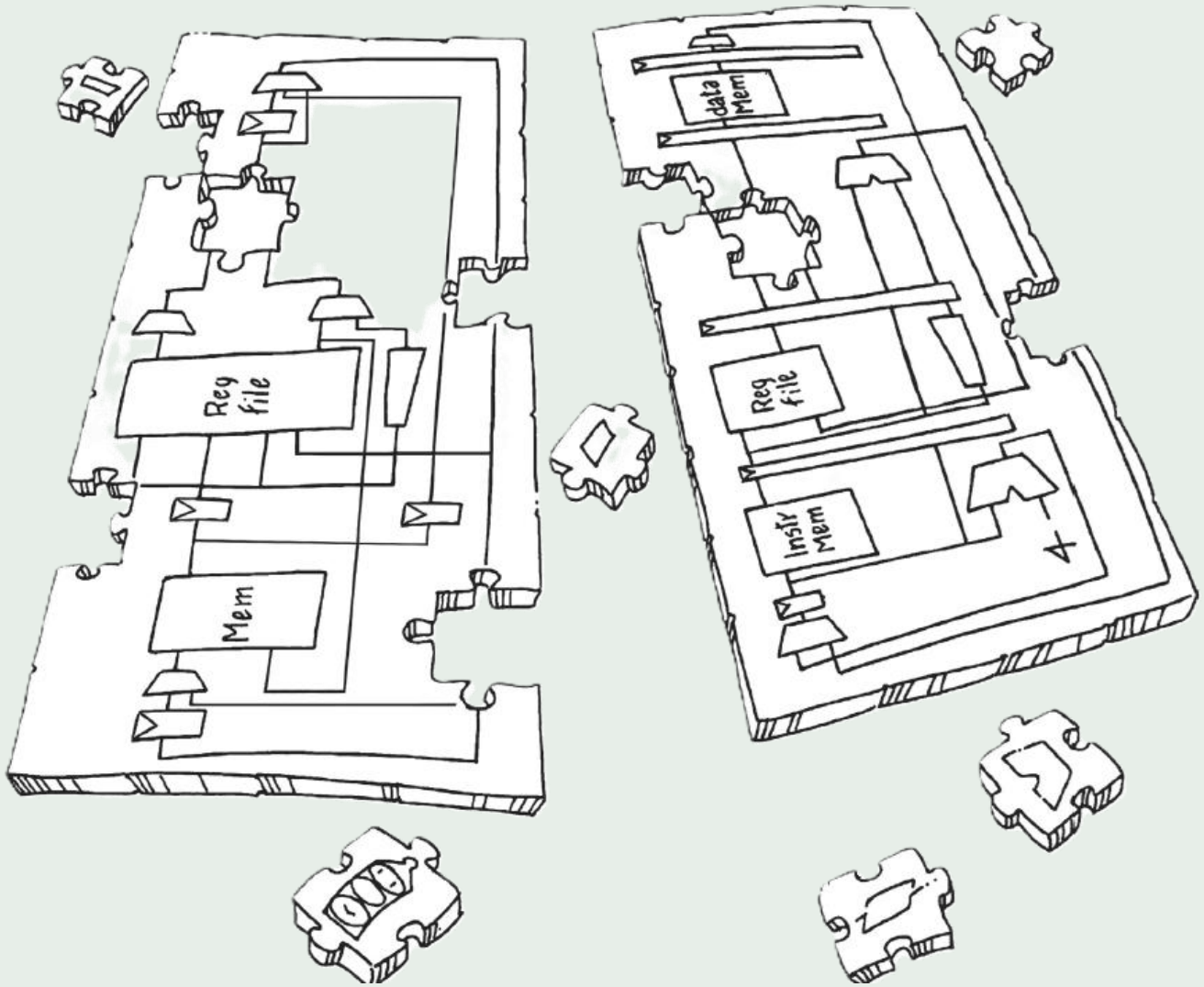


Fonte: Adaptado de Digital Design and Computer Architecture RISC-V Edition Sarah L. Harris David Harris

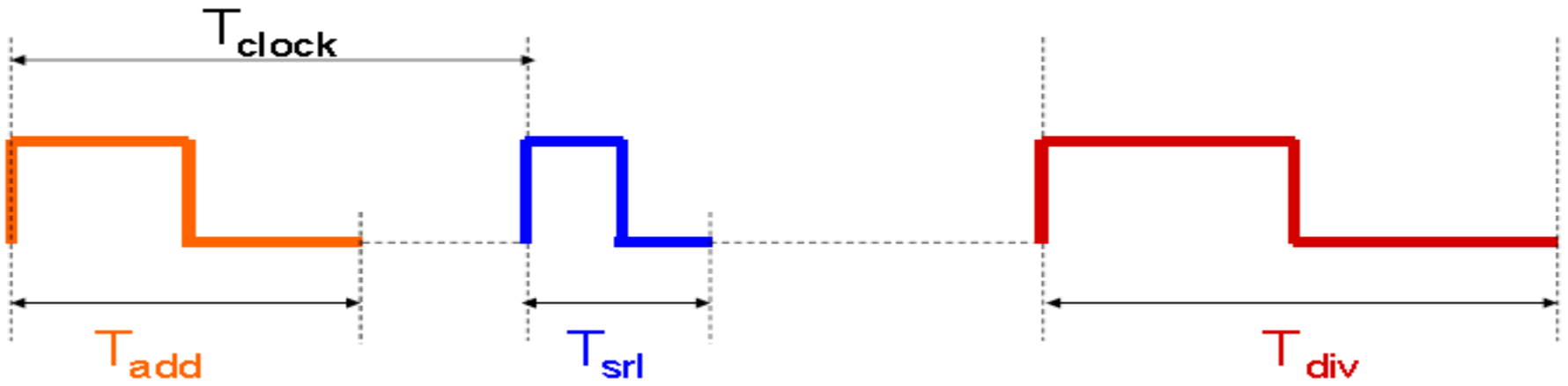
Arquitetura e Organização de Computadores 1

Prof. Luciano de Oliveira Neris
luciano@dc.ufscar.br

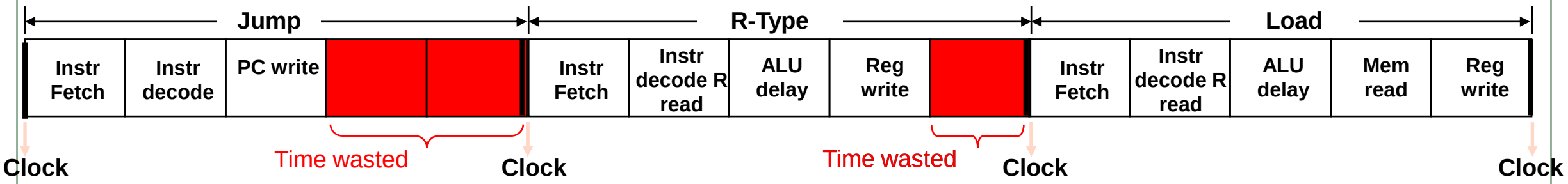
Datapath e Controle Multiciclo para RISC-V

Datapath Monociclo

- Cada instrução é executada em um 1 ciclo de clock
- Ciclo de clock deve ser longo o suficiente para executar a instrução mais longa
- Desvantagem: velocidade global limitada à velocidade da instrução mais lenta



Datapath Monociclo



Fonte: Prof. Savio chau: CSM151B

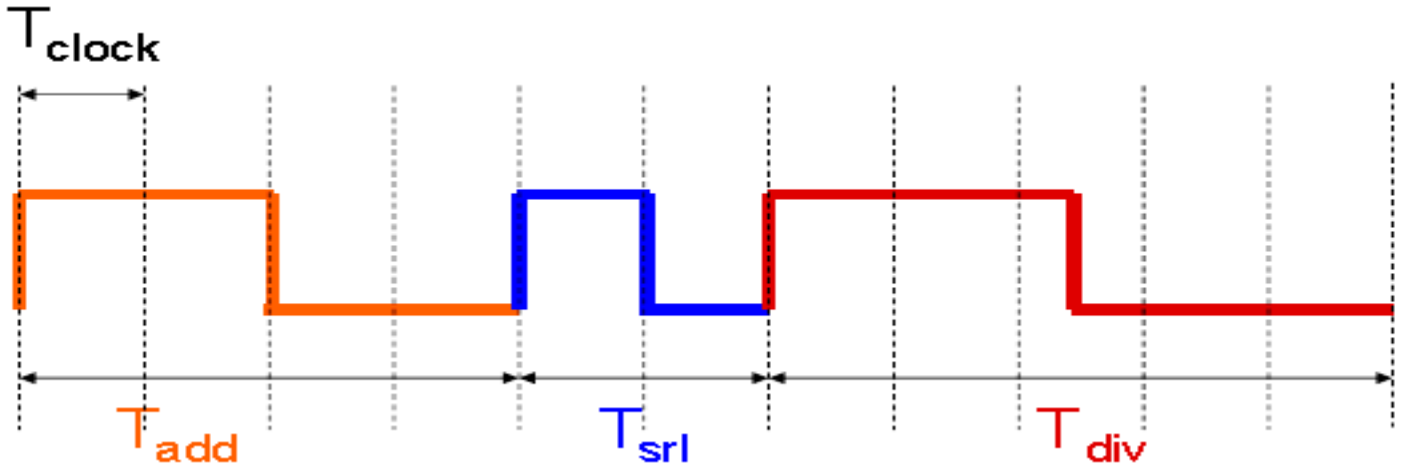
Datapath Monociclo

- Viola o princípio de "tornar o caso comum mais rápido"
- Datapaths monociclo não são mais utilizados em processadores modernos
- É mais eficiente executar cada instrução em um número variável de ciclos mais rápidos, utilizando apenas o necessário
- Este é o principio básico do datapath multiciclo

