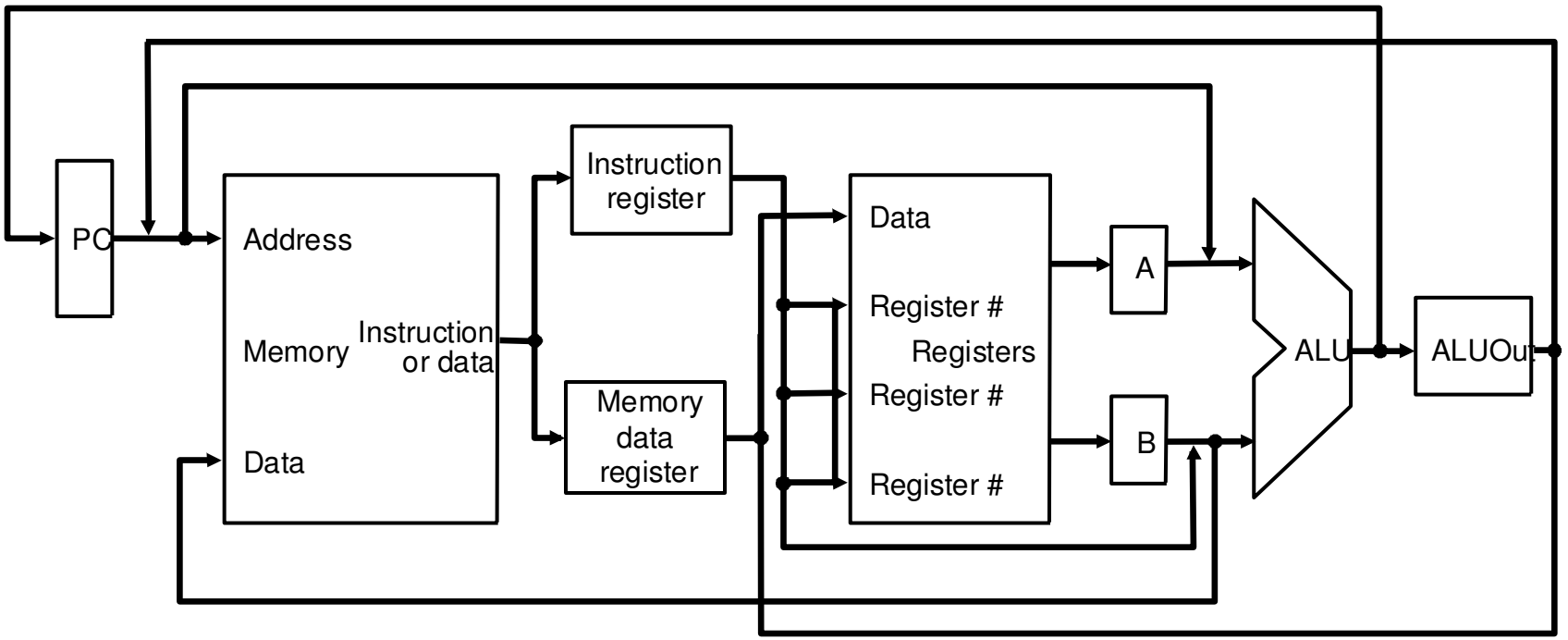
O Porcessador: Via de Dados e Controle (Parte B: multiciclo)

