BCS-29
Advanced Computer Architecture

Pipelined Processing

ARITHMETIC PIPELINE DESIGN

Other ILP Architectures

Pipeline Multiplication

Multiplication

- Consider as an example the multiplication of two 8-bit integers, A X B = P, where P is the 16-bit product.
- This fixed-point multiplication can be written as the summation of eight partial products as shown below: P = A X B = P0 + P1 + P2 + …+ P7.

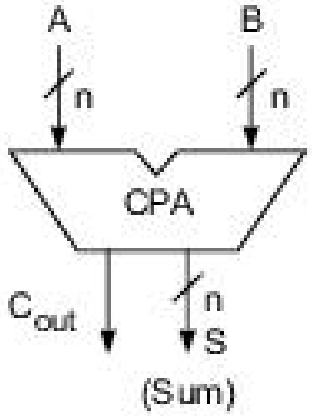
								1	0	1	1	0	1	0	1	=	A
							×)	1	0	0	1	0	0	1	1	=	B
								1	0	1	1	0	1	0	1	=	P_0
							1	0	1	1	0	1	0	1	0	=	P_1
						0	0	0	0	0	0	0	0	0	0	=	P_2
					0	0	0	0	0	0	0	0	0	0	0	=	P_3
				1	0	1	1	0	1	0	1	0	0	0	0	=	P_4
			0	0	0	0	0	0	0	0	0	0	0	0	0	=	P_5
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	=	P_6
+)	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	=	P_7
0	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	=	P

Pipeline Multiplication

Carry Propagate Adder

e.g. n=4

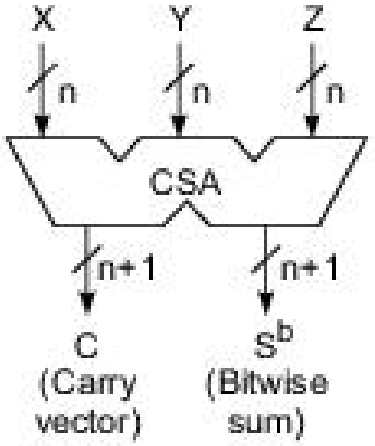
```
      A =   1 0 1 1
+)    B =   0 1 1 1
-------------------
      S = 1 0 0 1 0 = A + B
```



