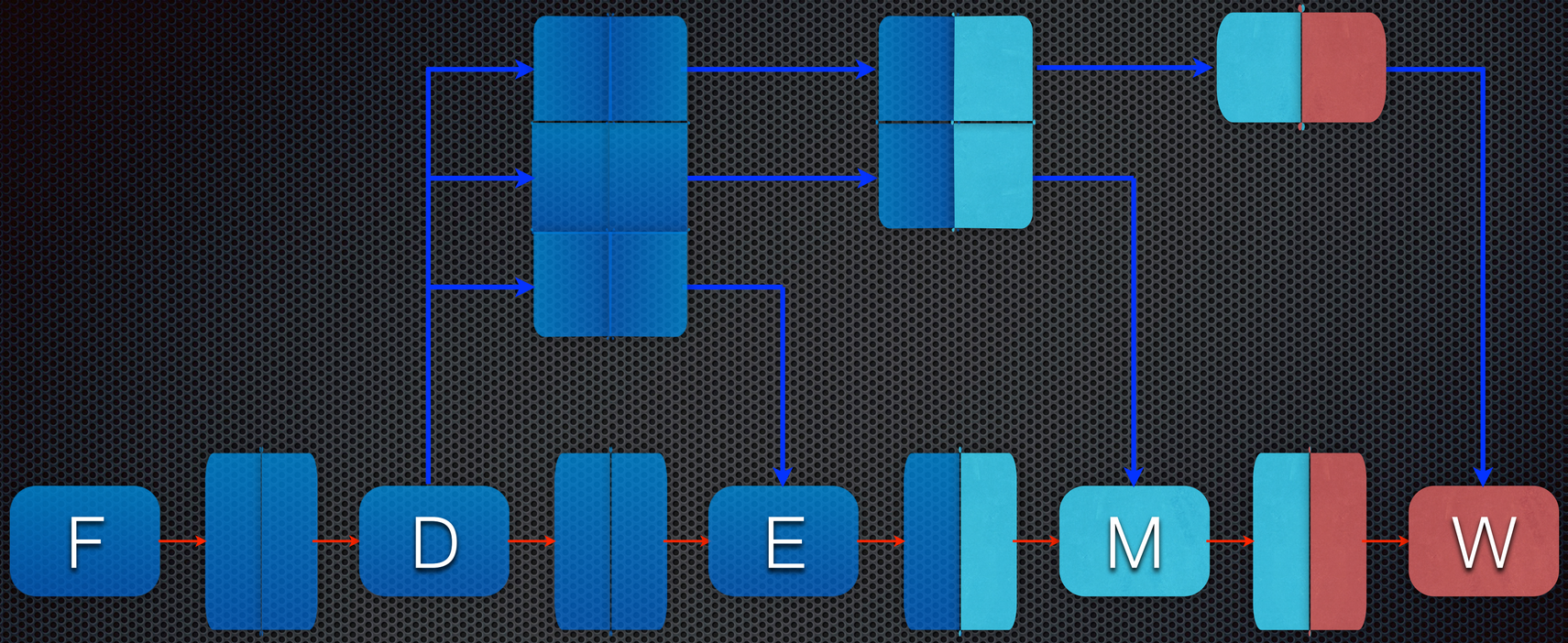


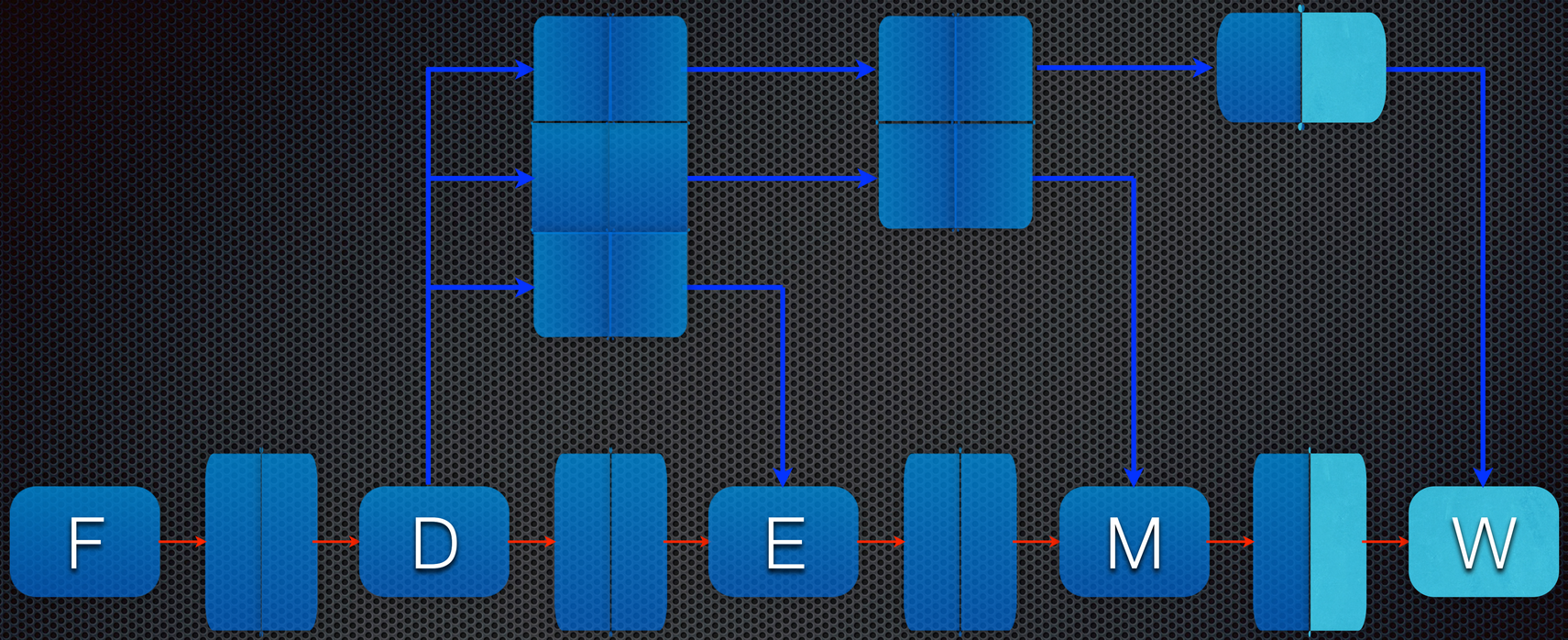
Another bubble is inserted, still stalled