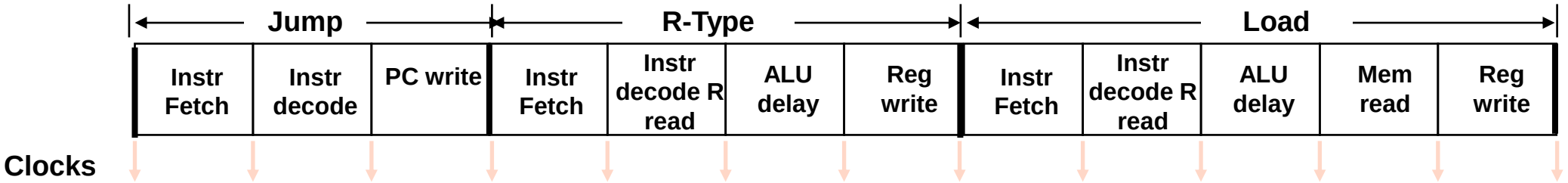
Multiciclo

Datapath Multiciclo

- Quebra o ciclo de execução em vários passos
- Executa cada passo em um ciclo de clock
- Cada instrução usa apenas o número de ciclos que ela necessita



Datapath Multiciclo



Fonte: Prof. Savio chau: CSM151B

Datapath Multiciclo

Vantagens:

- Redução no tempo médio de execução de cada instrução
- Uma mesma unidade funcional pode ser utilizada em ciclos distintos de uma mesma instrução (ou seja, utilizada mais de uma vez).
  - * *Utilizar multiplexadores para determinar a origem dos dados*
- Pergunta: Como subdividir o datapath / instruções?

Datapath Multiciclo

- Esquema geral: partindo do datapath monociclo, acrescentar registradores temporários para armazenar valores entre as diversas unidades funcionais utilizadas por uma instrução.
  - Esses registradores são "invisíveis" ao programador
  - Sua função é evitar perda de sincronização nas transições de clock
- Dessa forma, subdivide-se o ciclo longo por uma sequencia de ciclos mais curtos

Datapath Multiciclo

- Uma única memória para instruções e dados
- Apenas uma ULA (dispensa os uso de somadores extras)
- Dados a serem usados na mesma instrução em um ciclo de relógio posterior ficam armazenados nos registradores não-visíveis ao programador
- Dados a serem usados em outras instruções devem ser armazenados em elementos de memória visíveis ao programador (banco de registradores, PC ou memória)

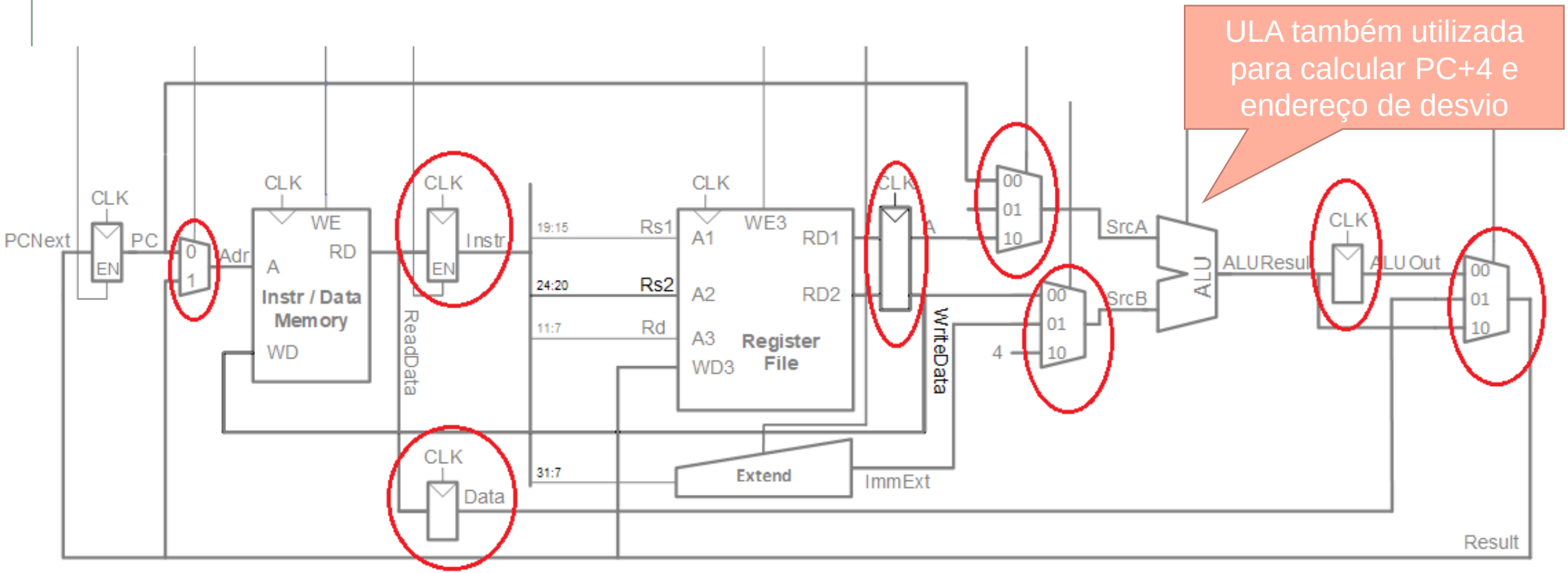
Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- IR (registrador de instrução) – usado para guardar a saída da memória para uma leitura de instrução
- MDR (registrador de dados da memória) – usado para guardar a saída da memória para uma leitura de dados
- A e B – usados para conter os valores dos registradores operandos lidos do banco de registradores
- ALUOut – contém a saída da ALU

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

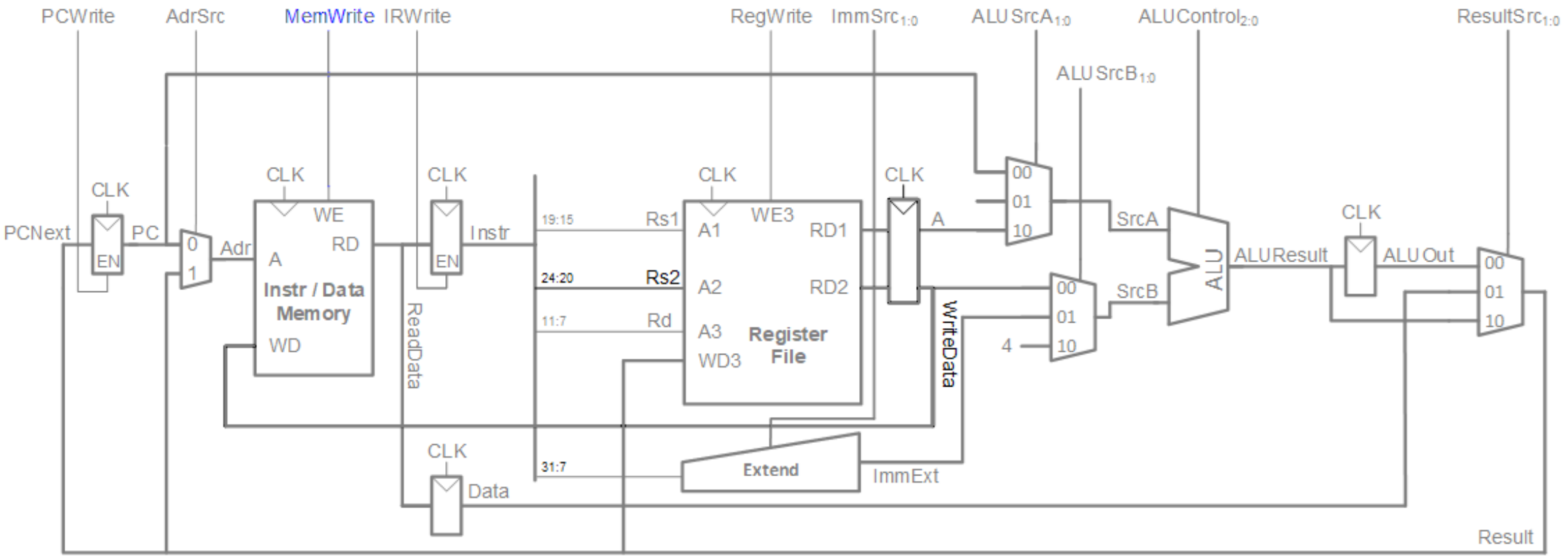
Datapath Multiciclo



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Sinais de Controle



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Memória única:
  - Requer um MUX para selecionar se o endereço de acesso à memória vem de PC (instrução) ou de SaídaALU (dados)
- ALU única:
  - Um MUX adicional é incluído na primeira entrada para escolher entre o registrador A ou o PC
  - O MUX da segunda entrada da ALU é expandido para três entradas, a fim de poder selecionar a constante 4 (incremento do PC) e o campo offset estendido e deslocado (desvios)

