Universidade de Brasília
Departamento de Ciência da Computação

Aula 17
Implementação RISC-V
Pipeline – Unidade Operativa e Controle

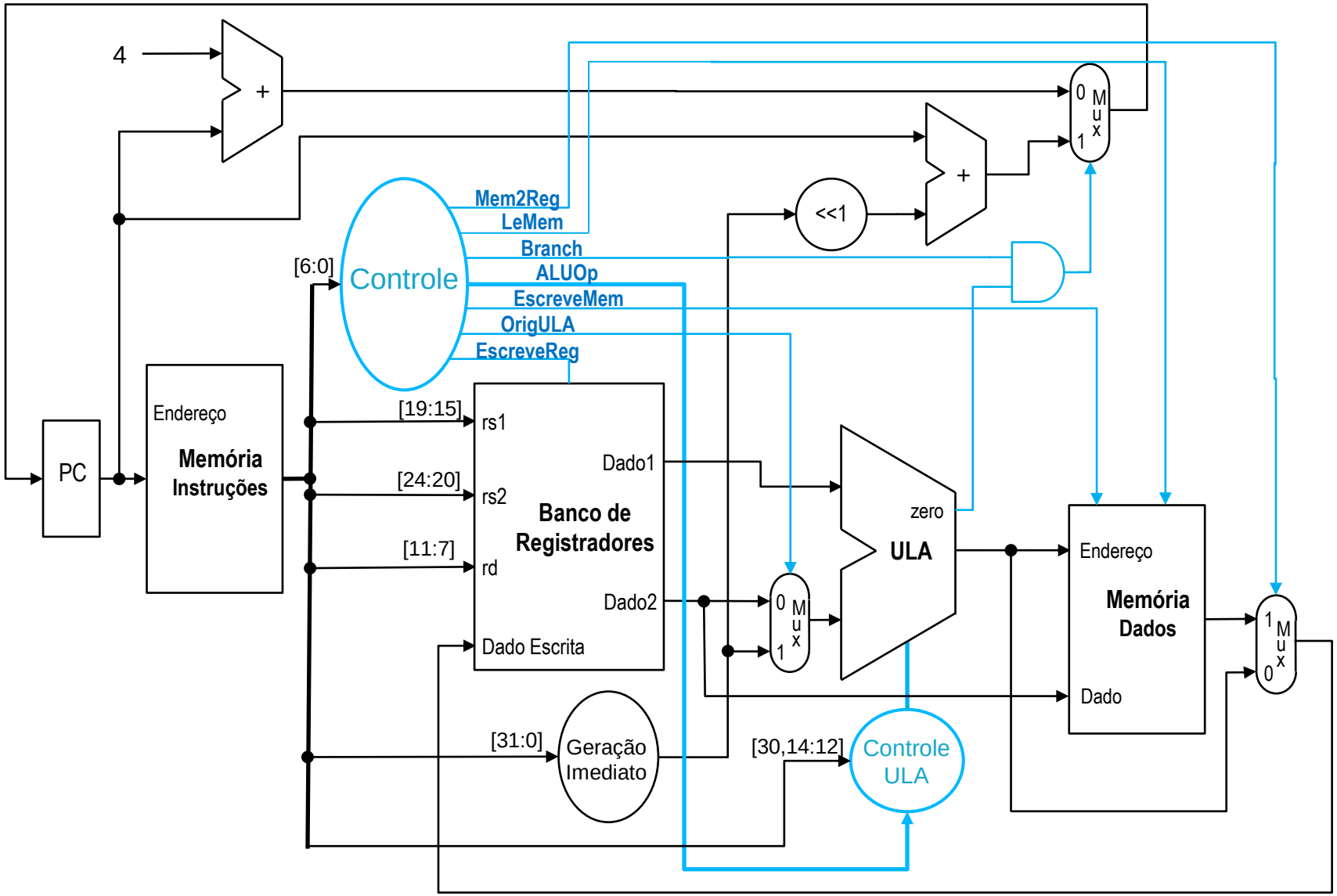
Controle do pipeline

- Temos 5 estágios.

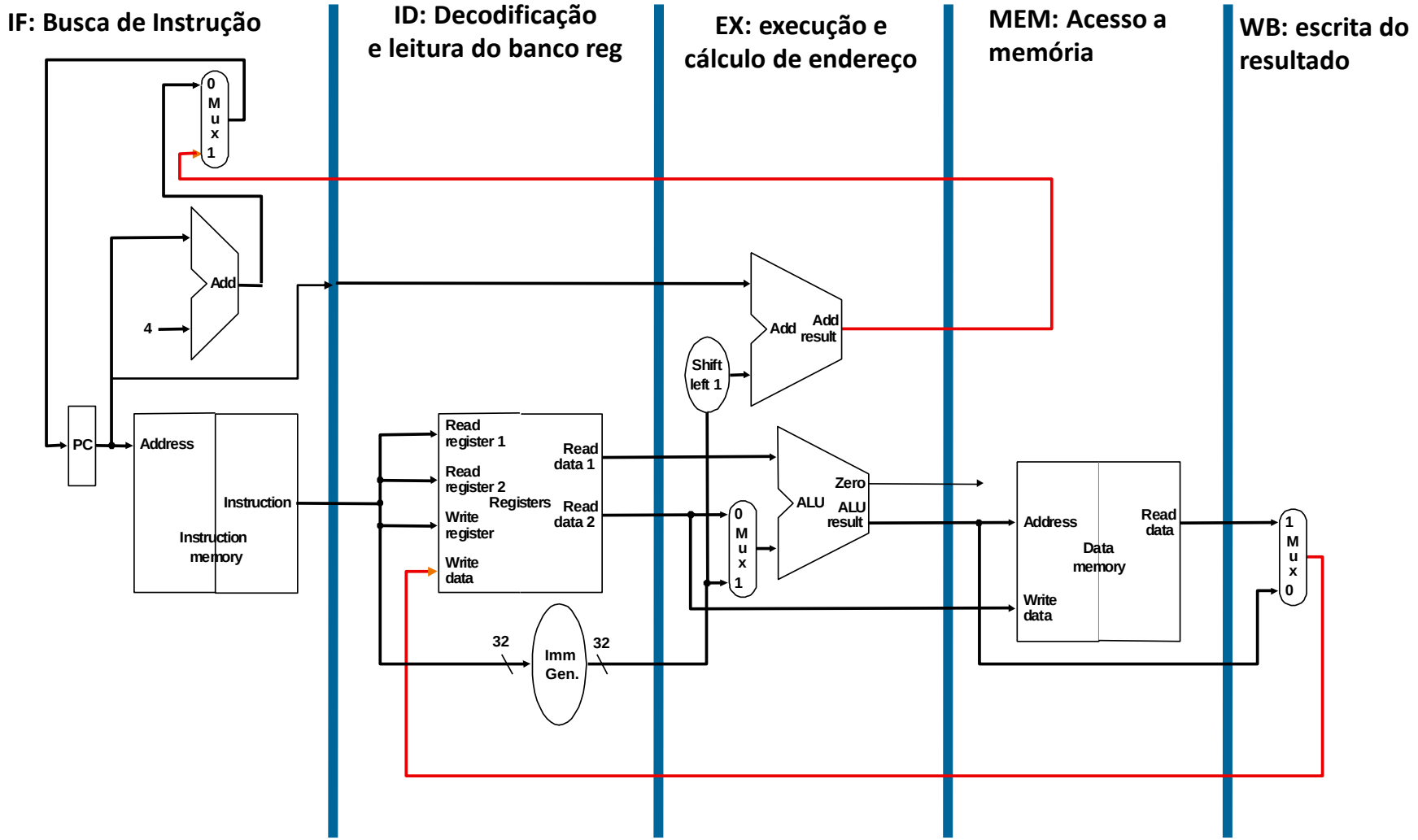
  O que deve ser controlado em cada estágio?

IF:	*Instruction Fetch* e PC *Increment*
ID:	*Instruction Decode / Register Fetch*
EX:	*Execution*
MEM:	*Memory Stage*
WB:	*Write Back*