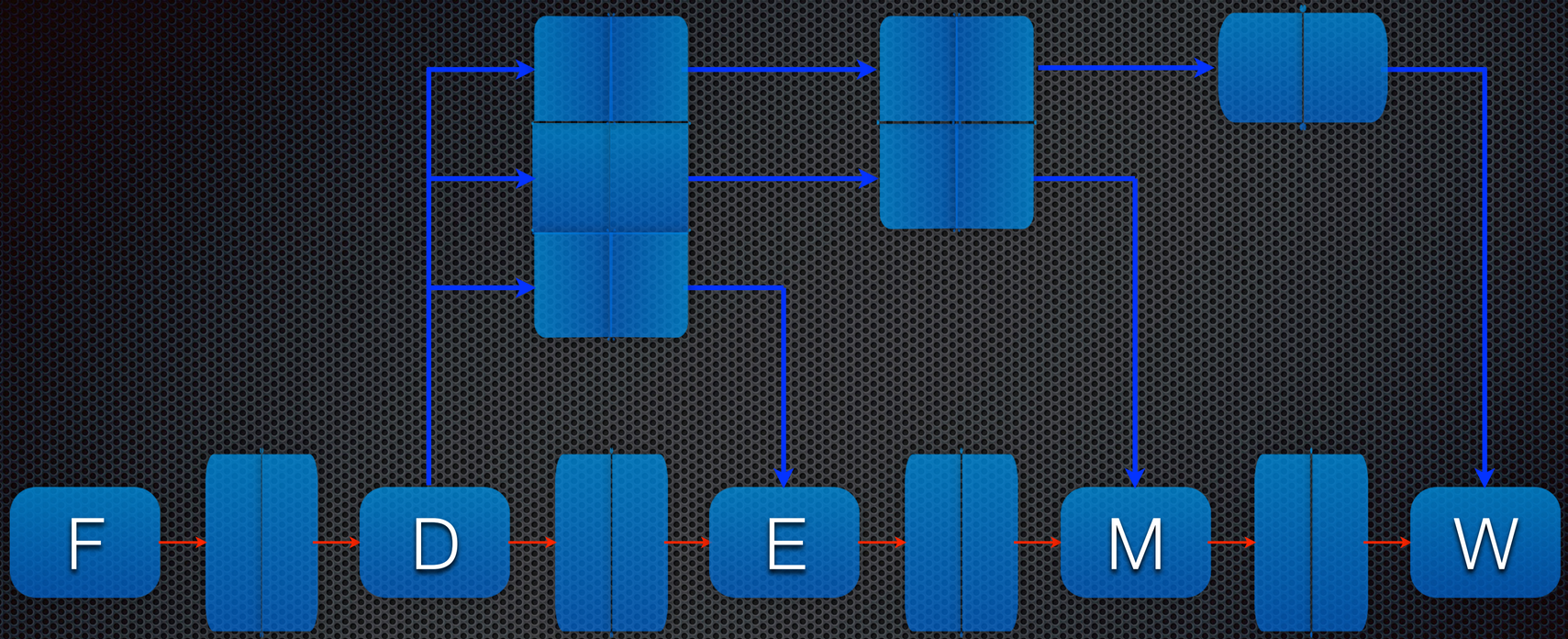


Instruction finishes in E and advances


F
D
E
M
W


F
D
E
M
W


F
D
E
M
W


F
D
E
M
W

Simulation

- When Fetch stage is reached, an instruction is fetched
- Decode examines a fetched instruction and creates an object that has all of the control information (type, opcode, condition, addressing mode, etc.) broken into fields by stage, plus execution state fields (operands, result)
- At each stage, the corresponding control field in the instruction object is examined and selects branches to carry out the operation (may be NOP)
- Using the state of the instruction (e.g., operand values), the stage carries out its task (e.g., computes a result), and updates the state (e.g., puts result in result field)

Simulation

- Non-piped simulation can still use the pipeline
- Just block fetch until the current instruction writes back
- Only one instruction will be in the pipe at a given time
  - Will take at least 5 cycles to pass through (can be delayed)
  - Since the pipe would be clocked roughly 5 times as fast, this will give a reasonable comparison

Pipelined Instruction Flow and Hazards

With Rescheduling to Improve Efficiency

Hazards

- Data Hazard: An instruction depends on a value from another instruction ahead of it in the pipe, which has not yet been computed
- Structural Hazard: A resource is not available (e.g., the FP unit is still busy with a DIV, or a cache miss occurs, or the next stage is stalled)
- Control Hazard: A conditional branch is evaluated in a later stage, and instructions from the wrong path are behind it in the pipeline

Simulating Data Hazards

- Each instruction object has a Target field with the number of the register it will write (if any)
- Decode stalls an instruction if it has an operand that is a Target in any instructions ahead of it in the pipe
- Write Back deletes the instruction object after putting the value in the register, enabling Decode to advance
- Forwarding allows a later stage to send a result to an earlier stage
  - Easy in simple pipes, but harder for longer pipes when handling faults/interrupts

Simulating Structural Hazards

- Back-to-front evaluation simulates blocking due to stalled stages
- Memory (or cache) can make a stage (fetch or memory) wait
- Slow operations (e.g., FDIV, vector logarithm) can make execute wait

Simulating Control Hazards

- If a conditional branch is taken, mark instructions behind it as squashed
- Set PC to new location so next fetch will get correct instruction
- Squashed instructions flow through the pipe as NOPs behind the branch
  - Could just change those instruction to NOPs, but that can be confusing for debugging via the UI, as it shows the instructions in the pipe in single step mode, and they aren’t the same as in the program code

Example

- Let’s look at a longer pipe, to see more realistic effect (MIPS R4400)
- Run 7 instructions through
- No branches so only data and structural hazards (D and S)

MIPS R4400 Pipe

- IF: Instruction First (I-cache fetch part 1)
- IS: Instruction Second (I-cache & decode)
- RF: Register File (Get operands)
- EX: EXecute (Perform operation)
- DF: Data First (D-cache fetch part 1)
- DS: Data Second (D-cache fetch part 2)
- TC: Tag Check (Check for miss)
- WB: Write Back (Save result)