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

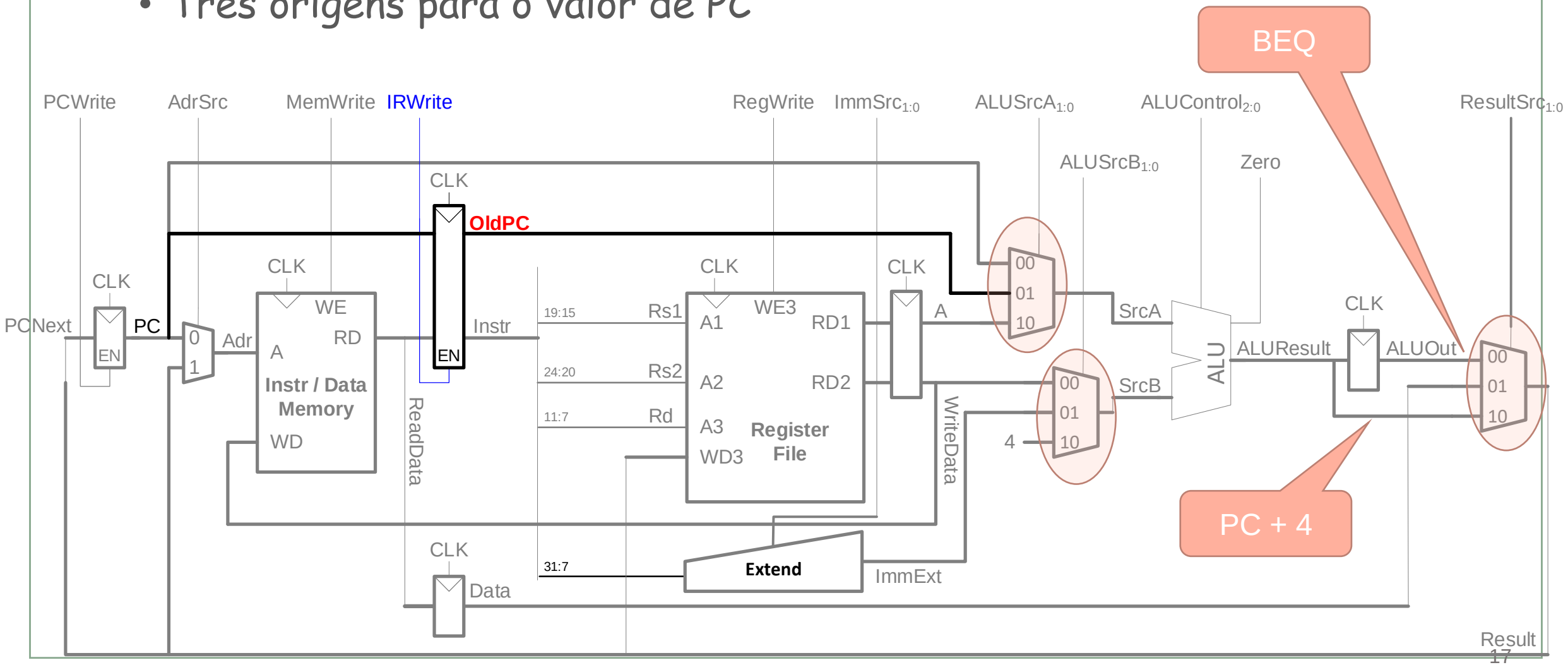
Datapath Multiciclo

- Duas origens para o valor de PC:
  - Saída da ALU (PC + 4): este valor sempre será armazenado no PC
    - ALUSrcA = 00
    - ALUSrcB = 10
    - ResultSrc =10
  - Registrador ULASaída, onde é armazenado o endereço de desvio, após ele ser calculado: este registrador armazena o endereço-alvo do desvio condicional, após este ter sido calculado pela ULA (beq)
    - ALUSrcA = 01
    - ALUSrcB = 01
    - ResultSrc =00