Caminho de Dados UNICICLO

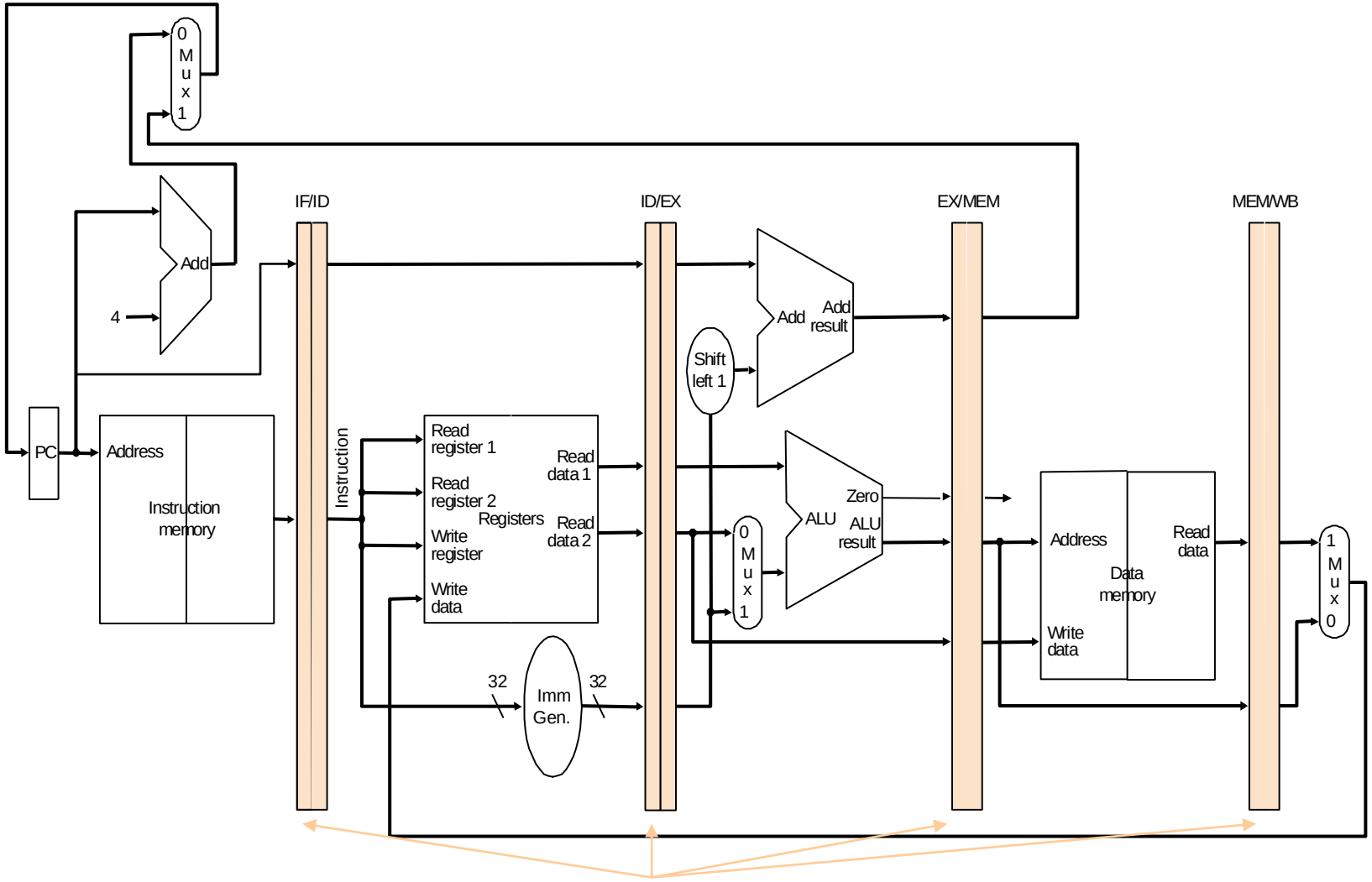


Caminho de dados UNICICLO



Fluxo de dados → com 2 exceções

Caminho de dados com *pipeline*

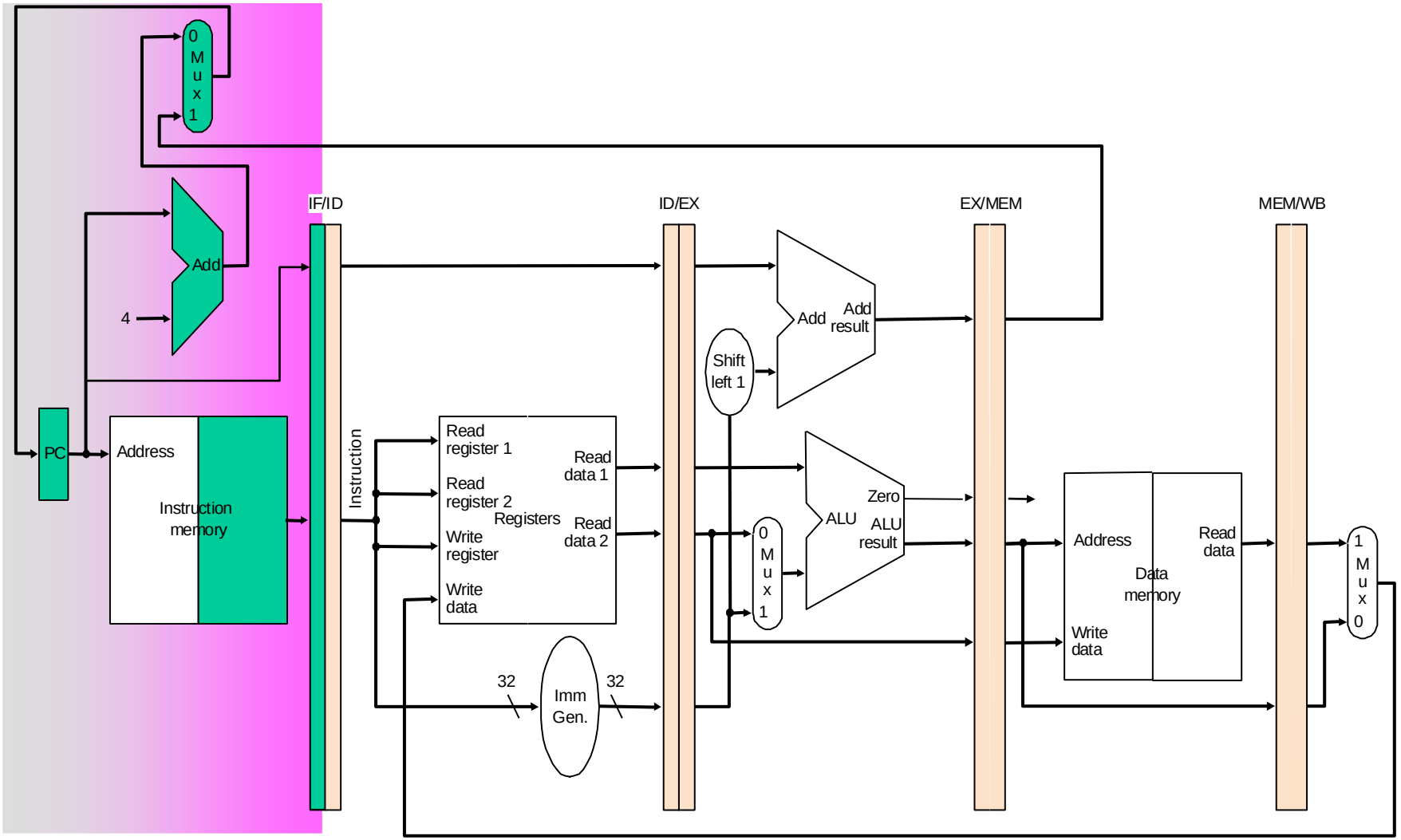


Registradores dos estágios do pipeline

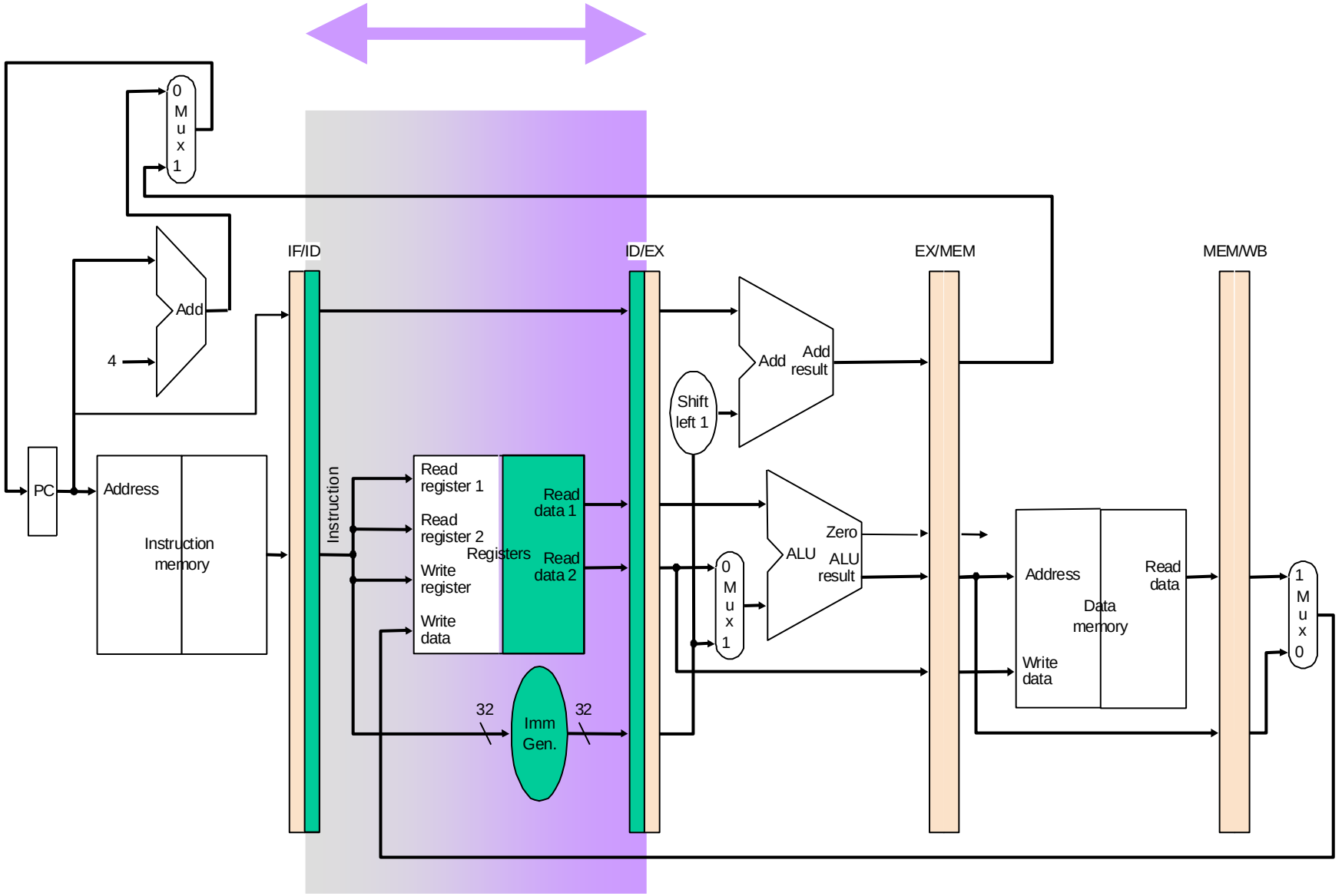
Quais seus tamanhos?