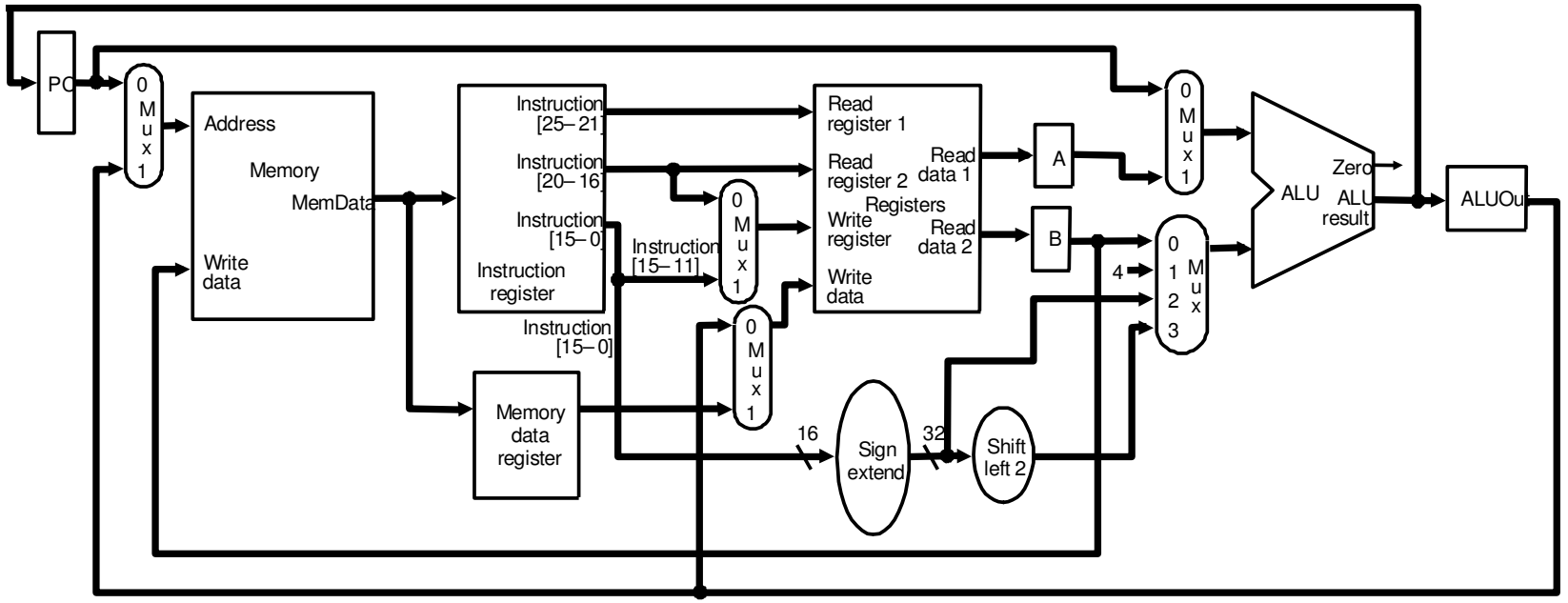
Abordagem Multiciclo

- **Dividir a instrução em passos, cada passo corresponde a um ciclo**
  - **Balancear a quantidade de trabalho a ser feito**
  - **Restringir cada ciclo para usar apenas uma "grande" unidade funcional**
    - 1 ULA, 1 Memória, 1 Banco de Registradores
- **No fim de um ciclo**
  - **Armazenar valores para uso em ciclos posteriores**
  - **Introduzir registradores internos adicionais!**