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

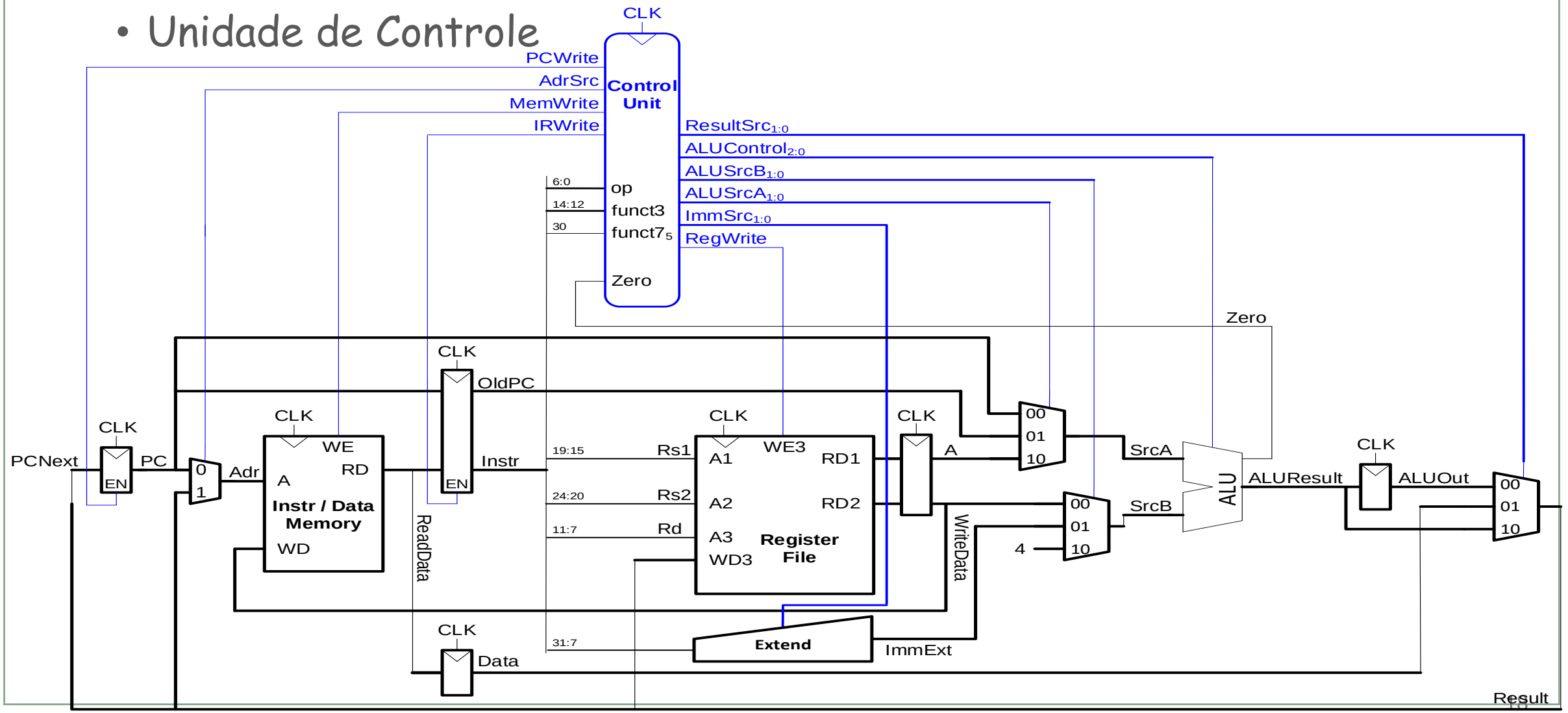
- Três origens para o valor de PC



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

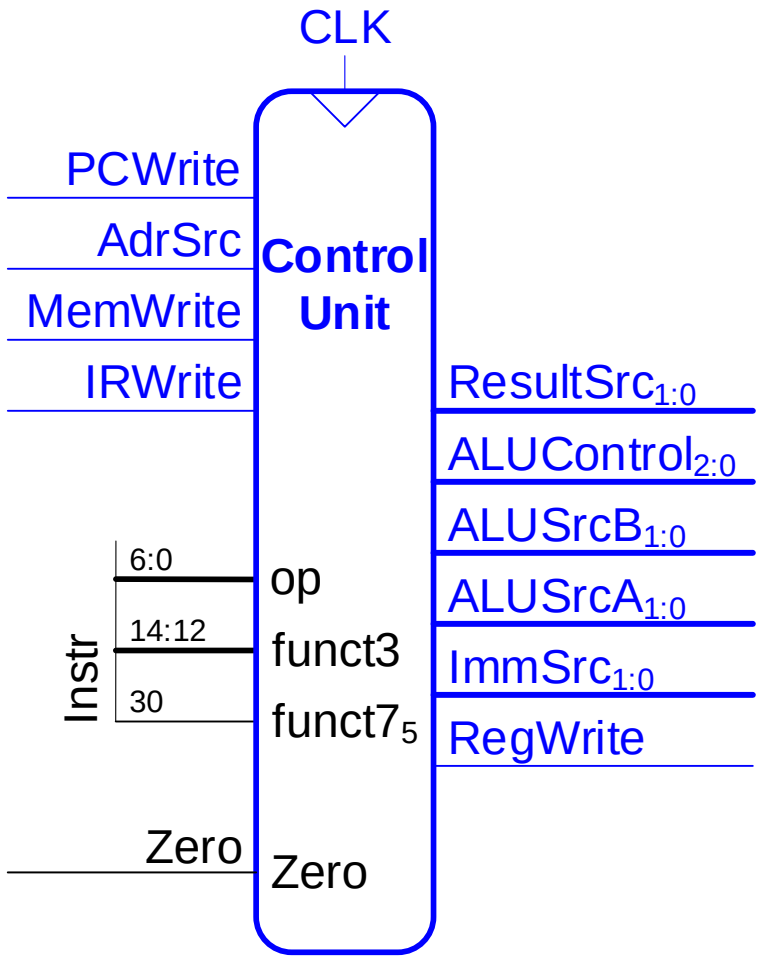
- Unidade de Controle

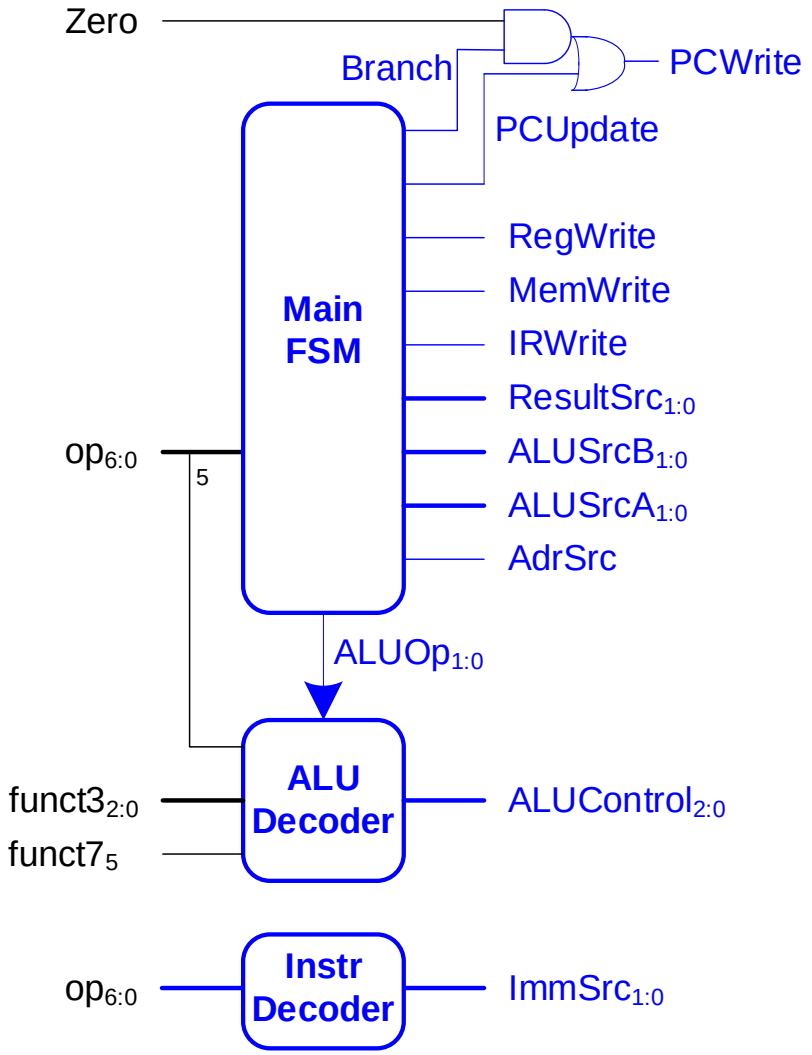


Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Unidade de Controle

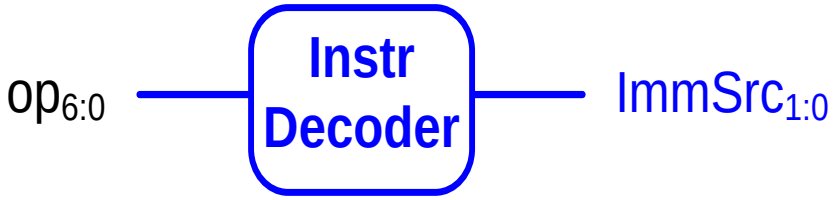




Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Unidade de Controle



op	Instruction	ImmSrc
3	`lw`	00
35	`sw`	01
51	R-type	XX
99	`beq`	10

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

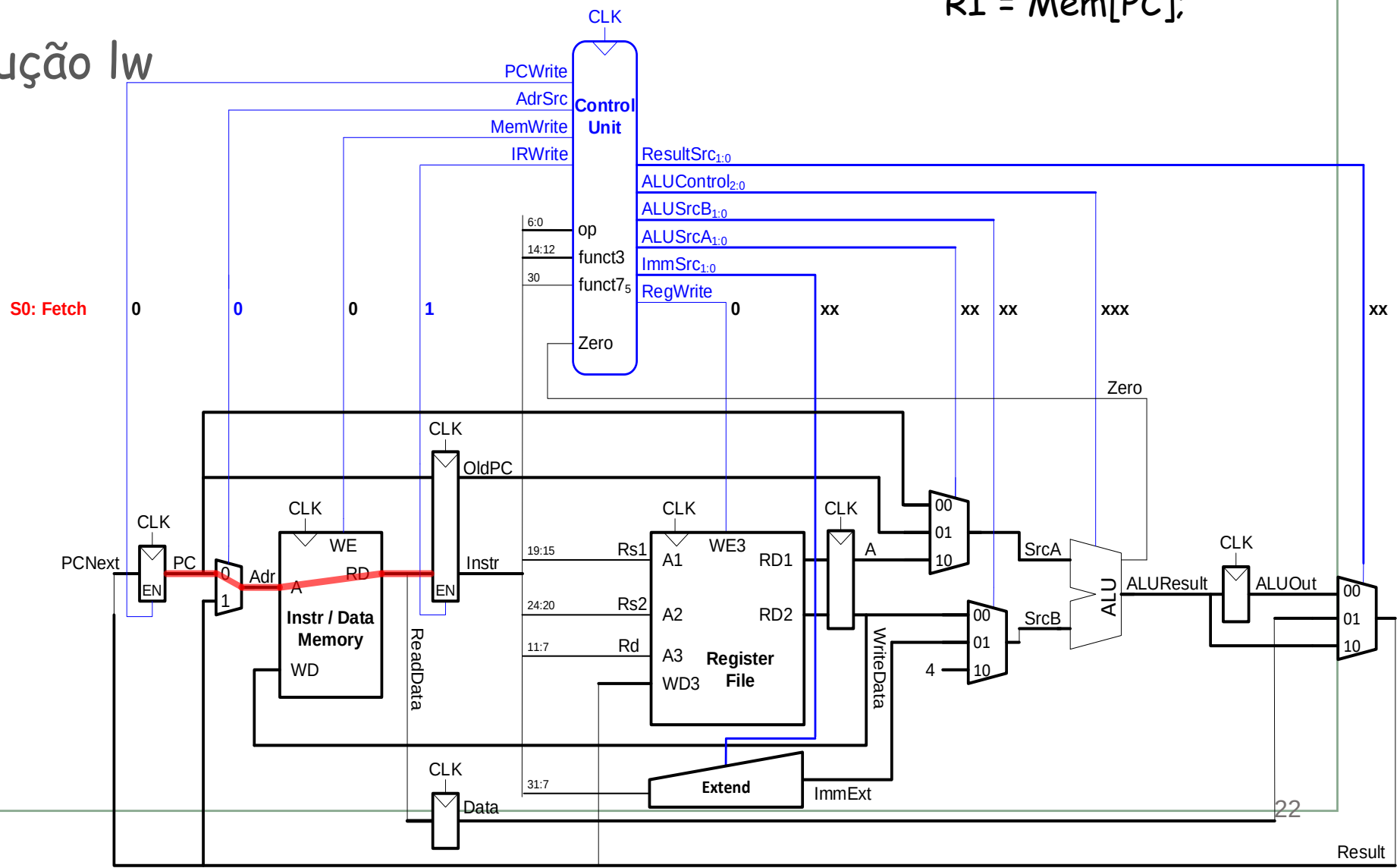
- Passos (etapas) das Instruções
  - 0. Busca da instrução
  - 1. Decodificação da instrução
    - Leitura dos registradores – mesmo que não sejam utilizados
    - Extensão do imediato – mesmo que não sejam utilizados
  - 2. Execução da operação
    - Instruções tipo R, tipo I e tipo B
    - Cálculo do endereço efetivo do operando – instruções load e store
    - Determinar se salto deve ser executado – instrução branch
  - 3. Acesso à memória
    - Instruções load e store
    - Escrita de registrador – instruções tipo R, tipo I e tipo J
  - 4. Escrita de registrador
    - Instrução load