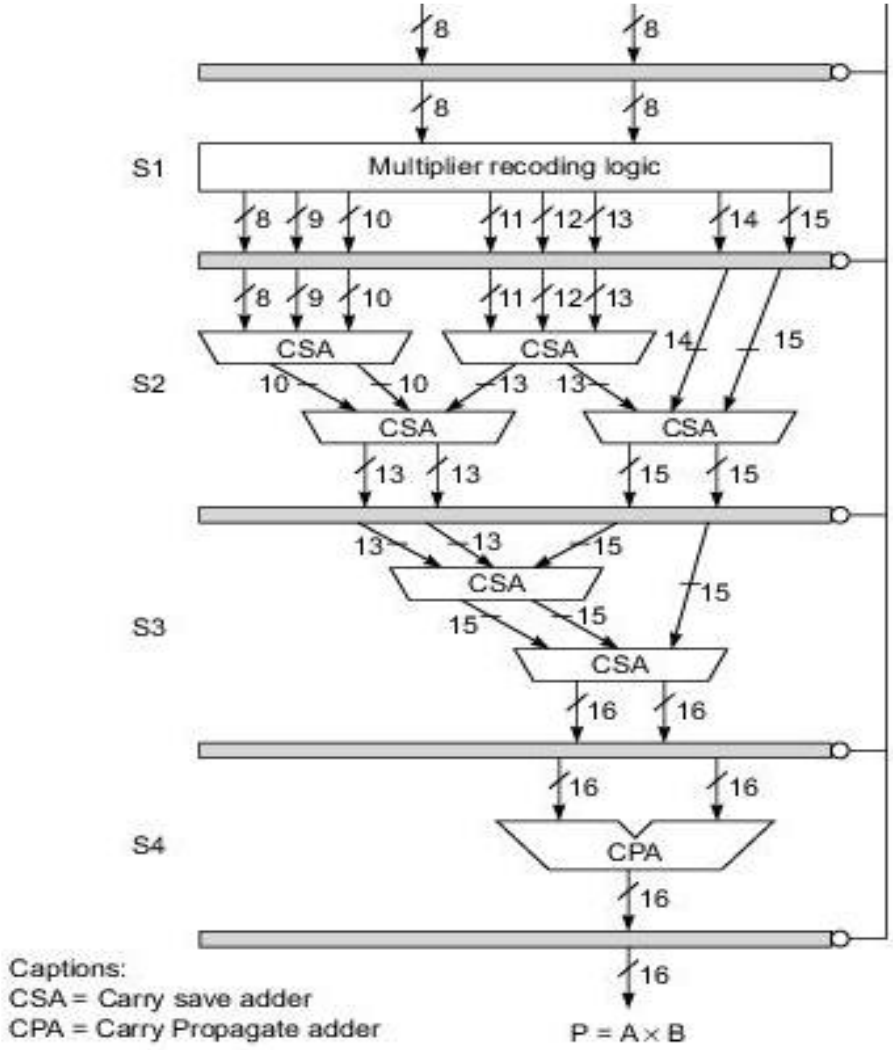
Carry Save Adder

e.g. n=4

```
      X =   0 0 1 0 1 1
      Y =   0 1 0 1 0 1
⊕     Z =   1 1 1 1 0 1
-------------------------
      S^b = 0 1 0 0 0 1 1
+)    C = 0 1 1 1 0 1 0
-------------------------
      S = 1 0 1 1 1 1 1 = S^b+C = X+Y+Z
```