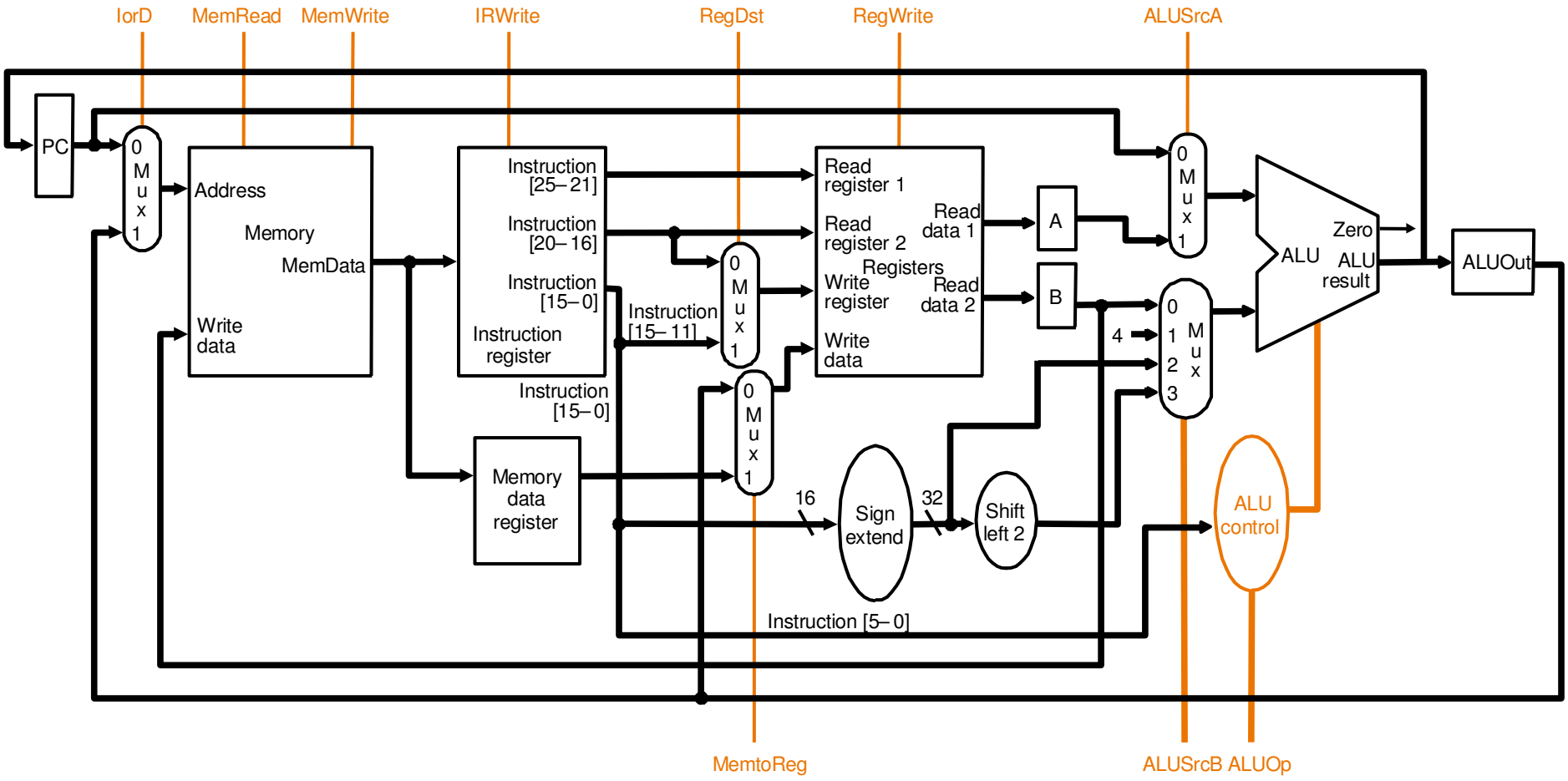


Abordagem Monociclo

- **Registrador de Instrução (IR) e Registrador da Memória de Dados (MDR) salvam saída da memória**
- **Registradores A e B salvam saída do banco de registradores**
- **ALUout salva saída da ALU**
- **Todos, exceto IR, guardam dados por um ciclo de clock ⇒ controle de escrita é desnecessário**
- **novos MUXes: endereço da memória, operandos da ALU**