TC can forward a result to EX

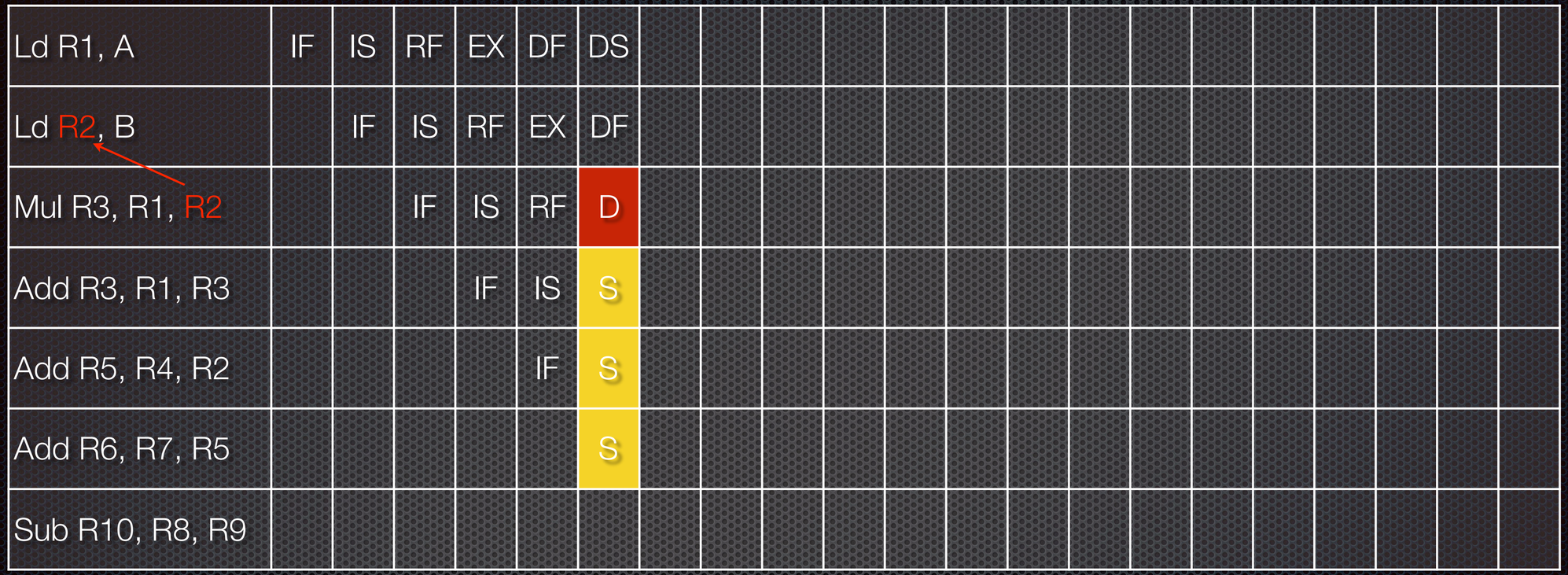
Ld R1, A	IF																			
Ld R2, B																				
Mul R3, R1, R2																				
Add R3, R1, R3																				
Add R5, R4, R2																				
Add R6, R7, R5																				
Sub R10, R8, R9																				

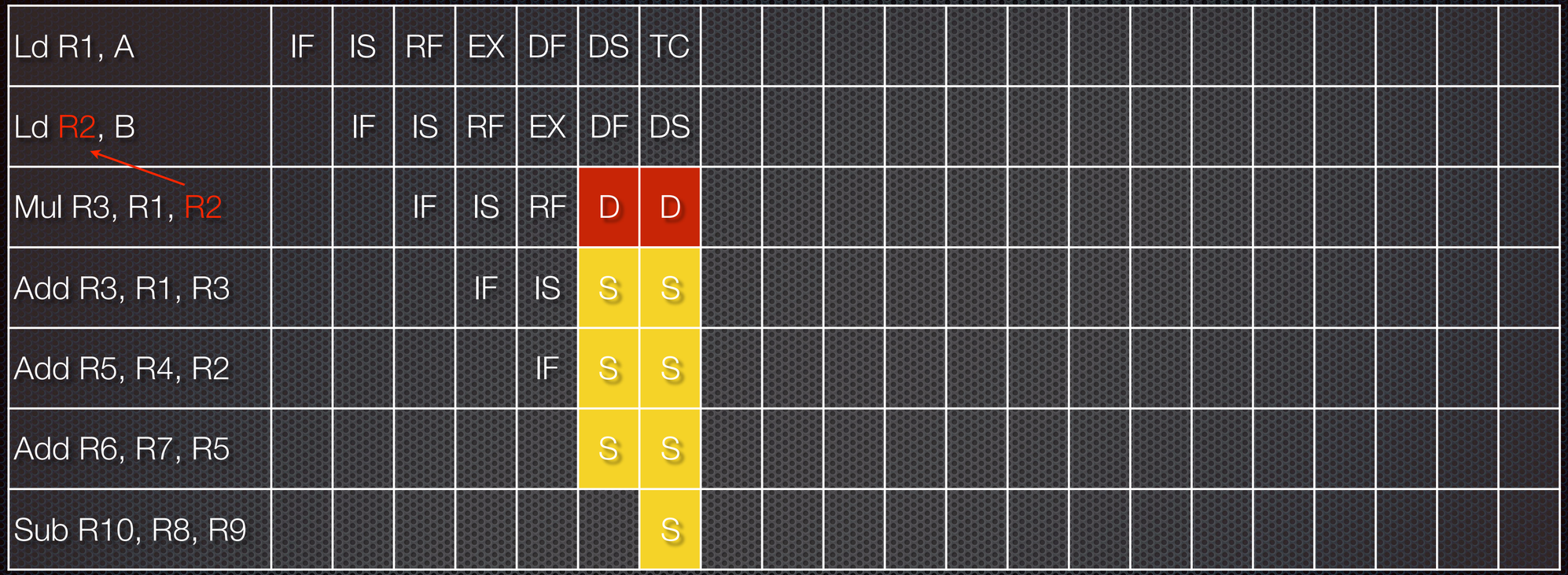
Ld R1, A	IF	IS																		
Ld R2, B		IF																		
Mul R3, R1, R2																				
Add R3, R1, R3																				
Add R5, R4, R2																				
Add R6, R7, R5																				
Sub R10, R8, R9																				