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

0. Busca da Instrução lw

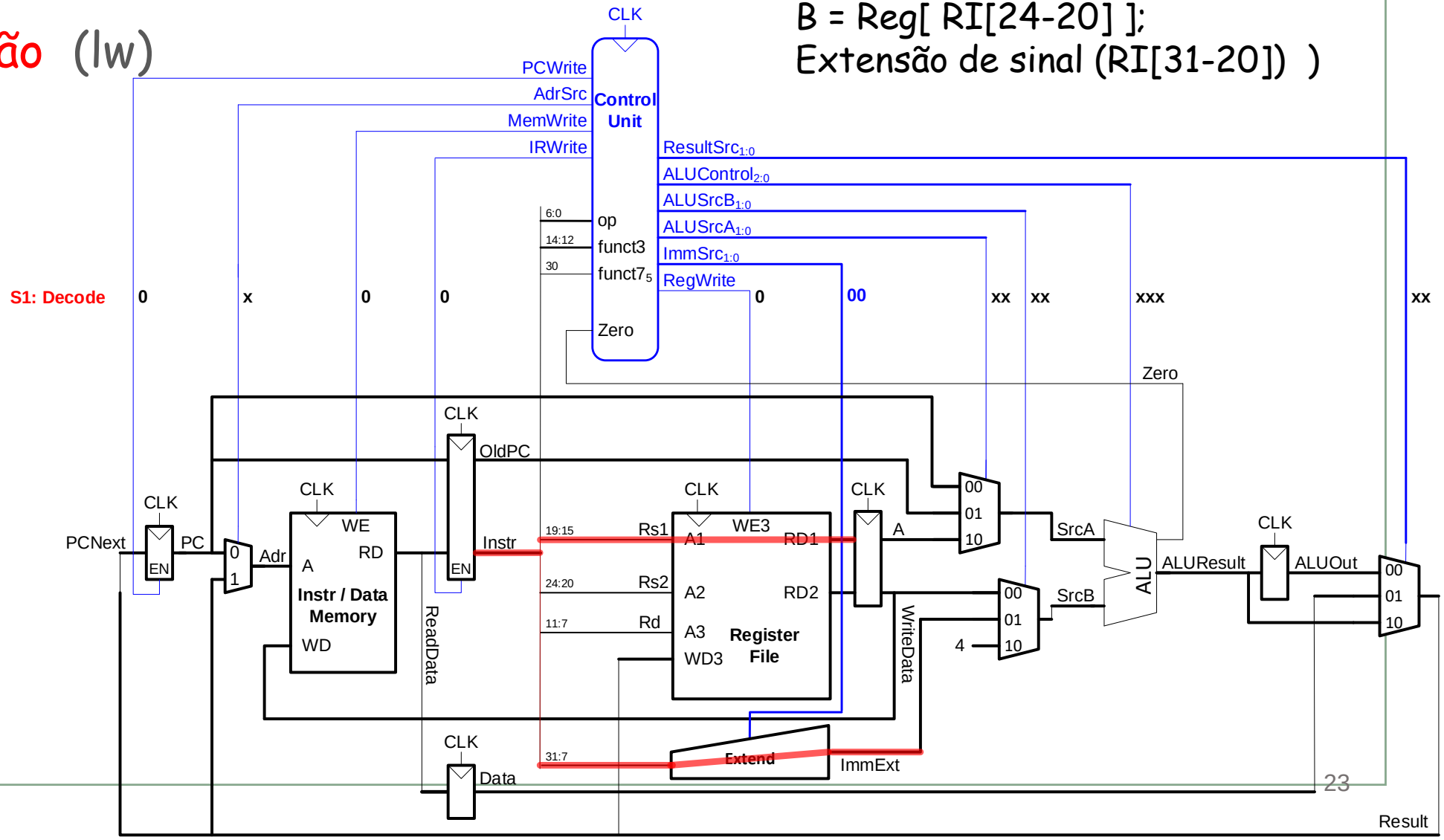
RI = Mem[PC];



Datapath Multiciclo

1. Decodificação (lw)

Geração dos Sinais de Controle
A = Reg[RI[19-15]];
B = Reg[RI[24-20]];
Extensão de sinal (RI[31-20]))

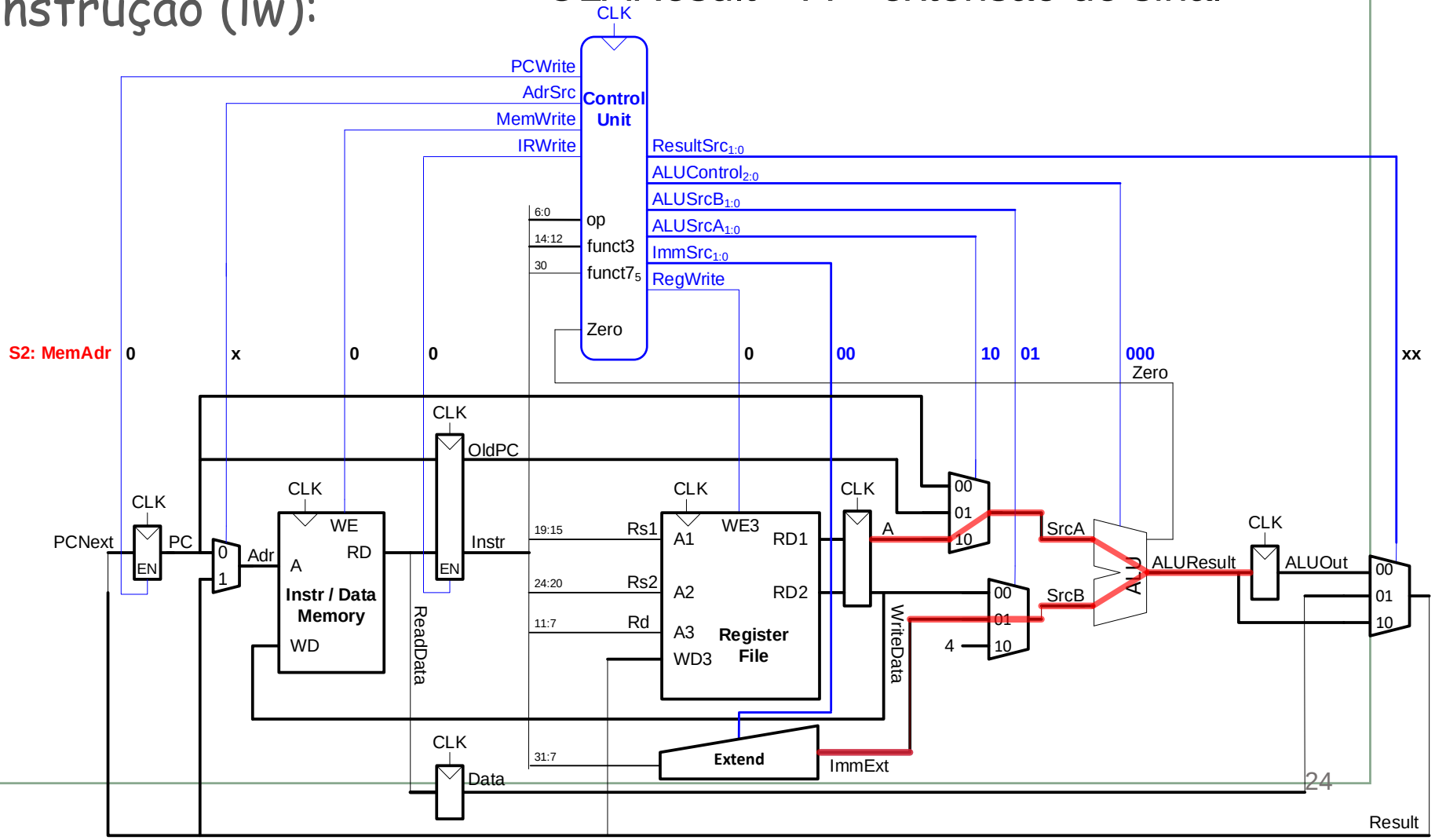


Datapath Multiciclo

- 2. Execução da Instrução (lw):