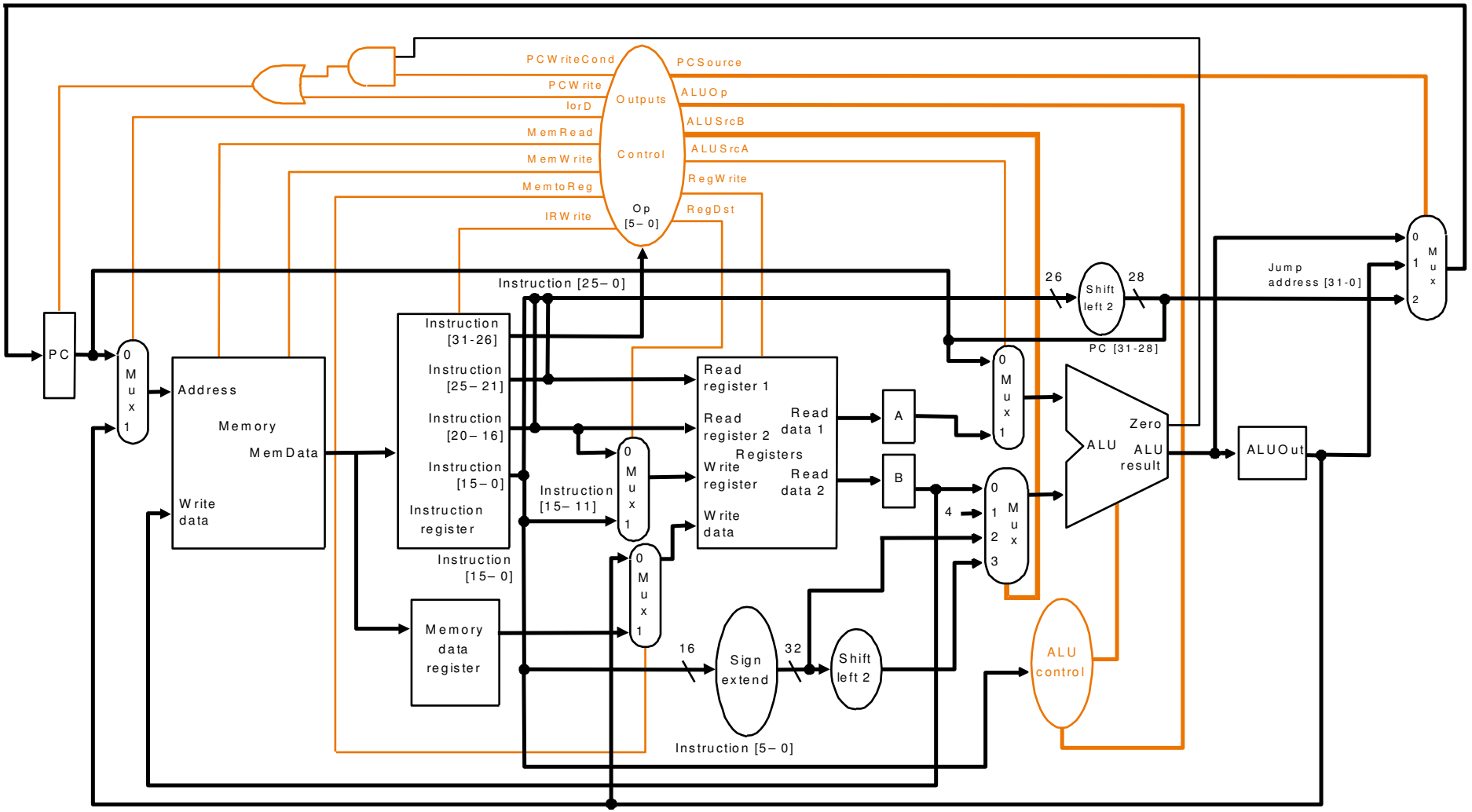


Via de dados multiciclo e sinais de controle



Falta implementar 3 possíveis fontes de carga para PC: PC+4, beq e j

Via de dados completa



Sinais de controle (1 bit)

	Desligado	Ligado
RegDst	Reg de escrita rt	Reg de escrita rd
RegWR		Escreve no banco
ALUSrcA	operando é o PC	operando é o reg A
MemRD		Lê a memória
MemWR		Escreve na memória
Memto Reg	Reg WR data ALUout	Reg WR data MDR
IorD	Endereço do PC	Endereço de ALUout
IRWrite		Escreve no IR
PCWrite		Escreve no PC
PCWriteCond		Escr PC se ALU=0

Sinais de controle (2 bits)

ALUop	00	Soma
	01	Subtração
	10	Function
ALUSrcB	00	Segundo operando é o registrador B
	01	“ constante 4
	10	“ sign-extend, 16 bits do IR
	11	“ idem acima, deslocado 2 bits à esquerda
PCSrc	00	PC + 4
	01	ALUout (target address)
	10	Jump (4 bits PC & 26 bits ender & 00)

Cinco Passos de Execução

- **Busca da Instrução**
- **Decodificação da instrução e leitura dos registradores**
- **Execução, Cálculo do Endereço de Memória, Cálculo e tomada (ou não) do desvio**
- **Acesso à Memória, Término das instruções lógicas e aritméticas**
- **Escrita no registrador (para operações do tipo lw)**

Agora, instruções utilizam de 3 até 5 ciclos!

Passo 1: Busca da Instrução

- **Usar PC para obter a instrução e armazená-la no IR**
- **Incrementar o PC+4**
- **Esses passos podem ser descritos usando RTL "Register-Transfer Language"**

```
IR = Memory[PC];
PC = PC + 4;
```

Passo 2: Decodificação da Instrução e Leitura dos Regs

- **Ler registradores rs e rt**
- **Computar o endereço do desvio, caso a instrução seja um desvio**
- **RTL:**

```
A = Reg[IR[25-21]];
B = Reg[IR[20-16]];
ALUOut = PC + (sign-extend(IR[15-0]) << 2);
```

-

Passo 3: Execução

- **ULA deve executar uma das três operações seguintes**
- **Referência à Memória:**

```
ALUOut = A + sign-extend(IR[15-0]);
```

- **Tipo-R:**

```
ALUOut = A op B;
```

- **Desvio:**

```
if (A==B) PC = ALUOut;
```

Passo 4: Tipo-R ou Acesso à Memória

- **Loads e stores acessam a memória**

```
MDR = Memory[ALUOut];
      or
Memory[ALUOut] = B;
```