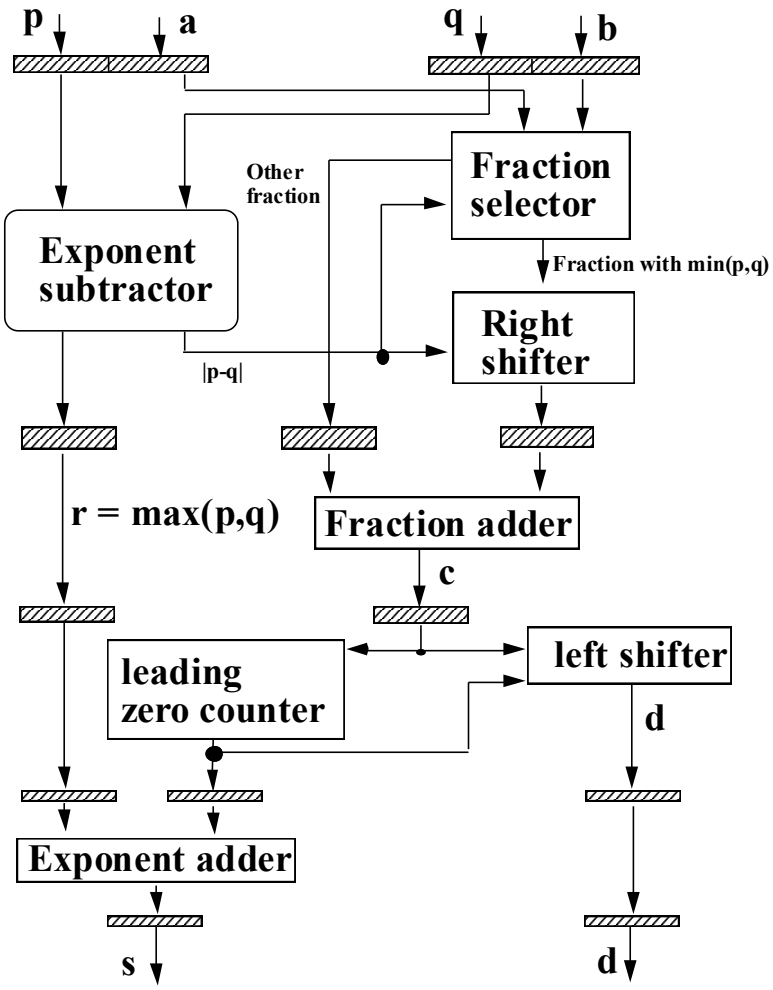


Pipeline Multiplication

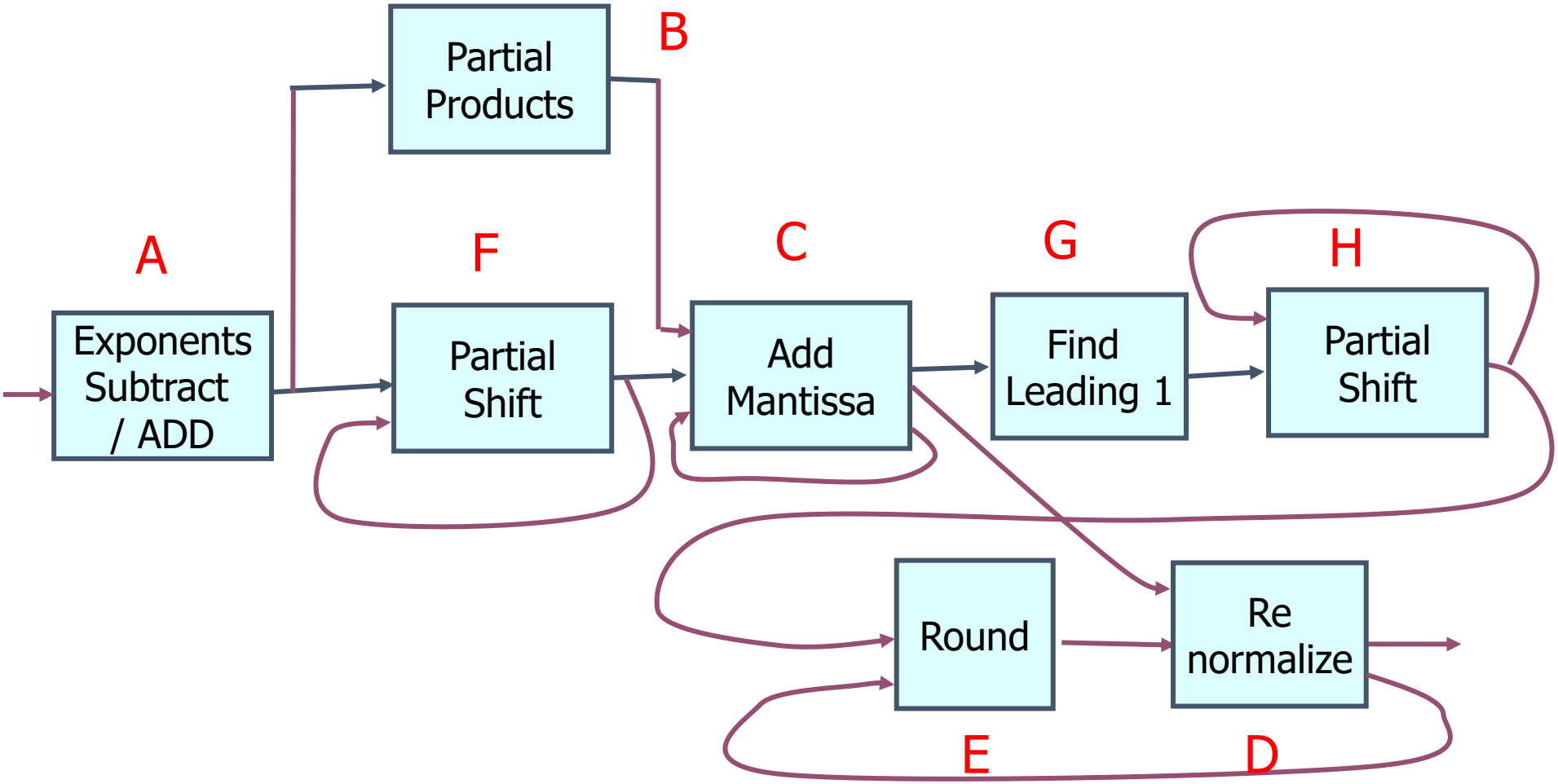


Floating Point Addition

- Linear pipeline with four functional stages. Inputs are two normalised floating-point numbers
  a*2^p and **b*2^q**
- Output is a normalised floating-point number **d* 2^s** which is the sum of the two inputs.
- The hardware units other than the latches can all be implemented using combinational logic.
- If time delay of interface latches is 10ns and if the time delays of the four stages are 60, 50, 90 and 80ns, respectively, then cycle time of pipeline can be chosen to be 100ns.