Ld R1, A	IF	IS	RF																
Ld R2, B		IF	IS																
Mul R3, R1, R2			IF																
Add R3, R1, R3																			
Add R5, R4, R2																			
Add R6, R7, R5																			
Sub R10, R8, R9																			

Ld R1, A	IF	IS	RF	EX																
Ld R2, B		IF	IS	RF																
Mul R3, R1, R2			IF	IS																
Add R3, R1, R3				IF																
Add R5, R4, R2																				
Add R6, R7, R5																				
Sub R10, R8, R9																				

Ld R1, A	IF	IS	RF	EX	DF															
Ld R2, B		IF	IS	RF	EX															
Mul R3, R1, R2			IF	IS	RF															
Add R3, R1, R3				IF	IS															
Add R5, R4, R2					IF															
Add R6, R7, R5																				
Sub R10, R8, R9																				


Ld R1, A
IF IS RF EX DF DS
Ld R2, B
IF IS RF EX DF
Mul R3, R1, R2
IF IS RF D
Add R3, R1, R3
IF IS S
Add R5, R4, R2
IF S
Add R6, R7, R5
S
Sub R10, R8, R9


Ld R1, A
IF IS RF EX DF DS TC
Ld R2, B
IF IS RF EX DF DS
Mul R3, R1, R2
IF IS RF D D
Add R3, R1, R3
IF IS S S
Add R5, R4, R2
IF S S
Add R6, R7, R5
S S
Sub R10, R8, R9
S

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB											
Ld R2, B		IF	IS	RF	EX	DF	DS	TC											
Mul R3, R2, R2			IF	IS	RF	D	D	EX											
Add R3, R1, R3				IF	IS	S	S	RF											
Add R5, R4, R2					IF	S	S	IS											
Add R6, R7, R5						S	S	IF											
Sub R10, R8, R9							S	S											

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF											
Add R3, R1, R3				IF	IS	S	S	RF	D											
Add R5, R4, R2					IF	S	S	IS	S											
Add R6, R7, R5						S	S	IF	S											
Sub R10, R8, R9							S	S	S											

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS										
Add R3, R1, R3				IF	IS	S	S	RF	D	D										
Add R5, R4, R2					IF	S	S	IS	S	S										
Add R6, R7, R5						S	S	IF	S	S										
Sub R10, R8, R9							S	S	S	S										

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX									
Add R5, R4, R2					IF	S	S	IS	S	S	RF									
Add R6, R7, R5						S	S	IF	S	S	IS									
Sub R10, R8, R9							S	S	S	S	IF									

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB								
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF								
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX								
Add R6, R7, R5						S	S	IF	S	S	IS	RF								
Sub R10, R8, R9							S	S	S	S	IF	IS								

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB								
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS							
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF							
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D							
Sub R10, R8, R9							S	S	S	S	IF	IS	S							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB								
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS						
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D						
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S						

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB								
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB					
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX					
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF					

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX	DF					
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF	EX					

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS				
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF	EX	DF				

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC			
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF	EX	DF	DS			

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB		
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF	EX	DF	DS	TC		

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Mul R3, R2, R2			IF	IS	RF	D	D	EX	DF	DS	TC	WB									
Add R3, R1, R3				IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB						
Add R5, R4, R2					IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB					
Add R6, R7, R5						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB		
Sub R10, R8, R9							S	S	S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB	

Optimal 14 / Actual 20 = 70% efficient

Reschedule

- Let’s see what happens if we reorder ops
  - Move the Sub up, to follow loads
  - Reverse order of first two Add ops

Ld R1, A	IF																			
Ld R2, B																				
Sub R10, R8, R9																				
Mul R3, R1, R2																				
Add R5, R4, R2																				
Add R3, R1, R3																				
Add R6, R7, R5																				