Busca de instrução

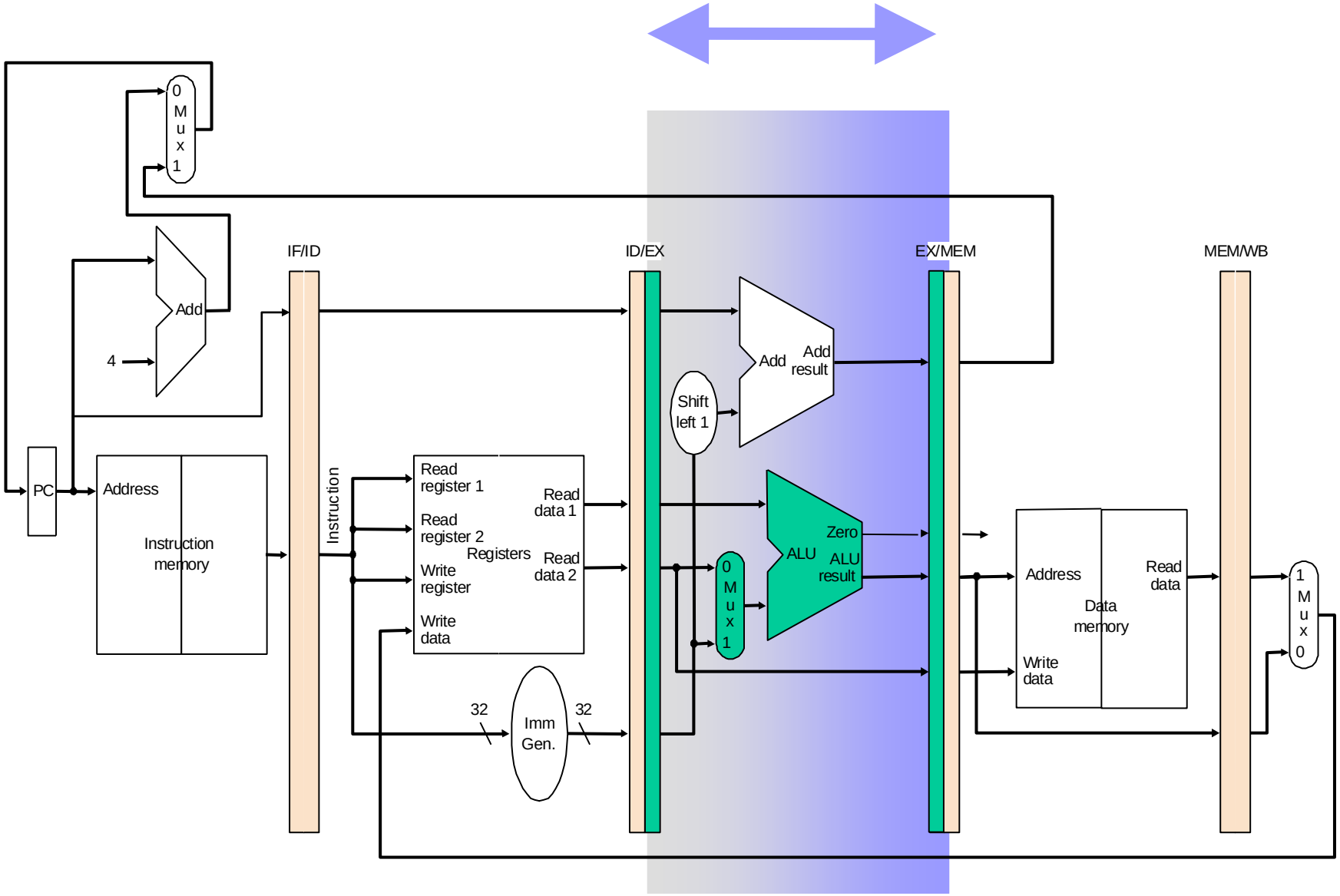
Ex.: Load



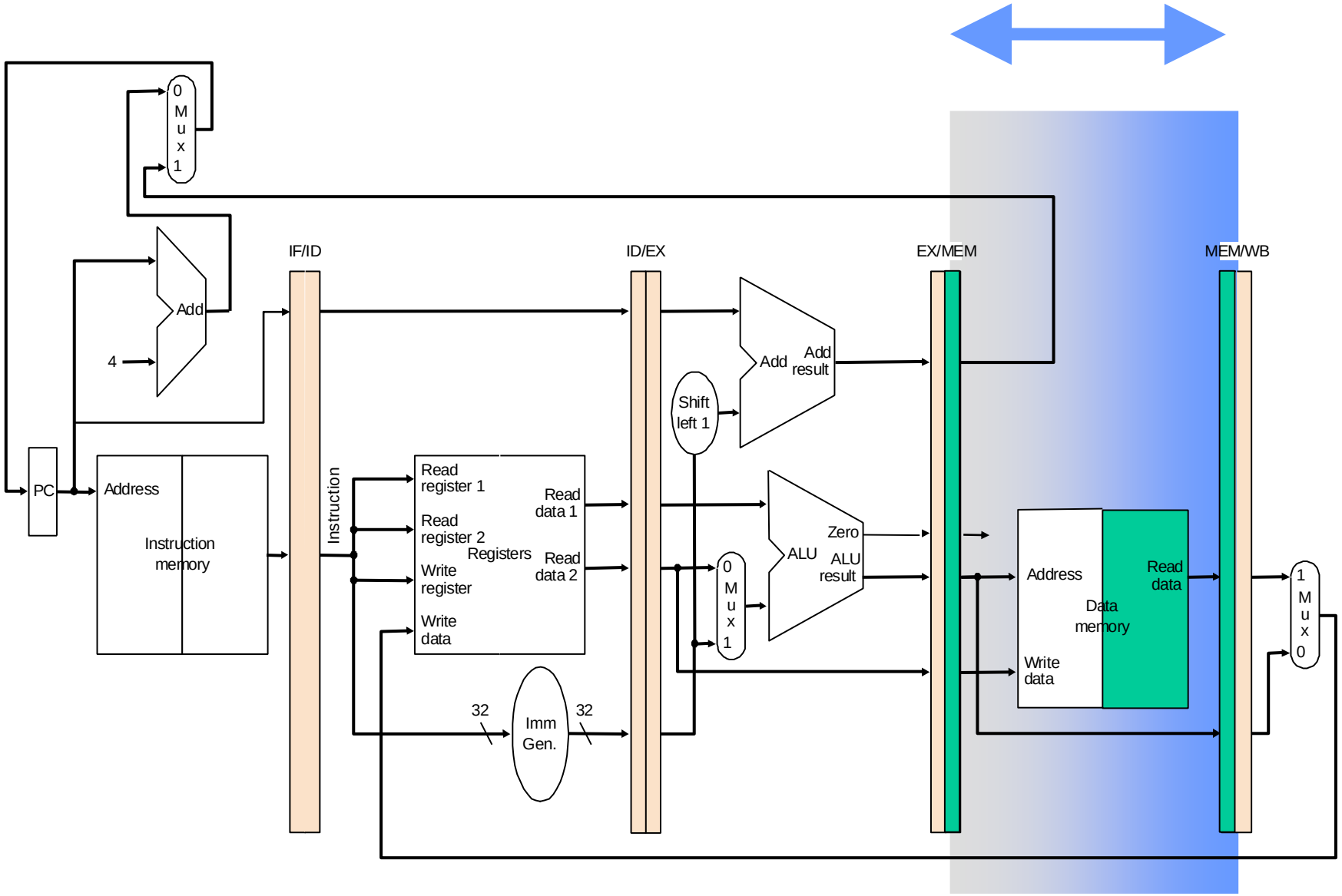
Decodificação/leitura de registros



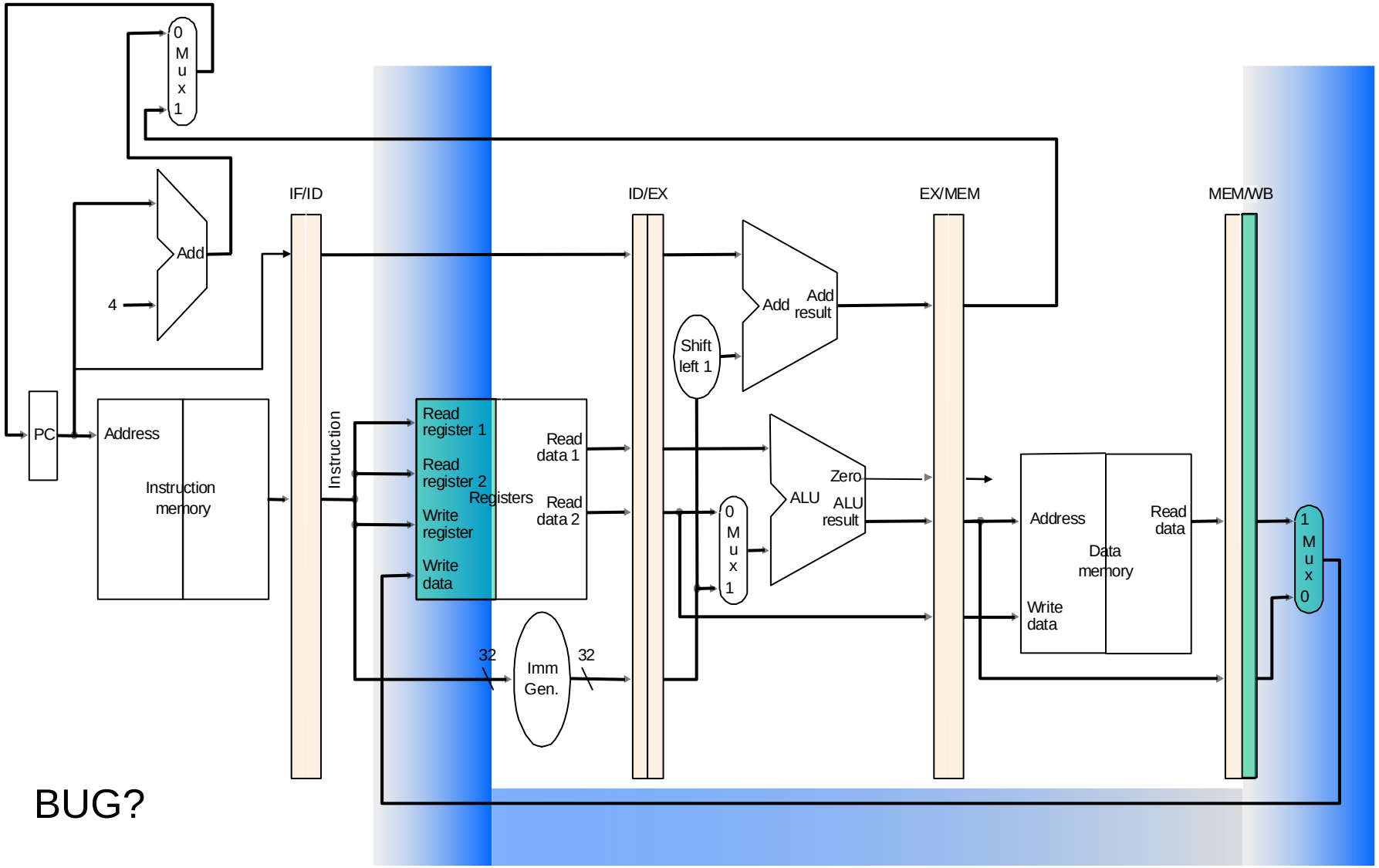