Combined Adder and Multiplier



Reservation Table for Multiplication

	1	2	3	4	5	6	7
A	X						
B		X	X				
C			X	X			
D					X		X
E						X	
F							
G							
H							

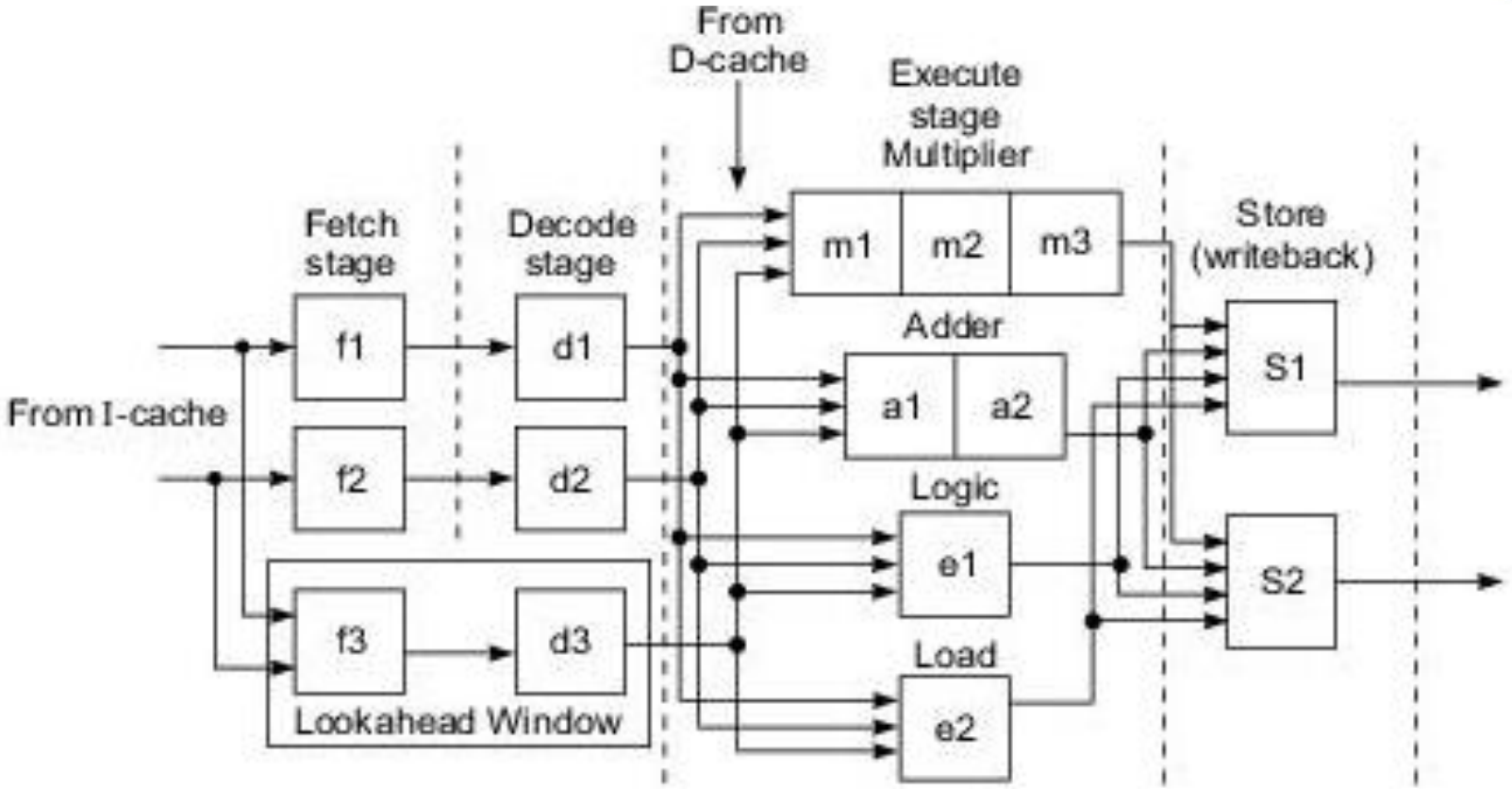
Reservation Table for Addition

	1	2	3	4	5	6	7	8	9
A	Y								
B									
C				Y					
D									Y
E								Y	
F		Y	Y						
G					Y				
H						Y	Y		

Superscalar Architectures

- Superscalar processors attempt to issue multiple instructions per cycle
  - However, essential dependencies are specified by sequential ordering so operations must be processed in sequential order
  - This proves to be a performance bottleneck that is very expensive to overcome
- Program contains no explicit information regarding dependencies that exist between instructions
- Dependencies between instructions must be determined by the hardware
  - It is only necessary to determine dependencies with sequentially preceding instructions that have been issued but not yet completed
- Compiler may re-order instructions to facilitate the hardware's task of extracting parallelism

Superscalar Architectures



Superscalar Architectures

IF	ID	EX	WB		
IF	ID	EX	WB		
IF	ID	EX	WB		
	IF	ID	EX	WB	
	IF	ID	EX	WB	
	IF	ID	EX	WB	
		IF	ID	EX	WB
		IF	ID	EX	WB
		IF	ID	EX	WB

- Superscalar:
  - Issue parallelism = IP = n inst / cycle
  - Operation latency = OP = 1 cycle
  - Peak IPC = n instr / cycle (n x speedup?)

