

SSC0510

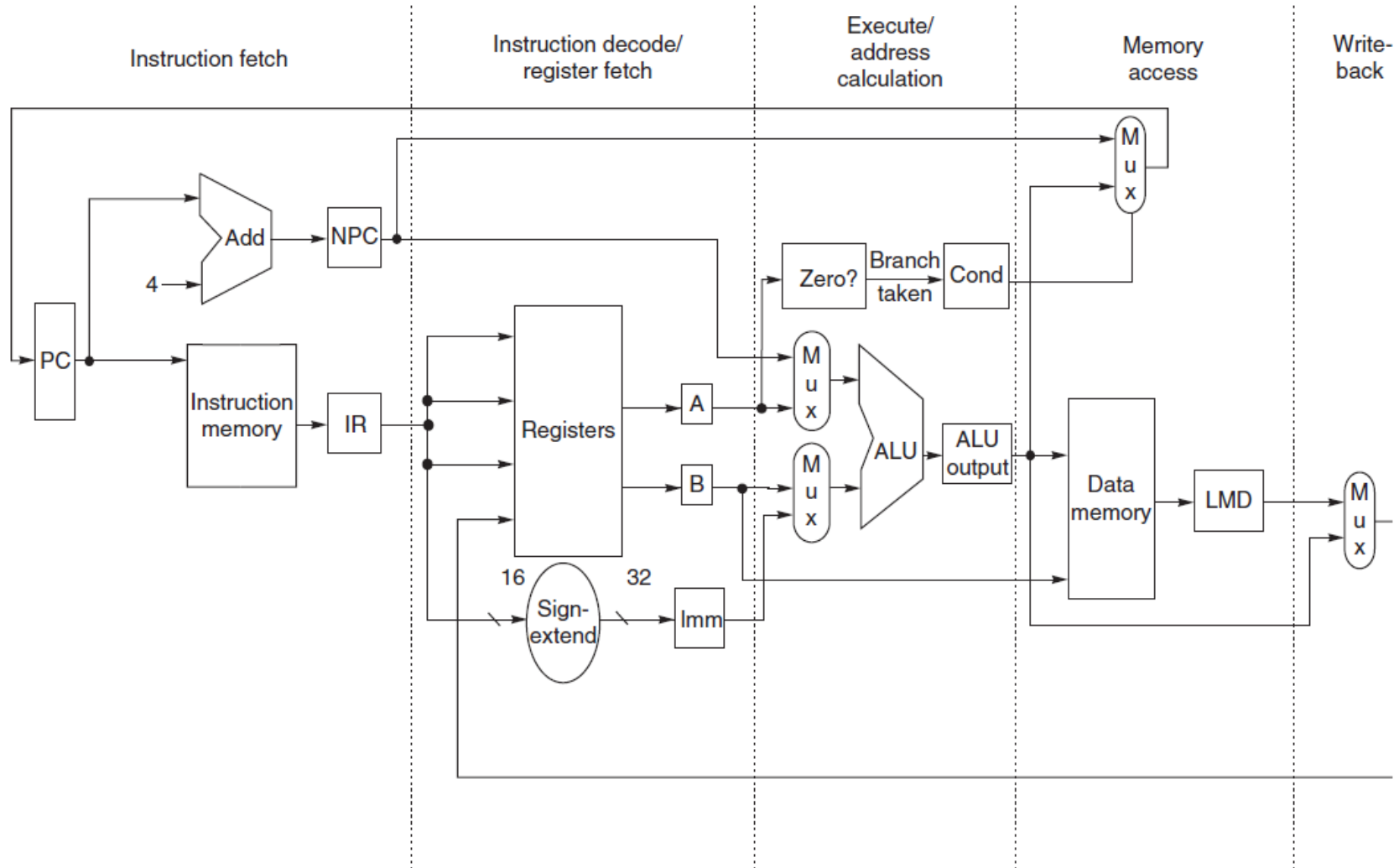
Arquitetura de Computadores

4ª Aula – Pipeline

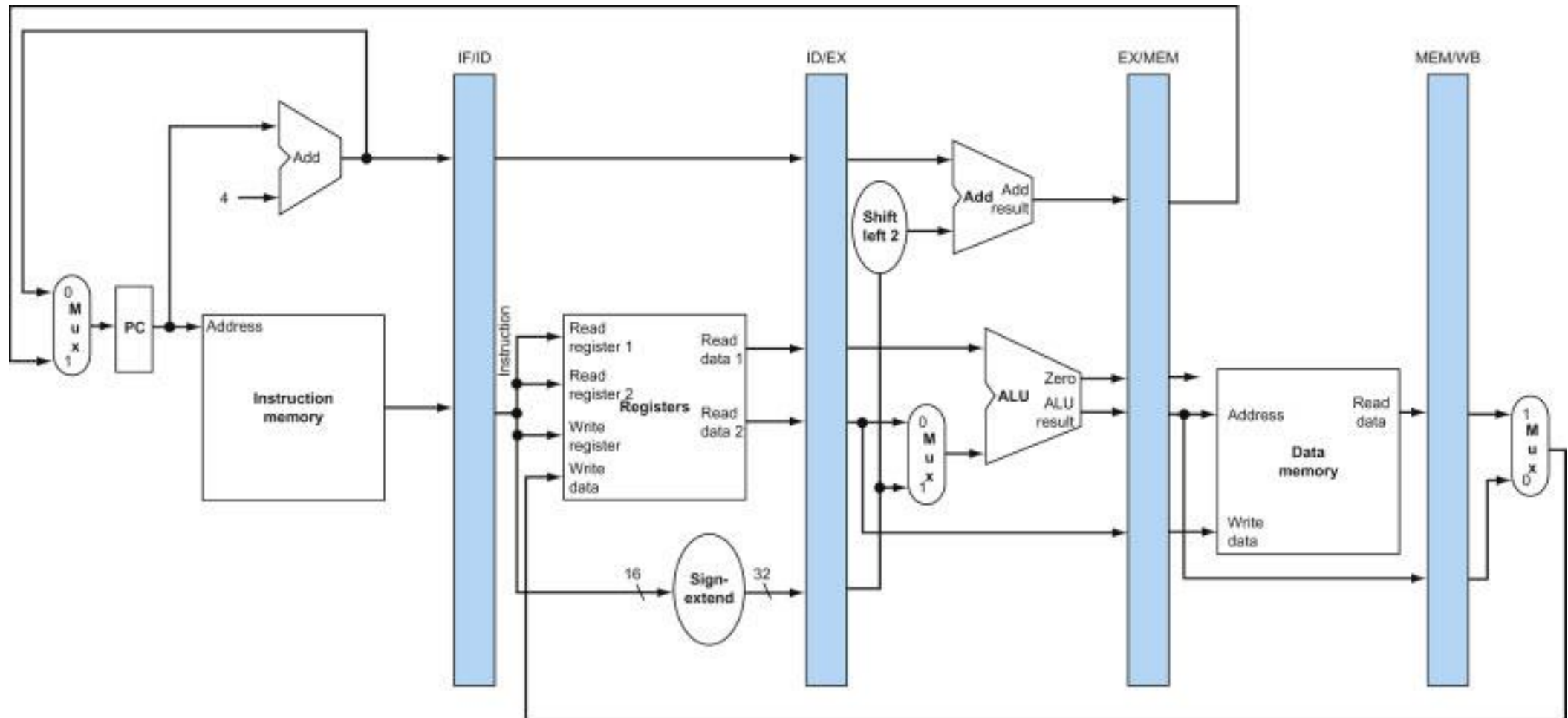
Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Como o Pipeline é implementado?



Como o Pipeline é implementado?