- **Instruções do tipo-R terminam a execução**

```
Reg[IR[15-11]] = ALUOut;
```

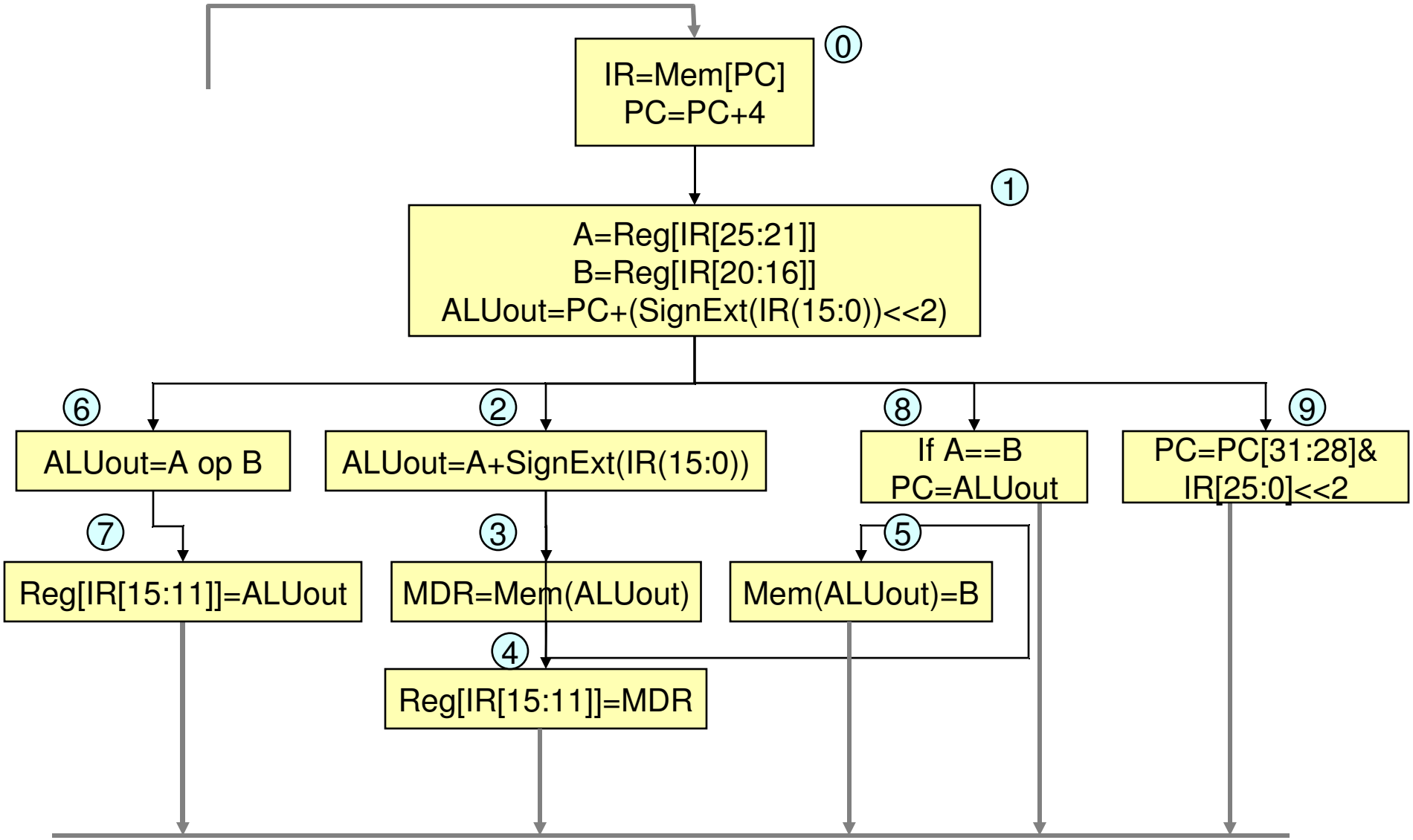
Passo 5: Escrita no registrador

- `Reg[IR[20–16]]= MDR;`

Resumo:

Step name	Action for R-type instructions	Action for memory-reference instructions	Action for branches	Action for jumps
Instruction fetch	IR = Memory[PC] PC = PC + 4			
Instruction decode/register fetch	A = Reg [IR[25-21]] B = Reg [IR[20-16]] ALUOut = PC + (sign-extend (IR[15-0]) << 2)			
Execution, address computation, branch/ jump completion	ALUOut = A op B	ALUOut = A + sign-extend (IR[15-0])	if (A ==B) then PC = ALUOut	PC = PC [31-28] II (IR[25-0]<<2)
Memory access or R-type completion	Reg [IR[15-11]] = ALUOut	Load: MDR = Memory[ALUOut] or Store: Memory [ALUOut] = B		
Memory read completion		Load: Reg[IR[20-16]] = MDR		

Outra visão, fluxograma



Questões rápidas

- **O código a seguir executará em quantos ciclos?**

```
            lw $t2, 0($t3)
            lw $t3, 4($t3)        #considere que não será tomado
            beq $t2, $t3, Label
            add $t5, $t2, $t3
            sw $t5, 8($t3)
Label:      ...
```

- **O que está acontecendo durante o 8º ciclo?**
- **Em qual ciclo acontece a soma $t2+$t3?**