Cálculo do endereço

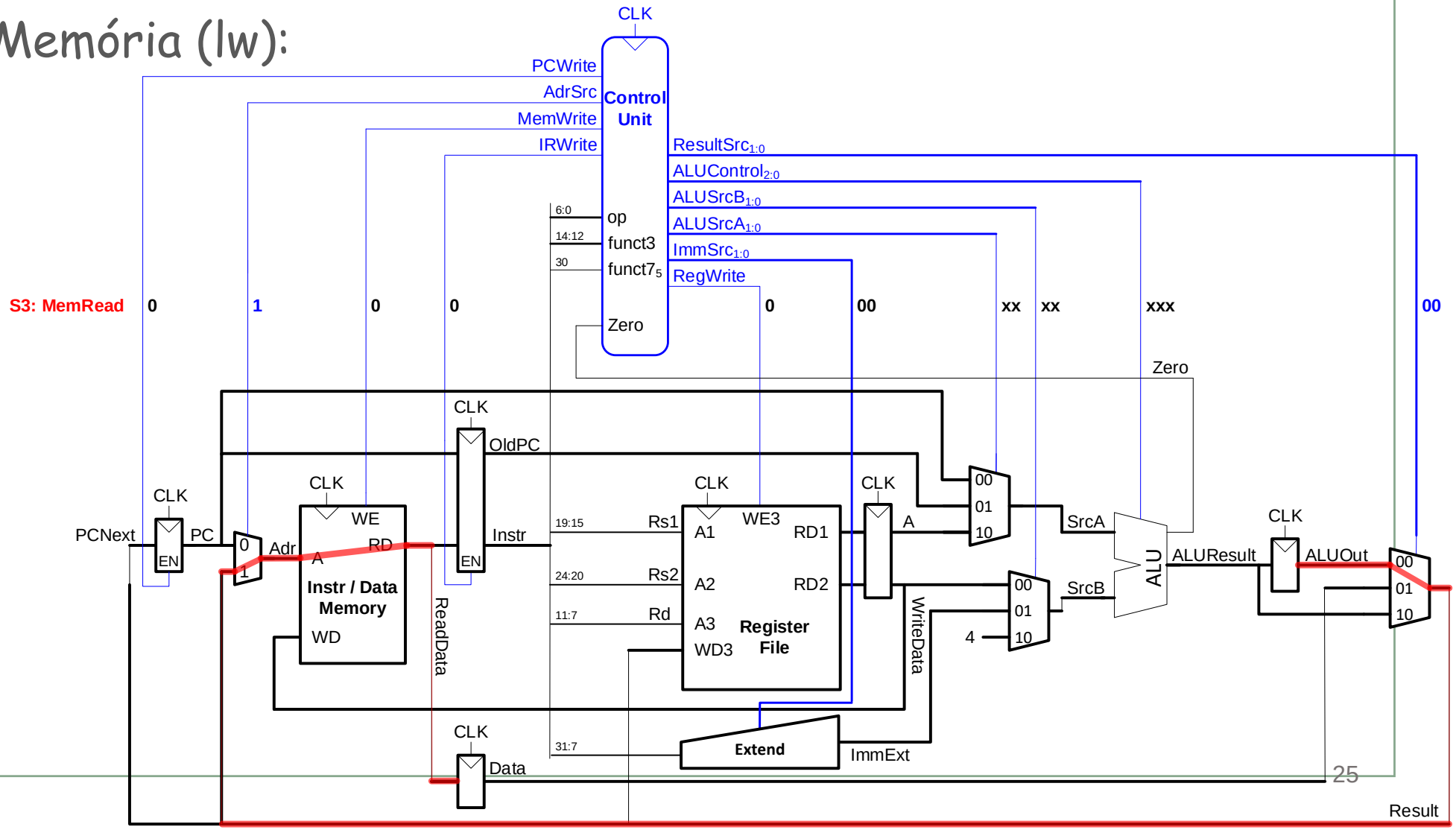
ULAResult = A + extensão de sinal



Datapath Multiciclo

Data = Mem[ULAOut];

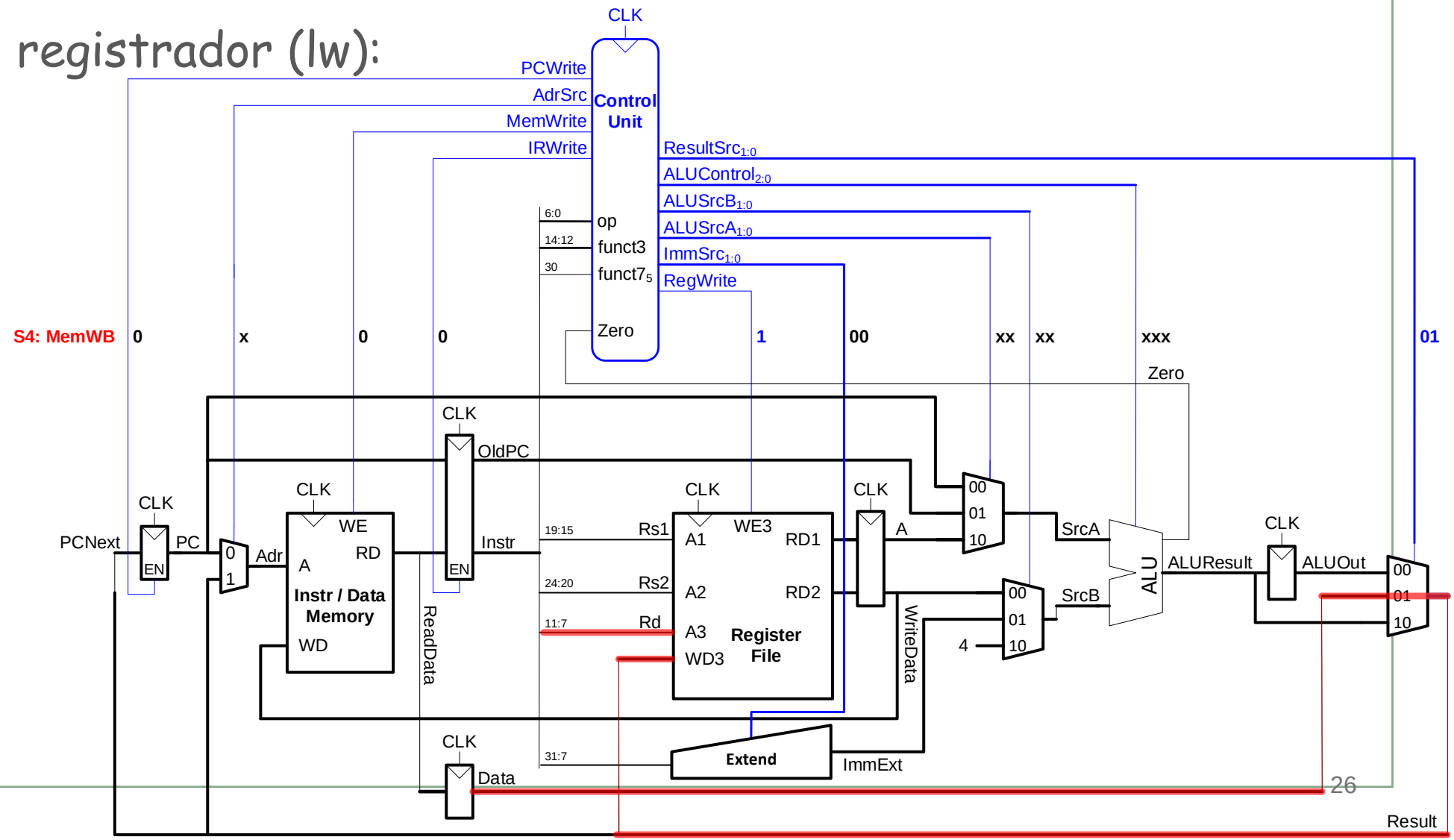
- 3. Acesso à Memória (lw):



Datapath Multiciclo

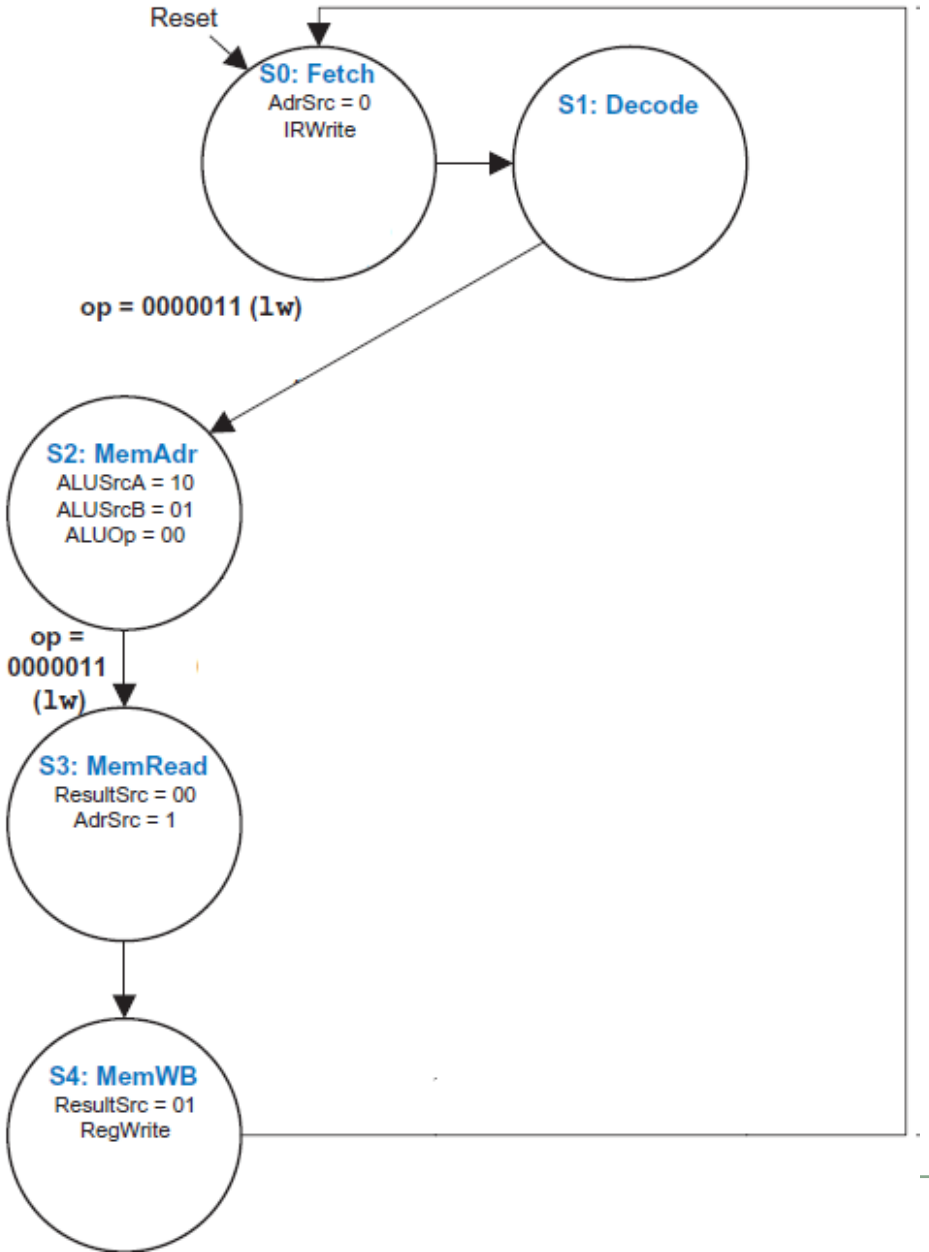
Reg[rd] = data

- 4. Escrita no registrador (lw):