Stage	Any instruction		
IF	$IF/ID.IR \leftarrow Mem[PC];$ $IF/ID.NPC, PC \leftarrow (if ((EX/MEM.opcode == branch) \& EX/MEM.cond) \{EX/MEM.ALUOutput\} else \{PC+4\});$		
ID	$ID/EX.A \leftarrow Regs[IF/ID.IR[rs]]; ID/EX.B \leftarrow Regs[IF/ID.IR[rt]]; ID/EX.NPC \leftarrow IF/ID.NPC; ID/EX.IR \leftarrow IF/ID.IR;$ $ID/EX.Imm \leftarrow sign-extend(IF/ID.IR[immediate field]);$		
	ALU instruction	Load or store instruction	Branch instruction
EX	$EX/MEM.IR \leftarrow ID/EX.IR;$ $EX/MEM.ALUOutput \leftarrow ID/EX.A \text{ func } ID/EX.B;$ or $EX/MEM.ALUOutput \leftarrow ID/EX.A \text{ op } ID/EX.Imm;$	$EX/MEM.IR \text{ to } ID/EX.IR$ $EX/MEM.ALUOutput \leftarrow ID/EX.A + ID/EX.Imm;$ $EX/MEM.B \leftarrow ID/EX.B;$	$EX/MEM.ALUOutput \leftarrow ID/EX.NPC + (ID/EX.Imm \ll 2);$ $EX/MEM.cond \leftarrow (ID/EX.A == 0);$
MEM	$MEM/WB.IR \leftarrow EX/MEM.IR;$ $MEM/WB.ALUOutput \leftarrow EX/MEM.ALUOutput;$	$MEM/WB.IR \leftarrow EX/MEM.IR;$ $MEM/WB.LMD \leftarrow Mem[EX/MEM.ALUOutput];$ or $Mem[EX/MEM.ALUOutput] \leftarrow EX/MEM.B;$	
WB	$Regs[MEM/WB.IR[rd]] \leftarrow MEM/WB.ALUOutput;$ or $Regs[MEM/WB.IR[rt]] \leftarrow MEM/WB.ALUOutput;$	For load only: $Regs[MEM/WB.IR[rt]] \leftarrow MEM/WB.LMD;$	

Exercício

- Mostrar a situação dos registradores IF/ID, ID/EX, EX/MEM, MEM/WB ao final do ciclo de clock 5 considerando a seguinte sequência de código:

End. Memória	Instrução
0	add \$1, \$2, \$3
4	add \$4, \$5, \$6
8	add \$7, \$8, \$9
12	add \$10, \$11, \$12
16	add \$13, \$14, \$15

Solução

O que faz cada estágio?

- IF

- Busca da instrução armazenada na posição de memória armazenada no PC (Início do ciclo)
- Escrita da instrução no registrador IF/ID.IR do final do ciclo e atualização do valor do PC

- ID

- Decodificação e Busca dos operandos (armazenados nas posições [25..21], [20..16] e [15..0] do IF/ID.IR armazenado no início do ciclo)
- Copia o IF/ID.IR para o ID/EX.IR e o IF/ID. NPC para o ID/EX.NPC

Solução

O que faz cada estágio?

- EX

- Execução da operação (depende da decodificação)
 - ULA : $ID/EX.A + ID/EX.B$
 - lw ou sw: $ID/EX.A + ID/EX.Imm$
 - beq: $ID/EX.NPC + (ID/EX.Imm \ll 2)$
- Copia o IF/ID.IR para o ID/EX.IR e ID/EX.B para o EX/MEM.B

- MEM

- Acessa a Memória para
 - leitura $MEM/WB.LMD = MEM[EX/MEM.ALUoutput]$ ou
 - escrita $MEM[EX/MEM.ALUoutput] = EX/MEM.B$