Execução/cálculo de endereço



Acesso à memória

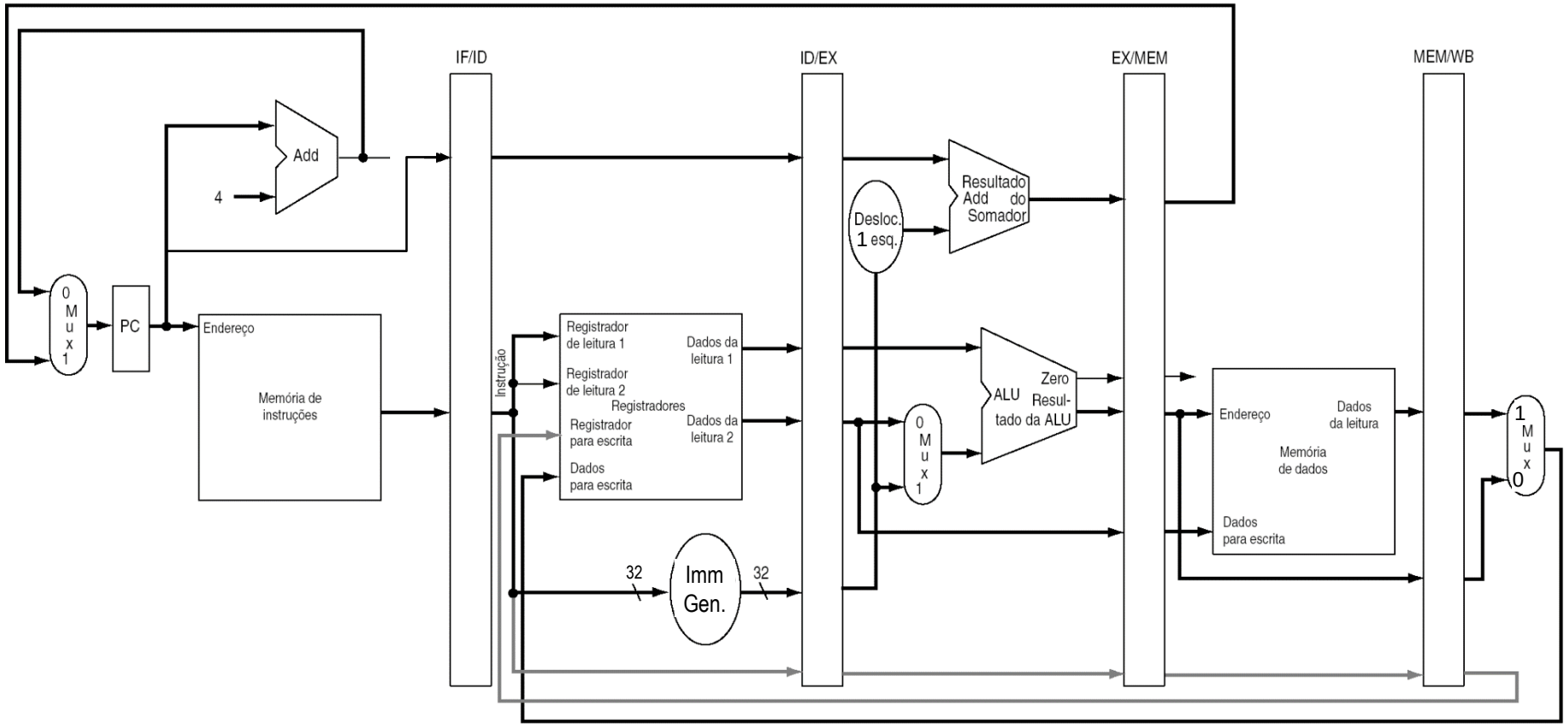


Escrita do Resultado

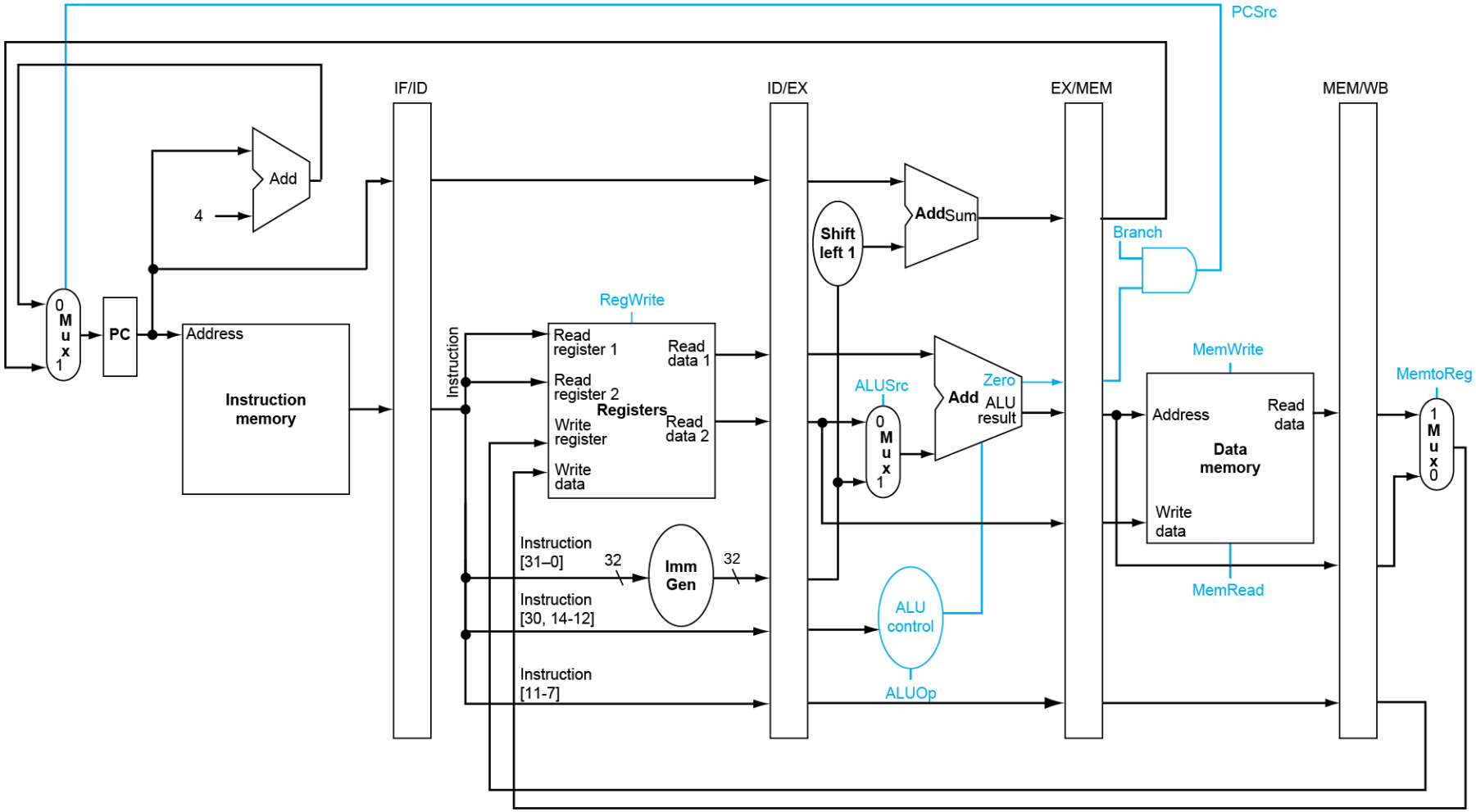


BUG?