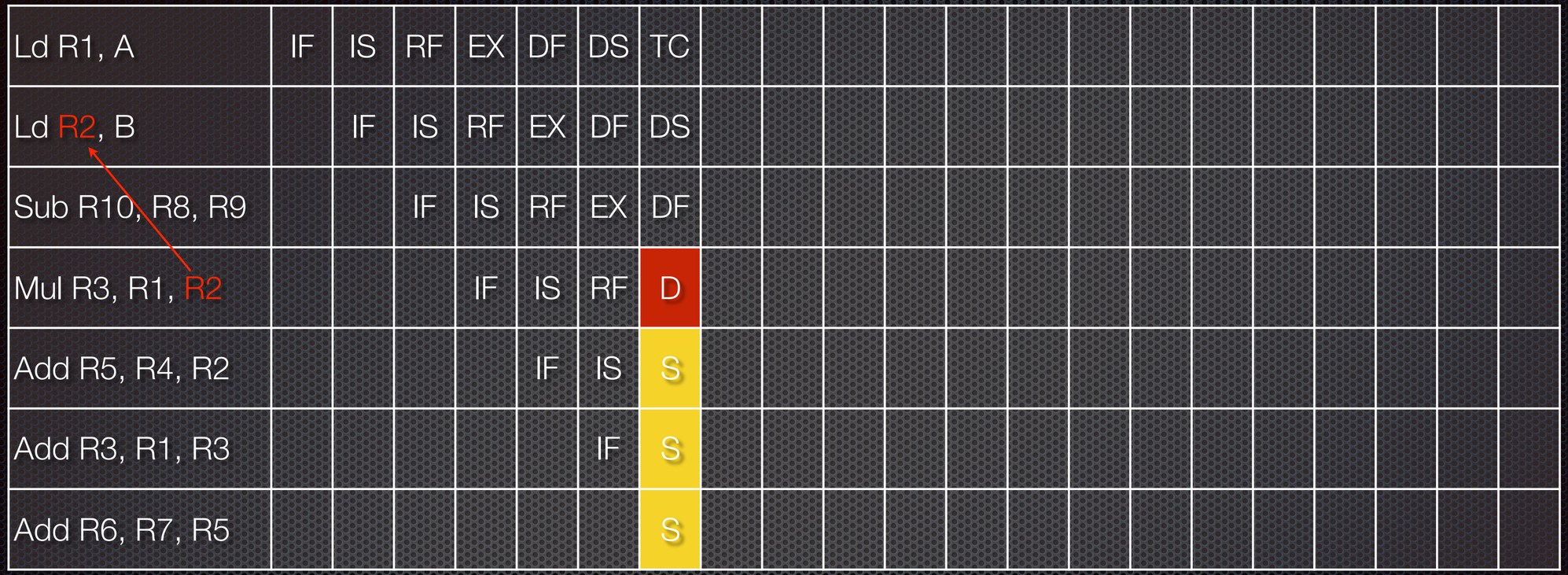
Ld R1, A	IF	IS																		
Ld R2, B		IF																		
Sub R10, R8, R9																				
Mul R3, R1, R2																				
Add R5, R4, R2																				
Add R3, R1, R3																				
Add R6, R7, R5																				

Ld R1, A	IF	IS	RF																	
Ld R2, B		IF	IS																	
Sub R10, R8, R9			IF																	
Mul R3, R1, R2																				
Add R5, R4, R2																				
Add R3, R1, R3																				
Add R6, R7, R5																				

Ld R1, A	IF	IS	RF	EX																
Ld R2, B		IF	IS	RF																
Sub R10, R8, R9			IF	IS																
Mul R3, R1, R2				IF																
Add R5, R4, R2																				
Add R3, R1, R3																				
Add R6, R7, R5																				

Ld R1, A	IF	IS	RF	EX	DF															
Ld R2, B		IF	IS	RF	EX															
Sub R10, R8, R9			IF	IS	RF															
Mul R3, R1, R2				IF	IS															
Add R5, R4, R2					IF															
Add R3, R1, R3																				
Add R6, R7, R5																				

Ld R1, A	IF	IS	RF	EX	DF	DS														
Ld R2, B		IF	IS	RF	EX	DF														
Sub R10, R8, R9			IF	IS	RF	EX														
Mul R3, R1, R2				IF	IS	RF														
Add R5, R4, R2					IF	IS														
Add R3, R1, R3						IF														
Add R6, R7, R5																				


Ld R1, A
IF IS RF EX DF DS TC
Ld R2, B
IF IS RF EX DF DS
Sub R10, R8, R9
IF IS RF EX DF
Mul R3, R1, R2
IF IS RF D
Add R5, R4, R2
IF IS S
Add R3, R1, R3
IF S
Add R6, R7, R5
S