Datapath Multiciclo

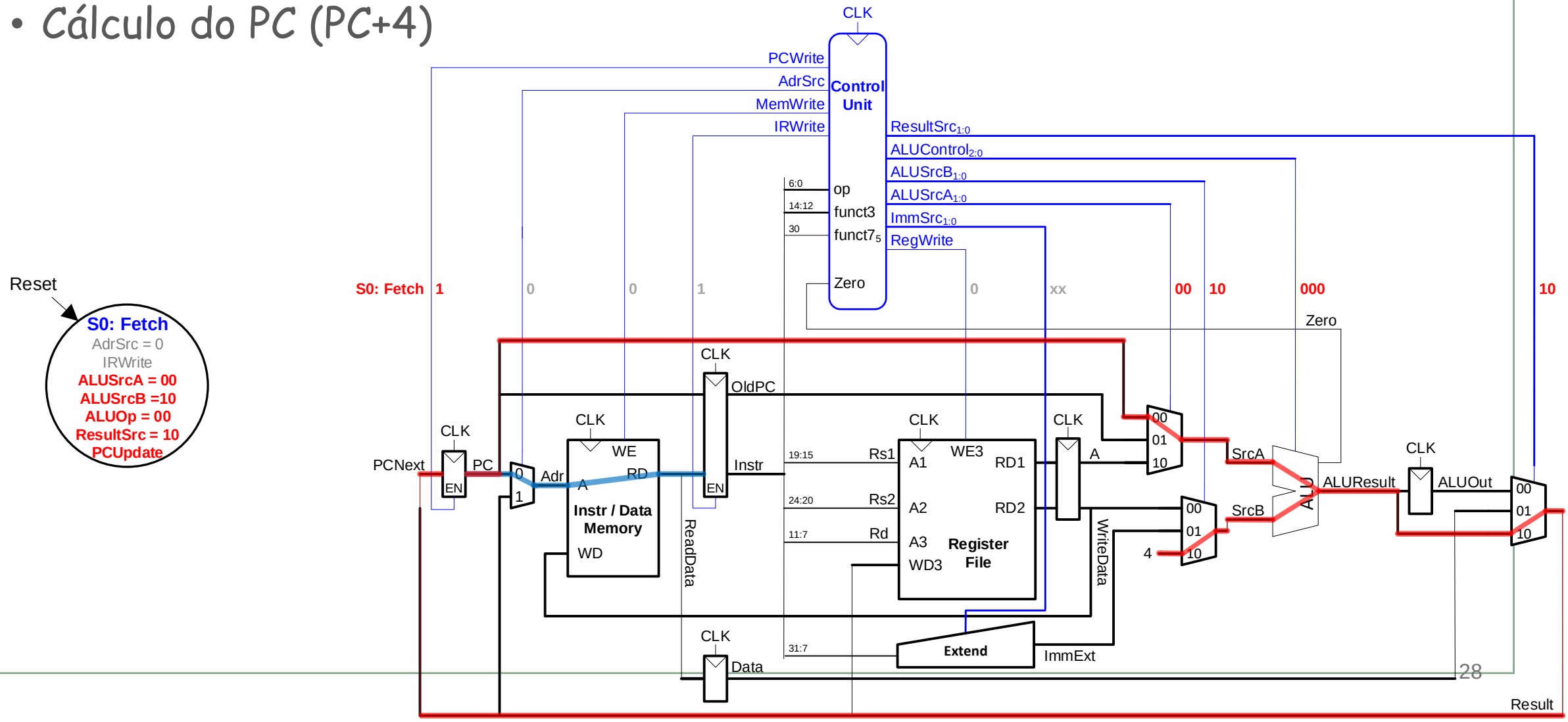
- Máquina de Estados (lw)



Datapath Multiciclo

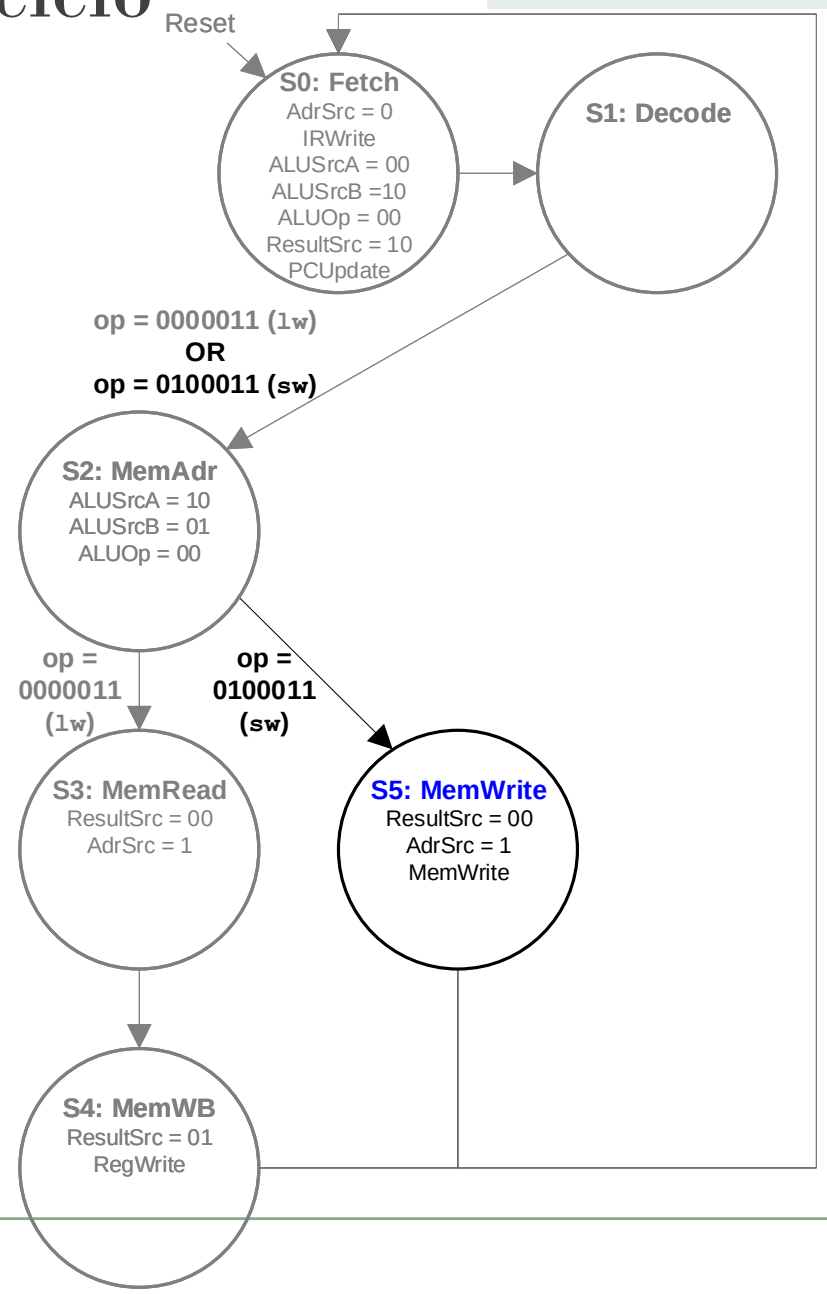
PC = PC+4

- Cálculo do PC (PC+4)