Corrigindo o BUG!!!



Controle do pipeline – Sinais de Controle



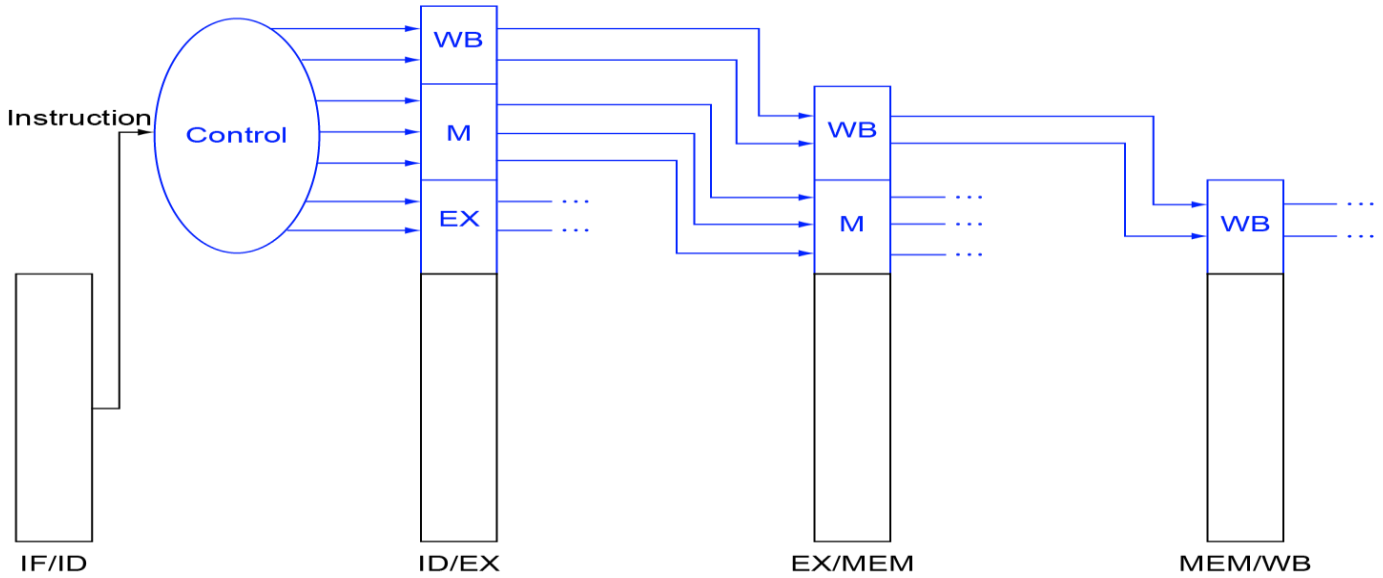
Controle do pipeline

- Temos 5 estágios. O que precisa ser controlado em cada estágio?
  - Busca da instrução:
    - Nenhum sinal de controle
  - Decodificação da instrução / leitura do banco de regs.:
    - Nenhum sinal de controle
  - Execução / cálculo de endereço:
    - ALUOp, ALUSrc
  - Acesso a memória:
    - Branch, MemRead, MemWrite
  - Escrita do resultado
    - Mem2Reg, RegWrite

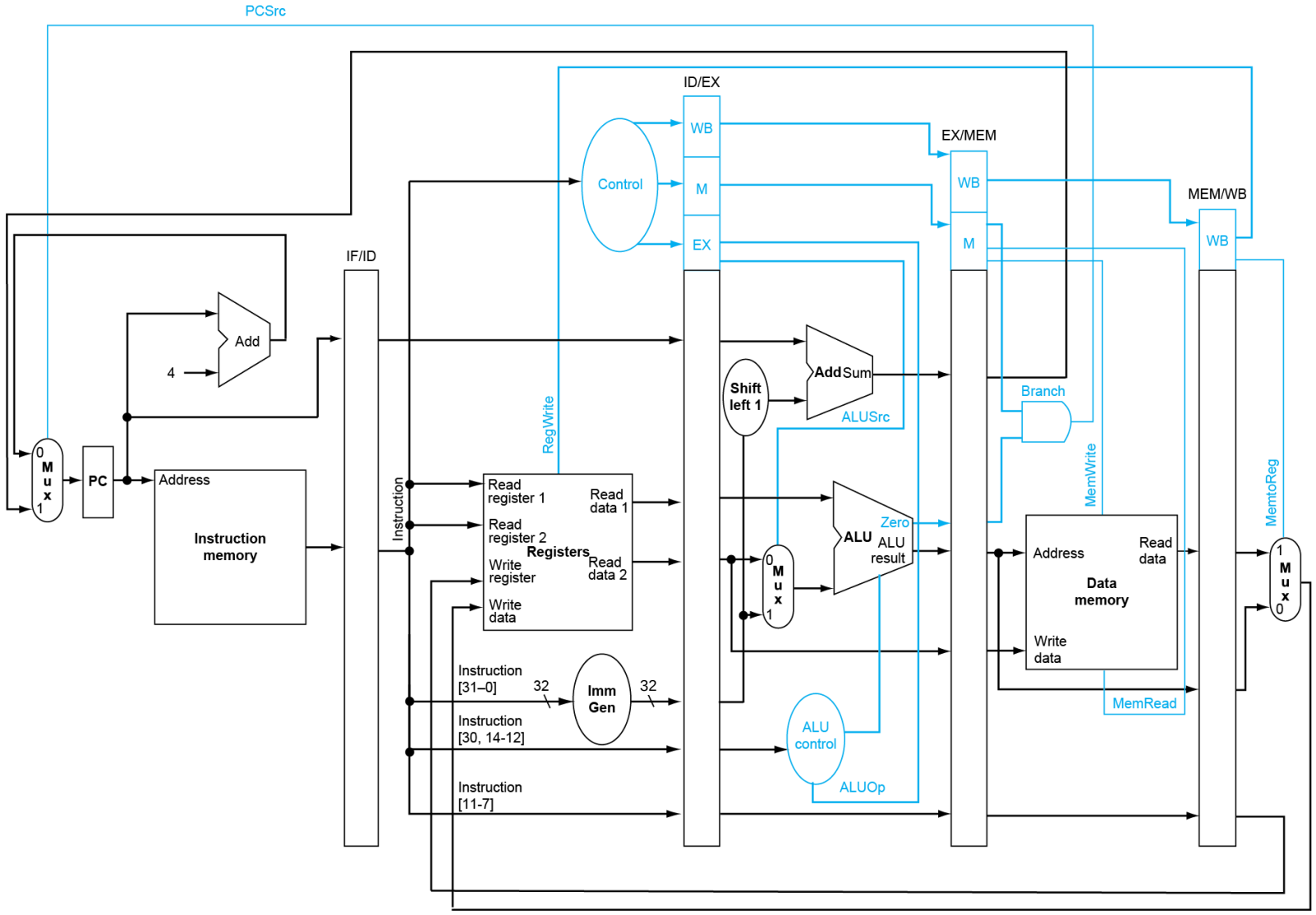
Controle do *Pipeline*

Sinais de controle são transmitidos da mesma forma que os dados

	Estágio de EX		Estágio MEM			Estágio WB	
Instrução	ALUSrc	ALUOp	Branch	MemRead	MemWrite	RegWrite	Mem2Reg
Tipo-R	0	10	0	0	0	1	0
lw	1	00	0	1	0	1	1
sw	1	00	0	0	1	0	X
beq	0	01	1	0	0	0	X