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB											
Ld R2, B		IF	IS	RF	EX	DF	DS	TC											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS											
Mul R3, R1, R2				IF	IS	RF	D	EX											
Add R5, R4, R2					IF	IS	S	RF											
Add R3, R1, R3						IF	S	IS											
Add R6, R7, R5							S	IF											

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC											
Mul R3, R1, R2				IF	IS	RF	D	EX	DF											
Add R5, R4, R2					IF	IS	S	RF	EX											
Add R3, R1, R3						IF	S	IS	RF											
Add R6, R7, R5							S	IF	IS											

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS										
Add R5, R4, R2					IF	IS	S	RF	EX	DF										
Add R3, R1, R3						IF	S	IS	RF	D										
Add R6, R7, R5							S	IF	IS	S										

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC									
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS									
Add R3, R1, R3						IF	S	IS	RF	D	EX									
Add R6, R7, R5							S	IF	IS	S	RF									

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC	WB								
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS	TC								
Add R3, R1, R3						IF	S	IS	RF	D	EX	DF								
Add R6, R7, R5							S	IF	IS	S	RF	EX								

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC	WB								
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS	TC	WB							
Add R3, R1, R3						IF	S	IS	RF	D	EX	DF	DS							
Add R6, R7, R5							S	IF	IS	S	RF	EX	DF							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB													
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB												
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB											
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC	WB									
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS	TC	WB								
Add R3, R1, R3						IF	S	IS	RF	D	EX	DF	DS	TC							
Add R6, R7, R5							S	IF	IS	S	RF	EX	DF	DS							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC	WB								
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS	TC	WB							
Add R3, R1, R3						IF	S	IS	RF	D	EX	DF	DS	TC	WB					
Add R6, R7, R5							S	IF	IS	S	RF	EX	DF	DS	TC					

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB												
Ld R2, B		IF	IS	RF	EX	DF	DS	TC	WB											
Sub R10, R8, R9			IF	IS	RF	EX	DF	DS	TC	WB										
Mul R3, R1, R2				IF	IS	RF	D	EX	DF	DS	TC	WB								
Add R5, R4, R2					IF	IS	S	RF	EX	DF	DS	TC	WB							
Add R3, R1, R3						IF	S	IS	RF	D	EX	DF	DS	TC	WB					
Add R6, R7, R5							S	IF	IS	S	RF	EX	DF	DS	TC	WB				

Optimal 14 / Actual 16 = 88% efficient

Conclusion

- Instruction rescheduling significantly boosts efficiency
- NO HARDWARE CHANGE REQUIRED!
- Fairly simple compiler analysis
  - Done in hardware on modern processors
- Only possible for separable operations

Exercise: Determine the efficiency of the following, and then reschedule the instructions to improve the efficiency

Ld R1, A																							
Add R5, R4, R1																							
Ld R2, B																							
Add R7, R6, R2																							
Ld R3, C																							
Add R8, R6, R3																							
Add R9, R7, R8																							
Add R10, R5, R2																							

Ld R1, A	IF																					
Add R5, R4, R1																						
Ld R2, B																						
Add R7, R6, R2																						
Ld R3, C																						
Add R8, R6, R3																						
Add R9, R7, R8																						
Add R10, R5, R2																						

Ld R1, A	IF	IS																				
Add R5, R4, R1		IF																				
Ld R2, B																						
Add R7, R6, R2																						
Ld R3, C																						
Add R8, R6, R3																						
Add R9, R7, R8																						
Add R10, R5, R2																						

Ld R1, A	IF	IS	RF																				
Add R5, R4, R1		IF	IS																				
Ld R2, B			IF																				
Add R7, R6, R2																							
Ld R3, C																							
Add R8, R6, R3																							
Add R9, R7, R8																							
Add R10, R5, R2																							

Ld R1, A	IF	IS	RF	EX																			
Add R5, R4, R1		IF	IS	RF																			
Ld R2, B			IF	IS																			
Add R7, R6, R2				IF																			
Ld R3, C																							
Add R8, R6, R3																							
Add R9, R7, R8																							
Add R10, R5, R2																							

Ld R1, A	IF	IS	RF	EX	DF																	
Add R5, R4, R1		IF	IS	RF	D																	
Ld R2, B			IF	IS	S																	
Add R7, R6, R2				IF	S																	
Ld R3, C					S																	
Add R8, R6, R3																						
Add R9, R7, R8																						
Add R10, R5, R2																						