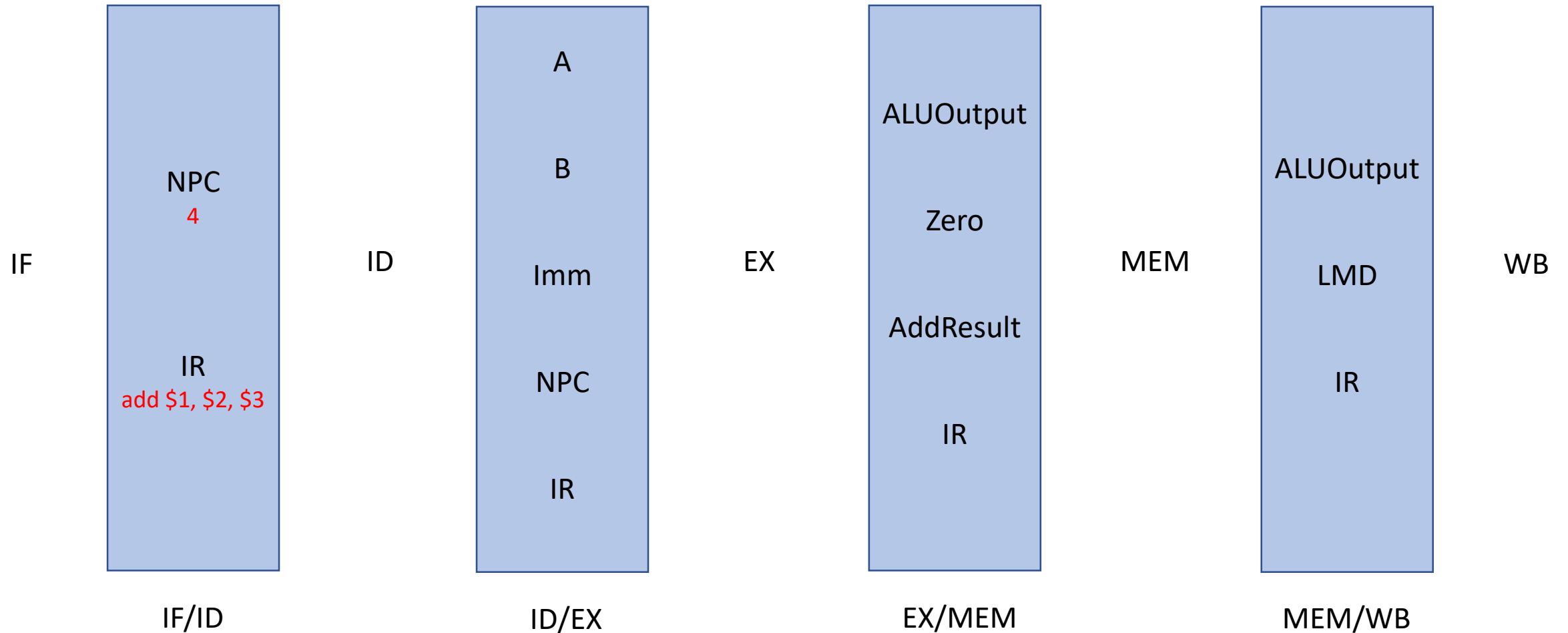
Solução

O que faz cada estágio?

- WB
 - Finaliza a execução da instrução escrevendo no banco de registradores
 - $\text{Bco_Reg}[\text{MEM/WB.IR}[\text{rd}]] = \text{MEM/WB. ALUOutput}$ ou
 - $\text{Bco_Reg}[\text{MEM/WB.IR}[\text{rt}]] = \text{MEM/WB. LMD}$