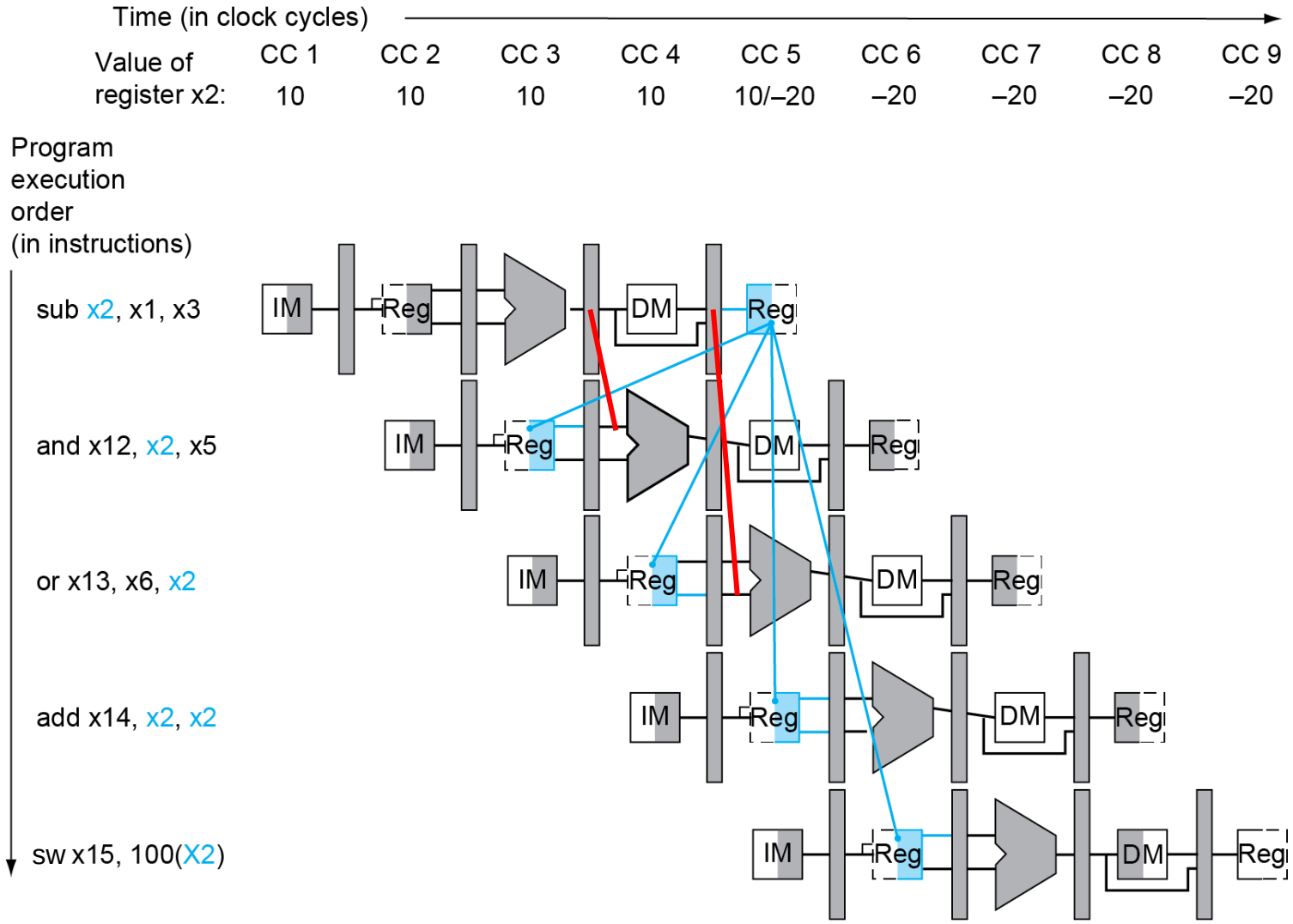


Caminho de dados pipeline com controle

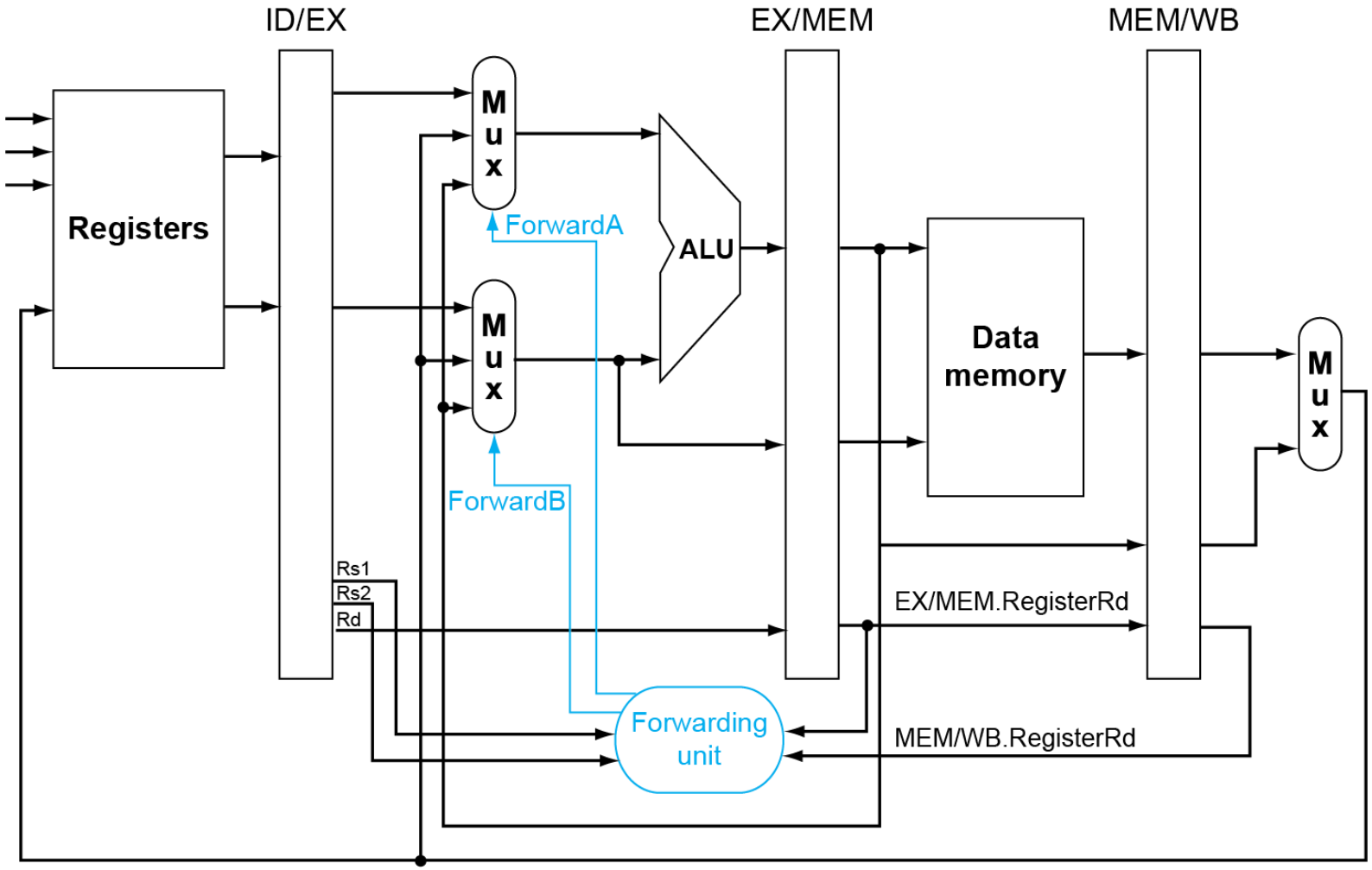


HAZARDS

- Hazard de Dados: Forwarding

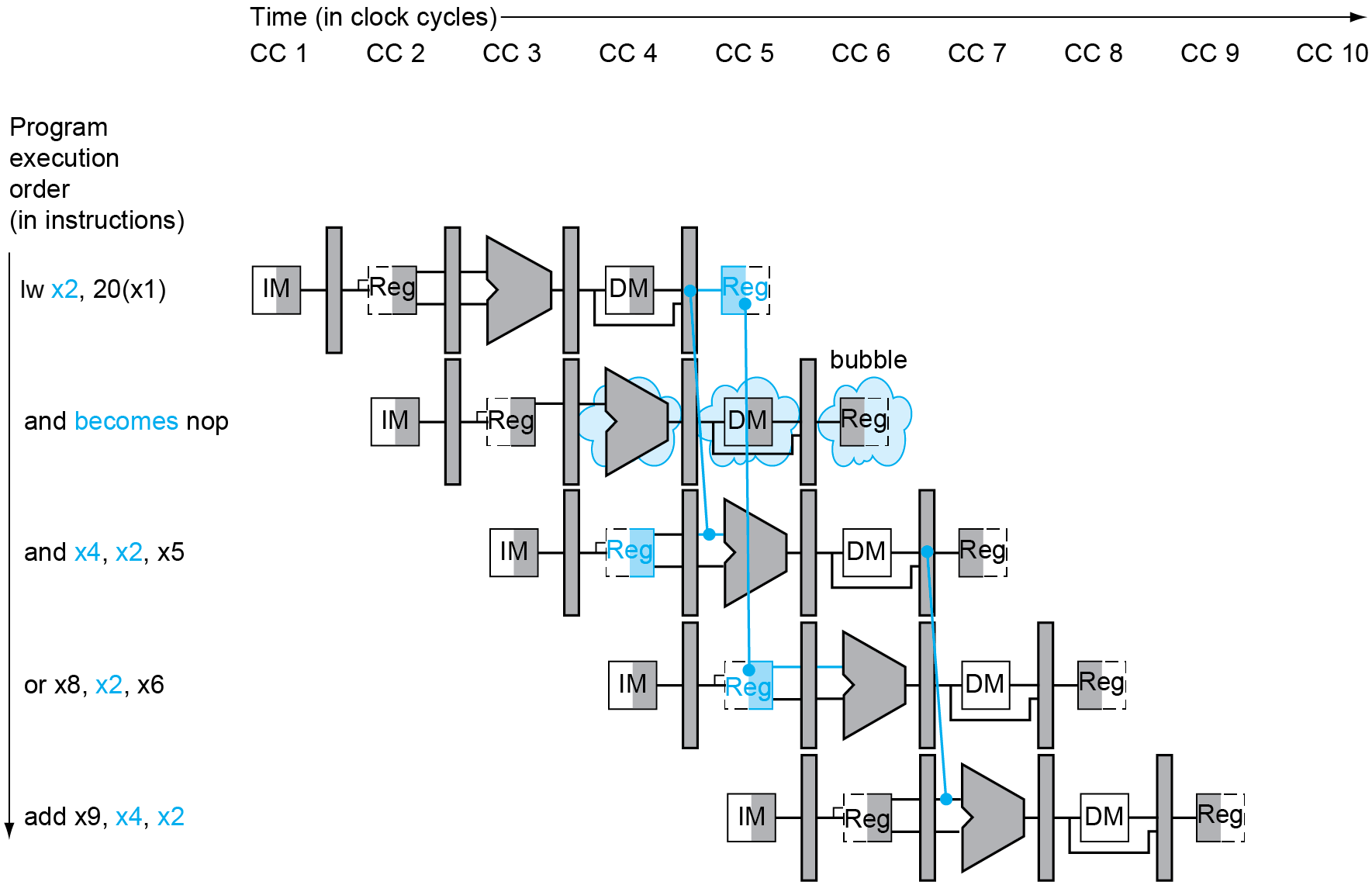


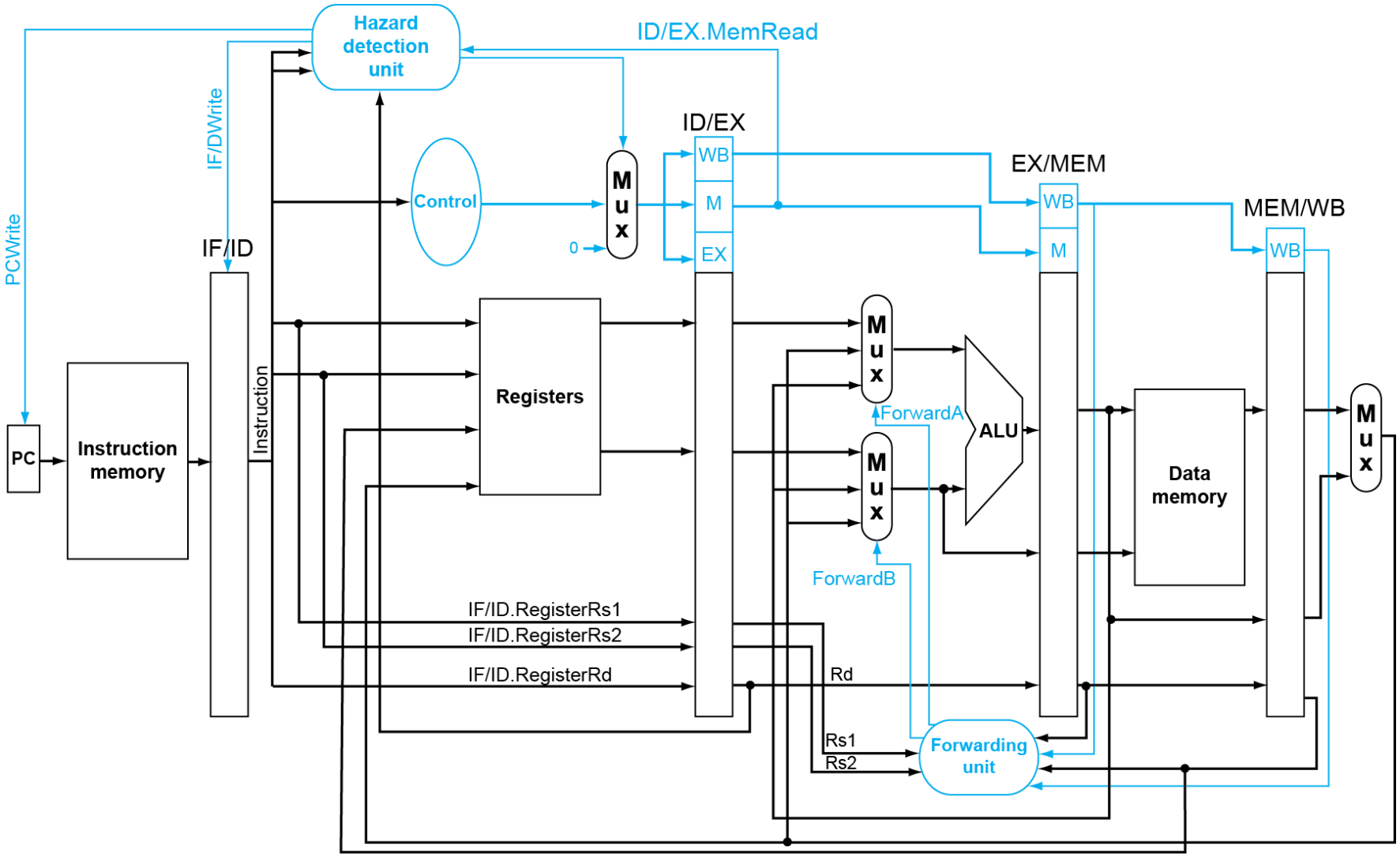
Hazard de dados: Forwarding



Obs.: Falta o mux OrigALU após o ForwardB!

Quando forward não é possível




Hazard detection unit
ID/EX.MemRead
IF/DWrite
PCWrite
ID/EX
EX/MEM
MEM/WB
Control
M u x
0
WB
M
EX
IF/ID
PC
Instruction memory
Instruction
Registers
ForwardA
ALU
ForwardB
Data memory
IF/ID.RegisterRs1
IF/ID.RegisterRs2
IF/ID.RegisterRd
Rd
Rs1
Rs2
Forwarding unit

Deteção de Hazards

- Comparar índices dos registradores a serem lidos e escritos, armazenados nos registradores do *pipeline:*

- Deve-se comparar igualmente se a instrução escreve em um registrador (EscreveReg = 1) ou