Superscalar Performance

Estimate the ideal execution time of N independent instructions through the pipeline.

- The time required by the scalar base machine(Single Pipeline) is

$$T(1,1) = k + N - 1 \text{ (base cycles)}$$

- The ideal execution time required by an m-issue superscalar machine is

$$T(m, 1) = k + (N - m)/m \text{ (base cycles)}$$

- The ideal speedup of the superscalar machine over the base machine is

$$S(m,1) = T(1,1)/T(m,1)$$

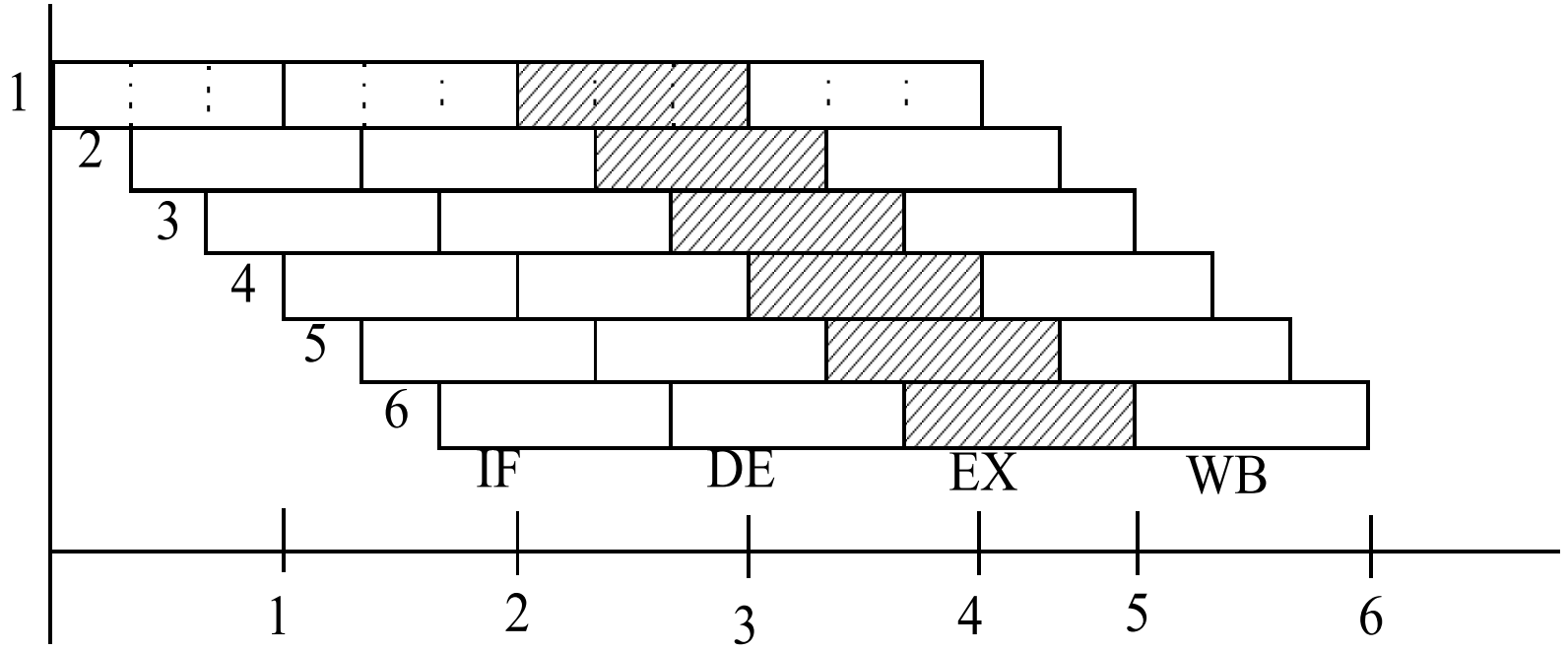
$$= (k + N - 1)/(k + (N - m)/m)$$

$$= m(k + N - 1)/(N + m(K-1))$$

Other ILP Architectures