Ld R1, A	IF	IS	RF	EX	DF	DS																	
Add R5, R4, R1		IF	IS	RF	D	D																	
Ld R2, B			IF	IS	S	S																	
Add R7, R6, R2				IF	S	S																	
Ld R3, C					S	S																	
Add R8, R6, R3						S																	
Add R9, R7, R8																							
Add R10, R5, R2																							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC																
Add R5, R4, R1		IF	IS	RF	D	D	EX																
Ld R2, B			IF	IS	S	S	RF																
Add R7, R6, R2				IF	S	S	IS																
Ld R3, C					S	S	IF																
Add R8, R6, R3						S	S																
Add R9, R7, R8							S																
Add R10, R5, R2																							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF															
Ld R2, B			IF	IS	S	S	RF	EX															
Add R7, R6, R2				IF	S	S	IS	RF															
Ld R3, C					S	S	IF	IS															
Add R8, R6, R3						S	S	IF															
Add R9, R7, R8							S	S															
Add R10, R5, R2								S															

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS														
Ld R2, B			IF	IS	S	S	RF	EX	DF														
Add R7, R6, R2				IF	S	S	IS	RF	D														
Ld R3, C					S	S	IF	IS	S														
Add R8, R6, R3						S	S	IF	S														
Add R9, R7, R8							S	S	S														
Add R10, R5, R2								S	S														

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC													
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS													
Add R7, R6, R2				IF	S	S	IS	RF	D	D													
Ld R3, C					S	S	IF	IS	S	S													
Add R8, R6, R3						S	S	IF	S	S													
Add R9, R7, R8							S	S	S	S													
Add R10, R5, R2								S	S	S													

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC												
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX												
Ld R3, C					S	S	IF	IS	S	S	RF												
Add R8, R6, R3						S	S	IF	S	S	IS												
Add R9, R7, R8							S	S	S	S	IF												
Add R10, R5, R2								S	S	S	S												

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF											
Ld R3, C					S	S	IF	IS	S	S	RF	EX											
Add R8, R6, R3						S	S	IF	S	S	IS	RF											
Add R9, R7, R8							S	S	S	S	IF	IS											
Add R10, R5, R2								S	S	S	S	IF											

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS										
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF										
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D										
Add R9, R7, R8							S	S	S	S	IF	IS	S										
Add R10, R5, R2								S	S	S	S	IF	S										

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC									
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS									
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D									
Add R9, R7, R8							S	S	S	S	IF	IS	S	S									
Add R10, R5, R2								S	S	S	S	IF	S	S									

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC								
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX								
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF								
Add R10, R5, R2								S	S	S	S	IF	S	S	IS								

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF							
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D							
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S							

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS						
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D						
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S						

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC					
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX					
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF					

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB				
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX	DF				
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF	EX				

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB				
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX	DF	DS			
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF	EX	DF			

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB				
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX	DF	DS	TC		
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF	EX	DF	DS		

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB				
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB	
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF	EX	DF	DS	TC	

Ld R1, A	IF	IS	RF	EX	DF	DS	TC	WB															
Add R5, R4, R1		IF	IS	RF	D	D	EX	DF	DS	TC	WB												
Ld R2, B			IF	IS	S	S	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2				IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB								
Ld R3, C					S	S	IF	IS	S	S	RF	EX	DF	DS	TC	WB							
Add R8, R6, R3						S	S	IF	S	S	IS	RF	D	D	EX	DF	DS	TC	WB				
Add R9, R7, R8							S	S	S	S	IF	IS	S	S	RF	D	D	EX	DF	DS	TC	WB	
Add R10, R5, R2								S	S	S	S	IF	S	S	IS	S	S	RF	EX	DF	DS	TC	WB

Optimal 15 / Actual 23 = 65% efficient

Rescheduling

Ld R3, C	IF	IS	RF	EX	DF	DS	TC	WB															
Ld R1, A		IF	IS	RF	EX	DF	DS	TC	WB														
Ld R2, B			IF	IS	RF	EX	DF	DS	TC	WB													
Add R8, R6, R3				IF	IS	RF	EX	DF	DS	TC	WB												
Add R5, R4, R1					IF	IS	RF	EX	DF	DS	TC	WB											
Add R7, R6, R2						IF	IS	RF	EX	DF	DS	TC	WB										
Add R10, R5, R2							IF	IS	RF	D	EX	DF	DS	TC	WB								
Add R9, R7, R8								IF	IS	S	RF	EX	DF	DS	TC	WB							

Optimal 15 / Actual 16 = 94% efficient

Simulating Forwarding (5-stage pipe)

- Alternatively, as an instruction enters Write Back, its result can be copied to the operand field of the instruction in Decode
- Its value will still be written to the register on the next cycle
- Once the instruction in Decode has all of its operand values, it can advance
- If the forwarded value is the last one it needs, a cycle is saved over waiting for the result to be put in the register