  lê da memória (LeMem = 1)

Obs.:

Flush descarta uma instrução do pipeline.

Stall paralisa o pipeline.

Detecção de Hazard de dados

- 1. Hazard EX

  if (EX/MEM.EscreveReg
  and (EX/MEM.RegistradorRd ≠ 0)
  and (EX/MEM.RegistradorRd = ID/EX.RegistardorRs1)) ForwardA = 10

  if (EX/MEM.EscreveReg
  and (EX/MEM.RegistradorRd ≠ 0)
  and (EX/MEM.RegistradorRd = ID/EX.RegistardorRs2)) ForwardB = 10

Não se deve dar forward no x0!!!

- 2. Hazard MEM

  if (MEM/WB.EscreveReg
  and (MEM/WB.RegistradorRd $\neq$ 0)
  and (MEM/WB.RegistradorRd = ID/EX.RegistradorRs1)) ForwardA = 01

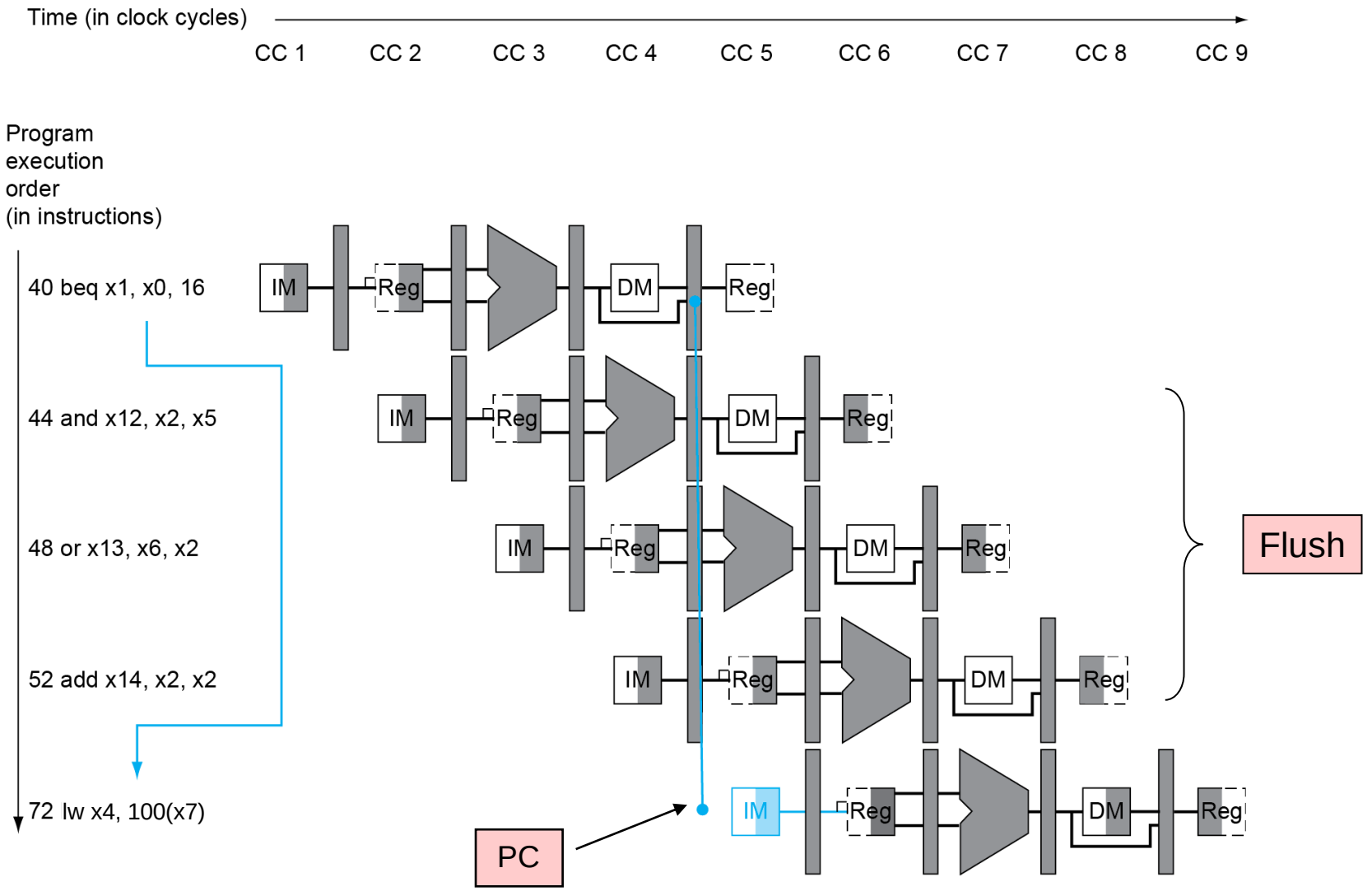
  if (MEM/WB.EscreveReg
  and (MEM/WB.RegistradorRd $\neq$ 0)
  and (MEM/WB.RegistradorRd = ID/EX.RegistradorRs2)) ForwardB = 01