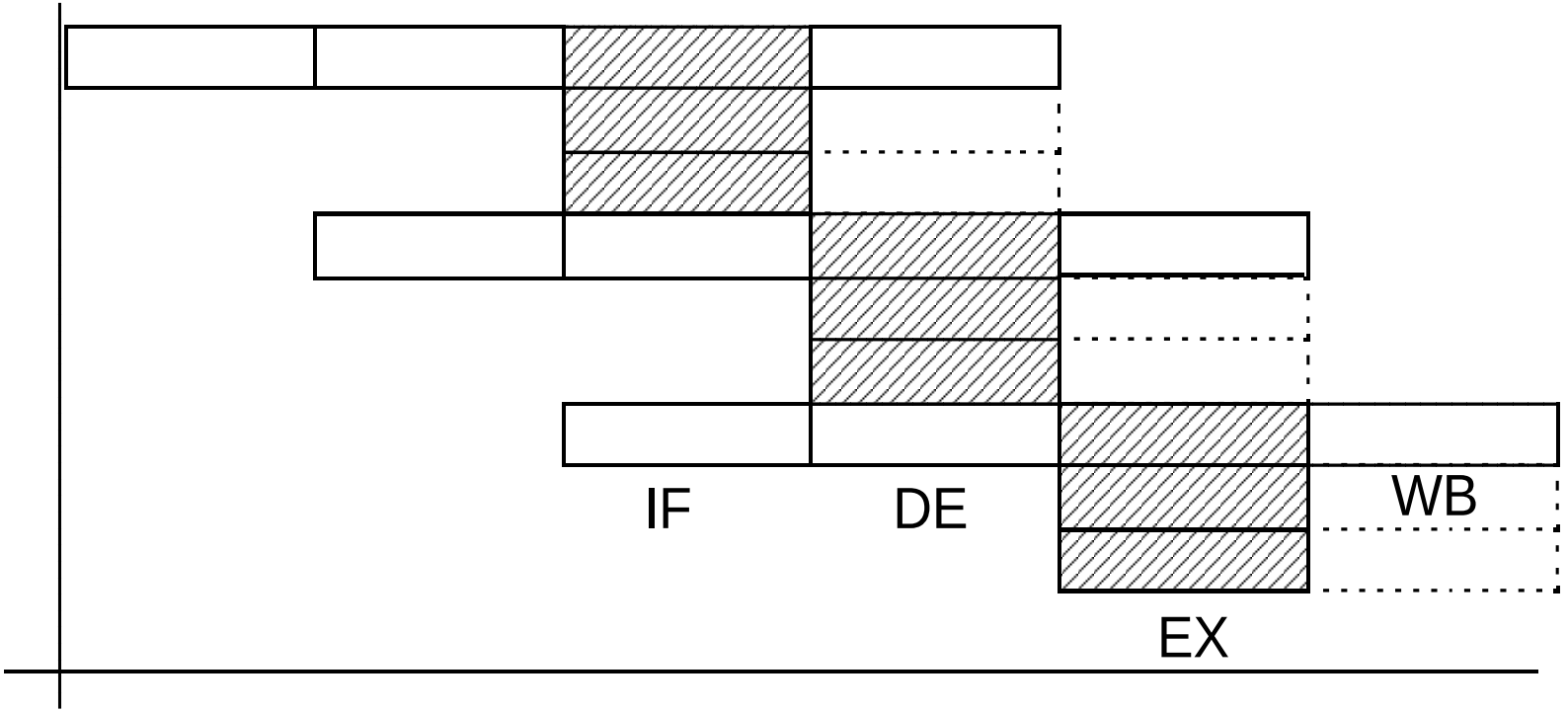
Superpipelined Architecture:

- cycle time = 1/m of baseline
- Issue parallelism = IP = 1 inst / minor cycle
- Operation latency = OP = m minor cycles
- Peak IPC = m instr / major cycle (m x speedup?)



Other ILP Architectures

- VLIW: Very Long Instruction Word
  - Issue parallelism = IP = n inst / cycle
  - Operation latency = OP = 1 cycle
  - Peak IPC = n instr / cycle = 1 VLIW / cycle



Other ILP Architectures

- Superpipelined-Superscalar
  - Issue parallelism = IP = n inst / minor cycle
  - Operation latency = OP = m minor cycles
  - Peak IPC = n x m instr / major cycle