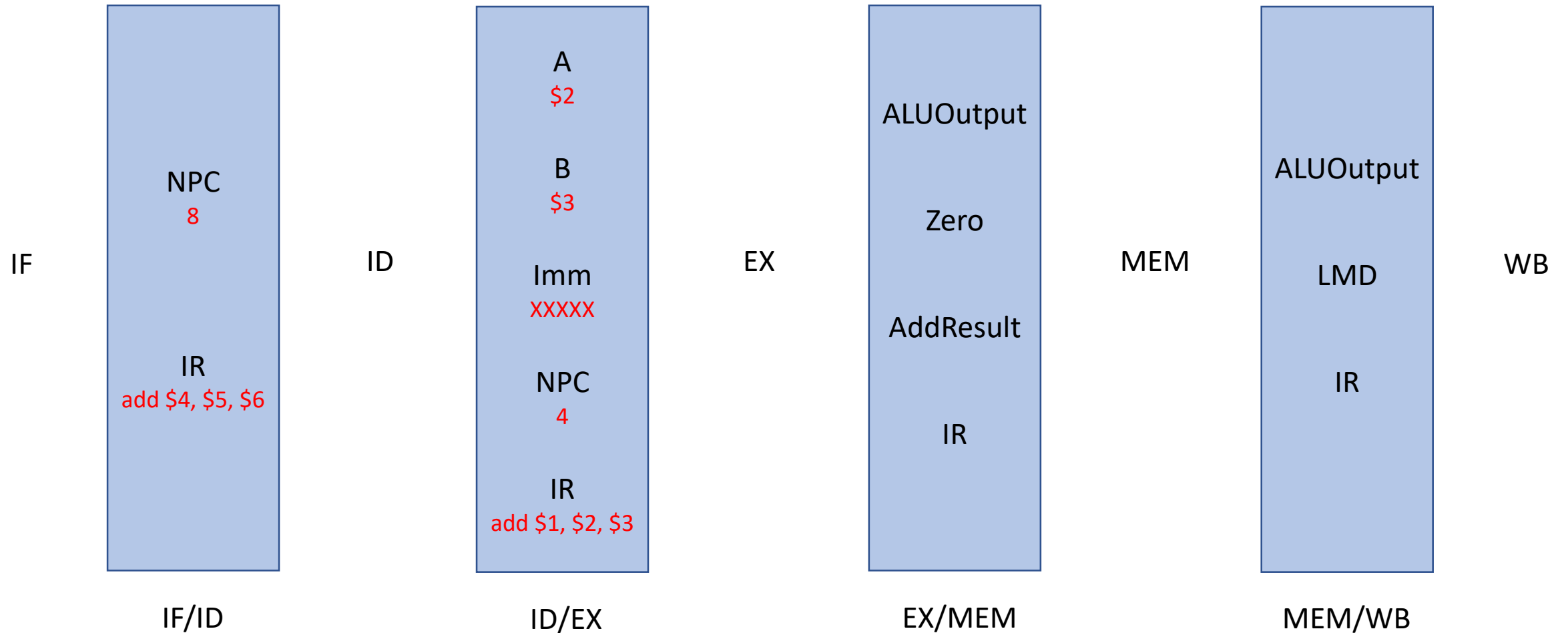
Solução

Ciclo de clock 1



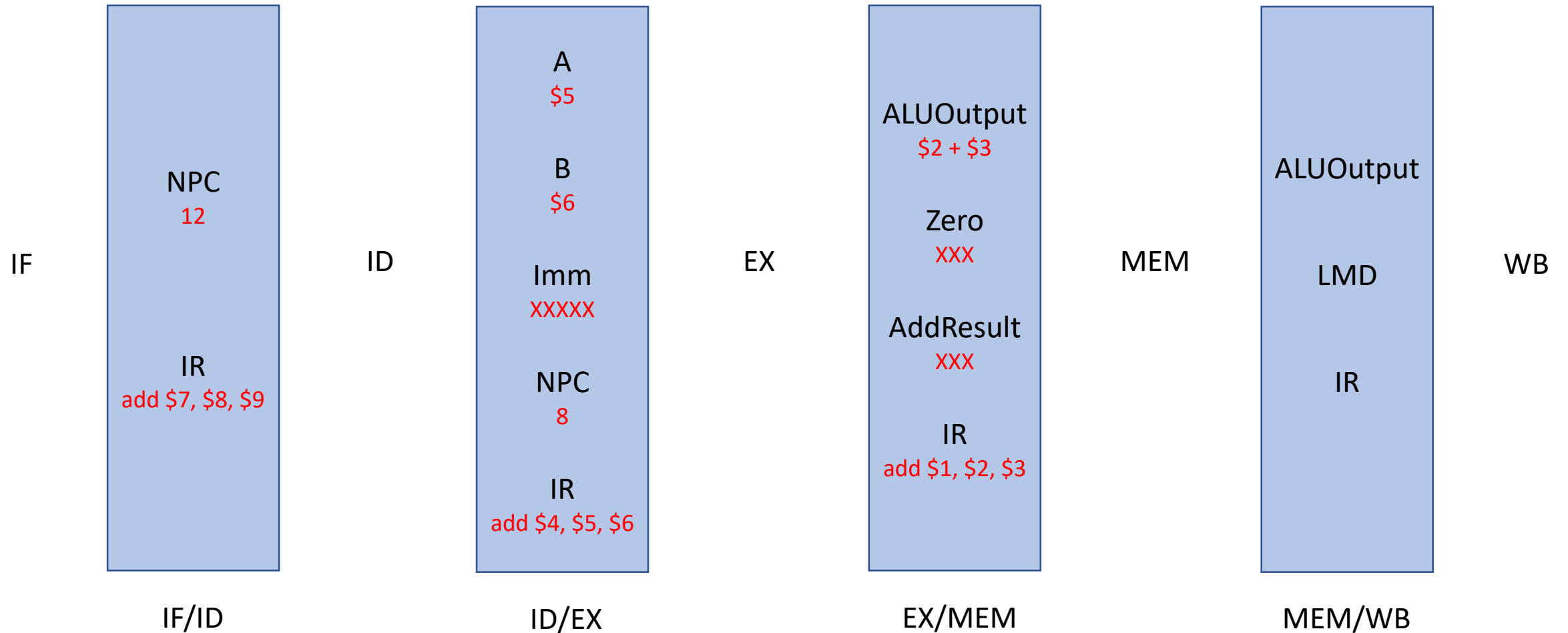
Solução

Ciclo de clock 2



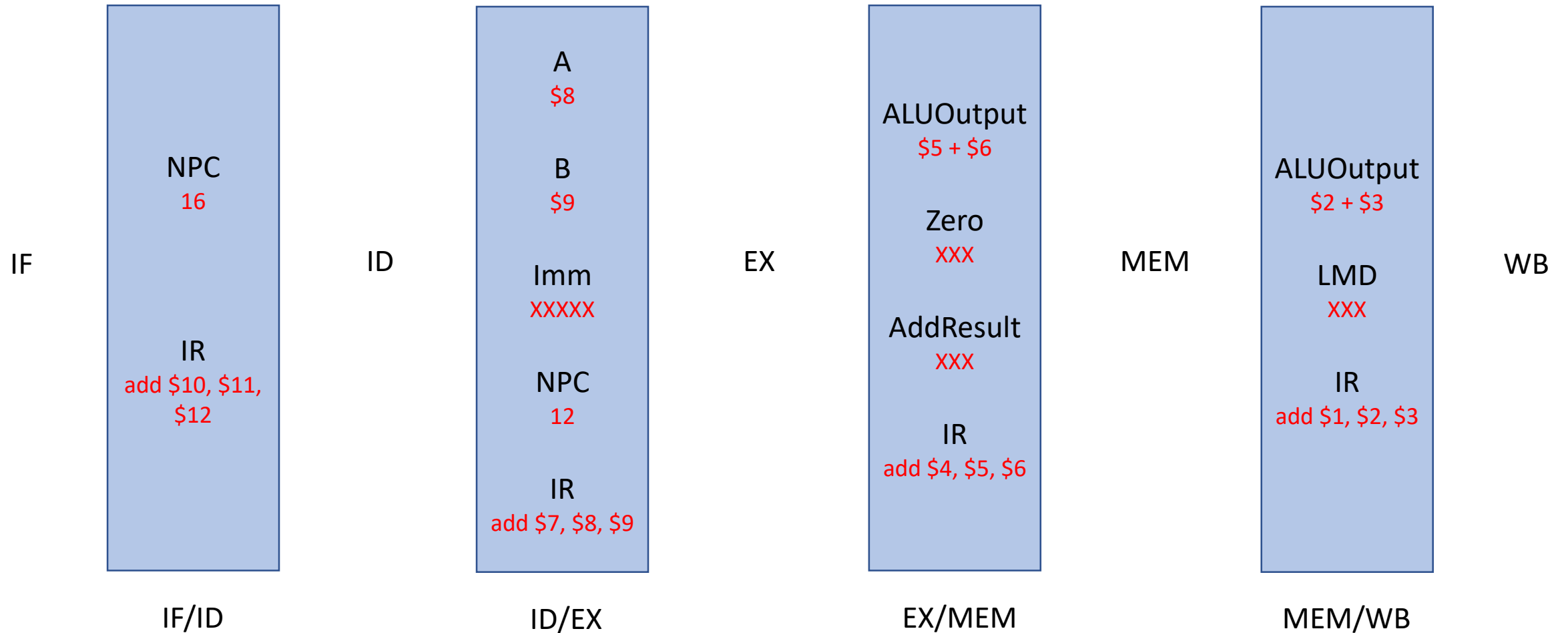
Solução

Ciclo de clock 3



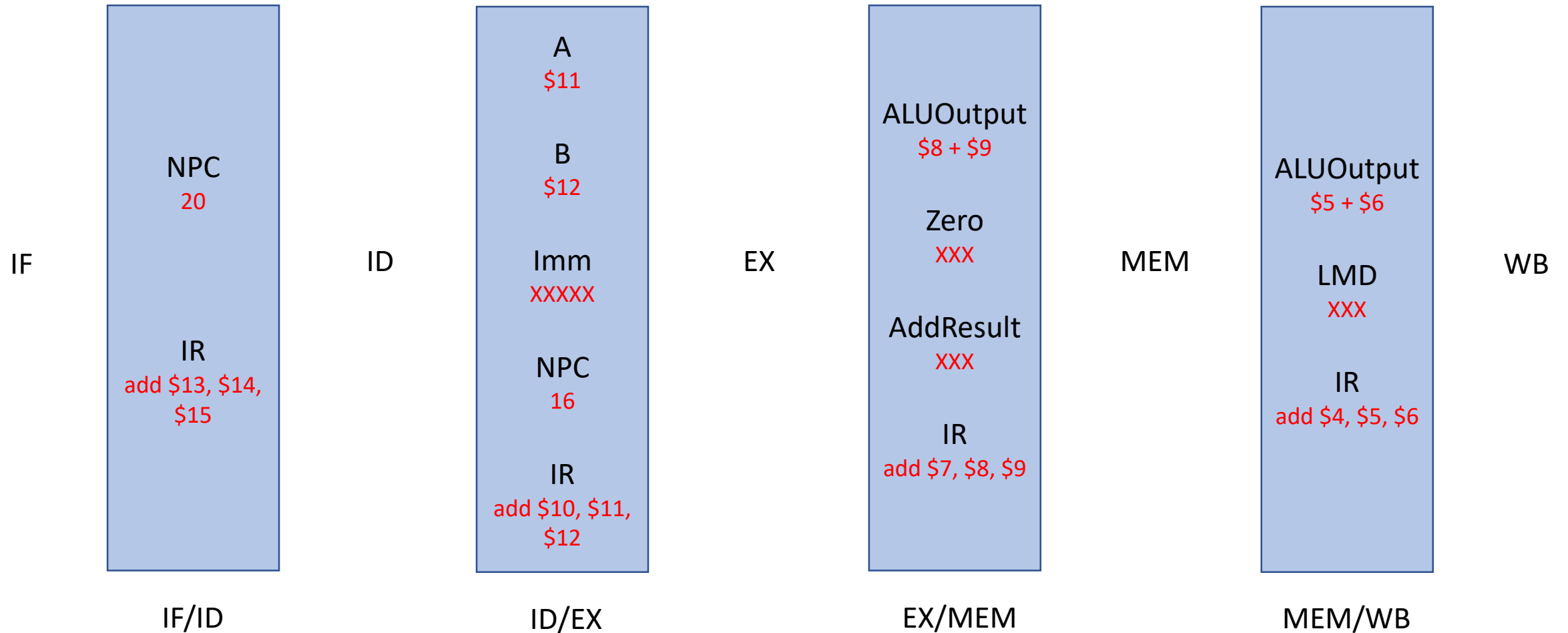
Solução

Ciclo de clock 4



Solução

Ciclo de clock 5



Solução

- Ao final do ciclo de clock 5 os conteúdos dos registradores serão:
 - IF/ID.NPC = 20
 - IF/ID.IR = add \$13, \$14, \$15
 - ID/EX.A = \$11
 - ID/EX.B = \$12
 - ID/EX.Imm = XXX
 - ID/EX.NPC = 16
 - ID/EX.IR = add \$10, \$11, \$12

Solução

- Ao final do ciclo de clock 5 os conteúdos dos registradores serão:
 - EX/MEM.ALUOutput = $\$8 + \9
 - EX/MEM.Zero = XXX
 - EX/MEM.AddResult = XXX
 - EX/MEM.IR = add $\$7, \$8, \$9$
- MEM/WB.ALUOutput = $\$5 + \6
- MEM/WB.LMD = XXX
- MEM/WB.IR = add $\$4, \$5, \$6$
- Bco_Reg[\$1] = $\$2 + \3 (MEM/WB.ALUOutput)