- 3. LW

  if (ID/EX.LeMem and ((ID/EX.RegistradorRd = IF/ID.RegistradorRs1)
  or (ID/EX.RegistradorRd = IF/ID.RegistradorRs2))) Ocasiona stall no pipeline

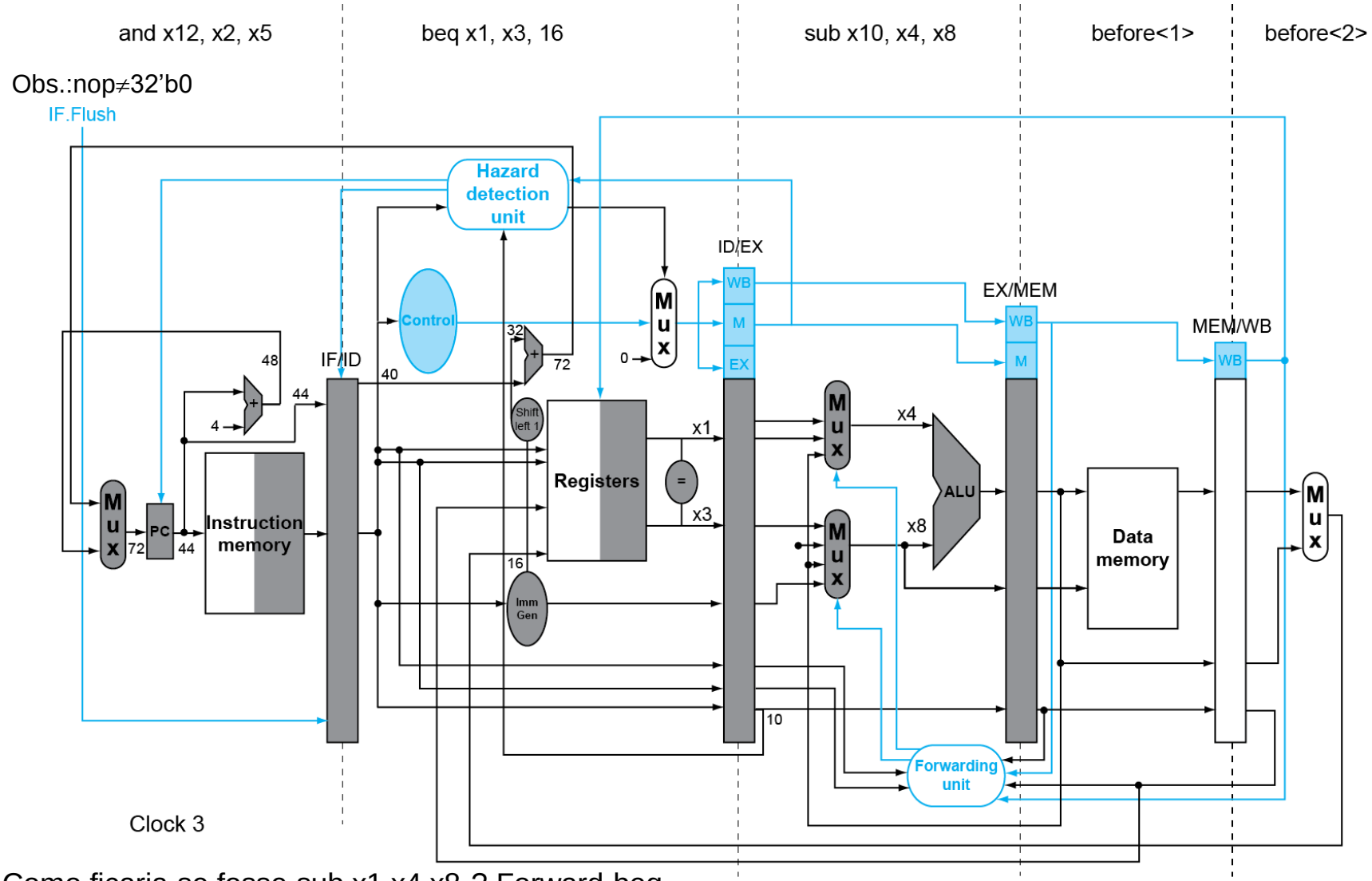
Hazard de Controle

- Prevendo: Desvio não-tomado



Hazard de Controle

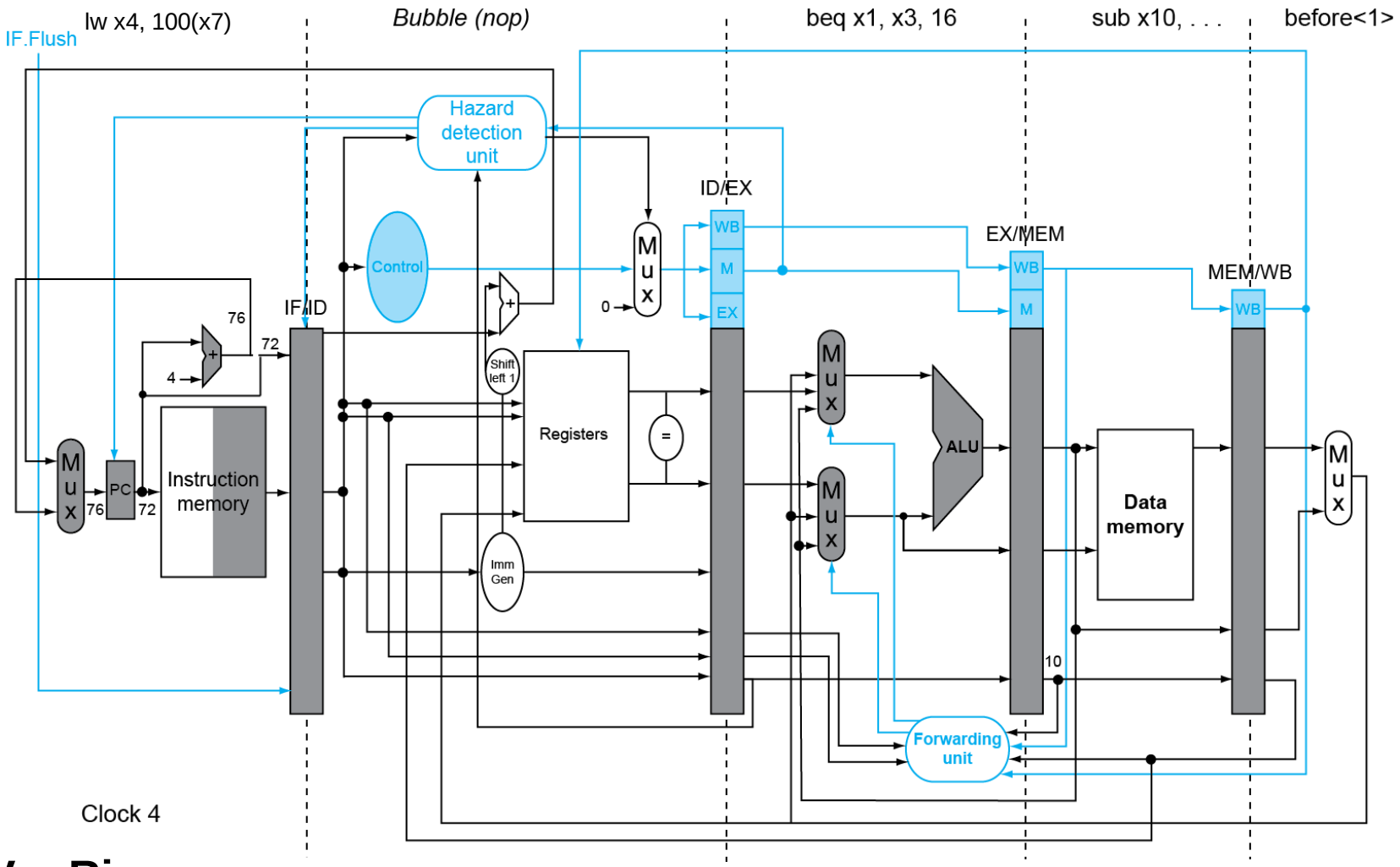
- Hardware otimizado para 1 bolha



Como ficaria se fosse sub x1,x4,x8 ? Forward beq

Hazard de Controle

- Hardware otimizado para 1 bolha



Ver Ripes

Comparação: Uniciclo x Multiciclo x Pipeline

- Suponha os seguintes tempos das unidades operativas:
  - Acesso a memória: 200ps
  - Operação da ULA: 100ps
  - Acesso ao banco de registradores: 50ps
- Dado o workload composto de:
  - 25% loads
  - 10% stores
  - 11%branches
  - 2% jumps
  - 52% operações com a ULA

Poderia considerar o
Setup time dos registradores
do Multiciclo e do Pipeline

Compare o desempenho, considerando para o pipeline

50% dos loads é seguido de uma operação que requer o argumento

25% dos desvios são previstos erradamente e que o atraso da previsão errada é 1 ciclo de clock

jumps usam 2 ciclos de clock

ignore outros hazards.

Solução

Uniciclo:

período de clock: 200+50+100+200+50 = 600ps

CPI=1

Tempo de execução médio da instrução: 600x1=600ps

Multiciclo:

período de clock: 200ps

CPI = 0.25×5+0.1×4+0.11×3+0.02×3+0.52×4 = 4.12

Tempo de execução médio da instrução: 200×4.12=824ps

Pipeline:

período de clock: 200ps

CPI = 0.25×1.5+0.1×1+0.11×1.25+0.02×2+0.52×1 = 1.17

Tempo de execução médio da instrução: 200×1.17=234ps