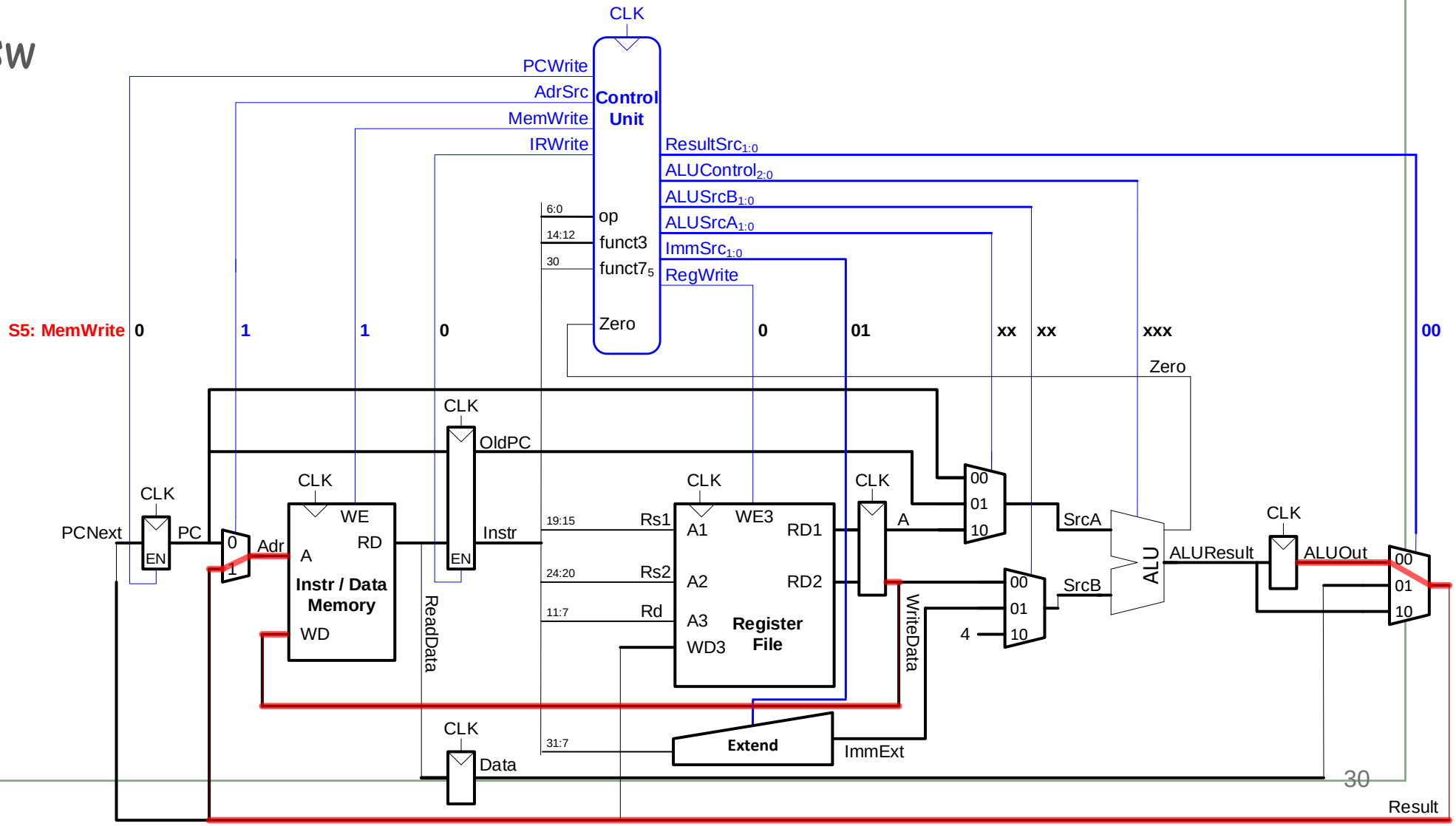
Datapath Multiciclo

- Instrução sw



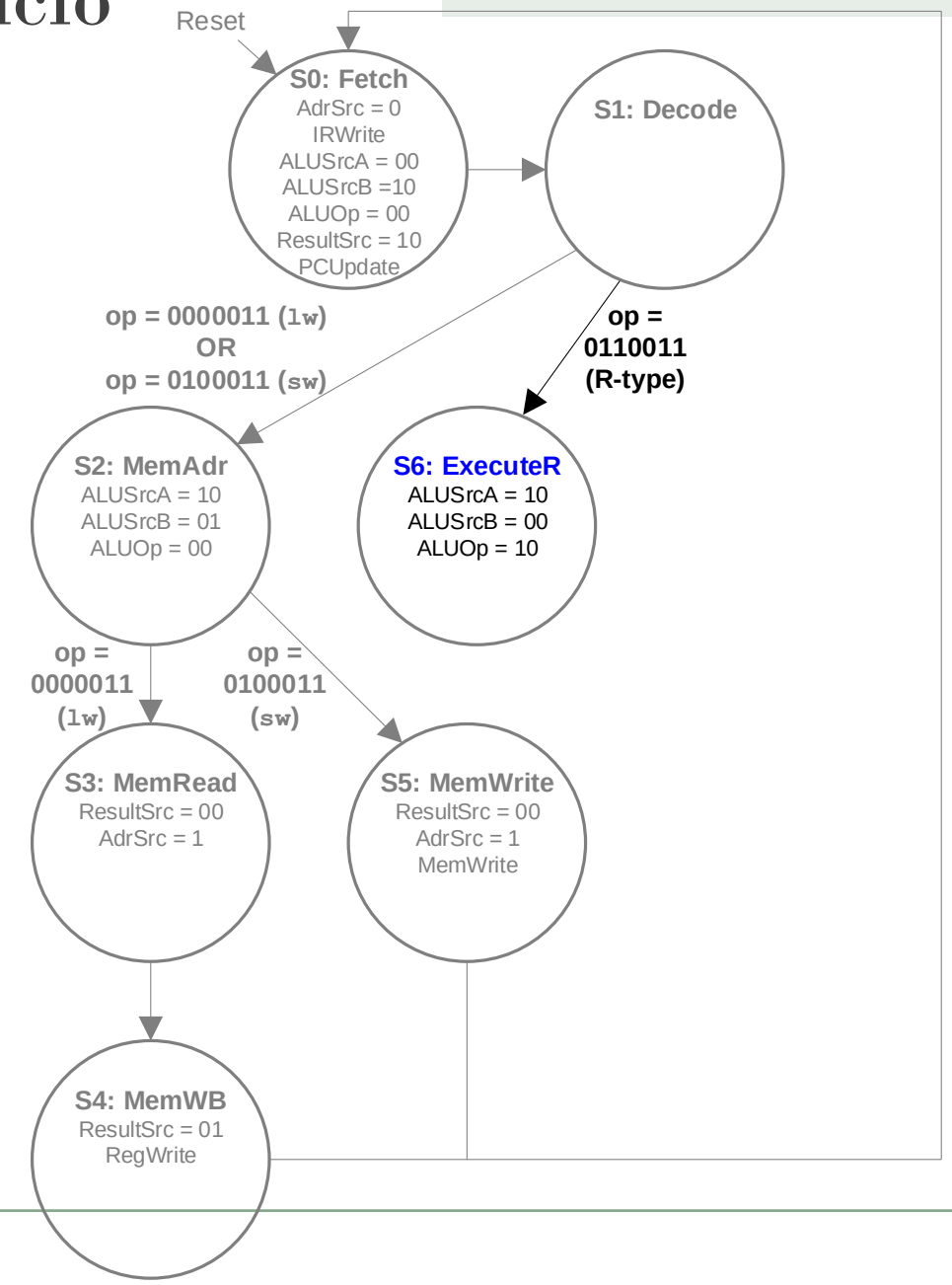
Datapath Multiciclo

- Instrução sw



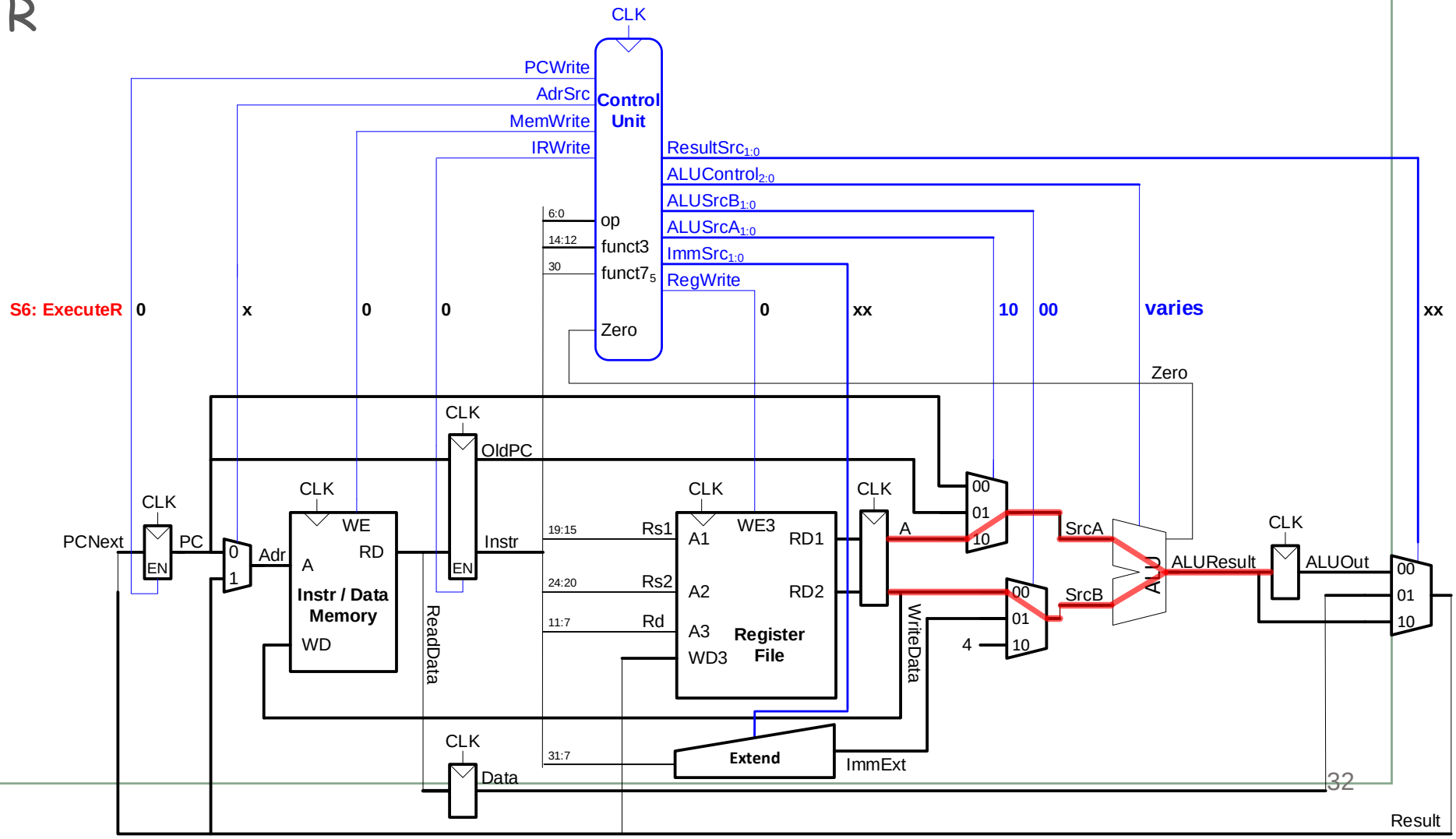
Datapath Multiciclo

- Instrução Tipo R



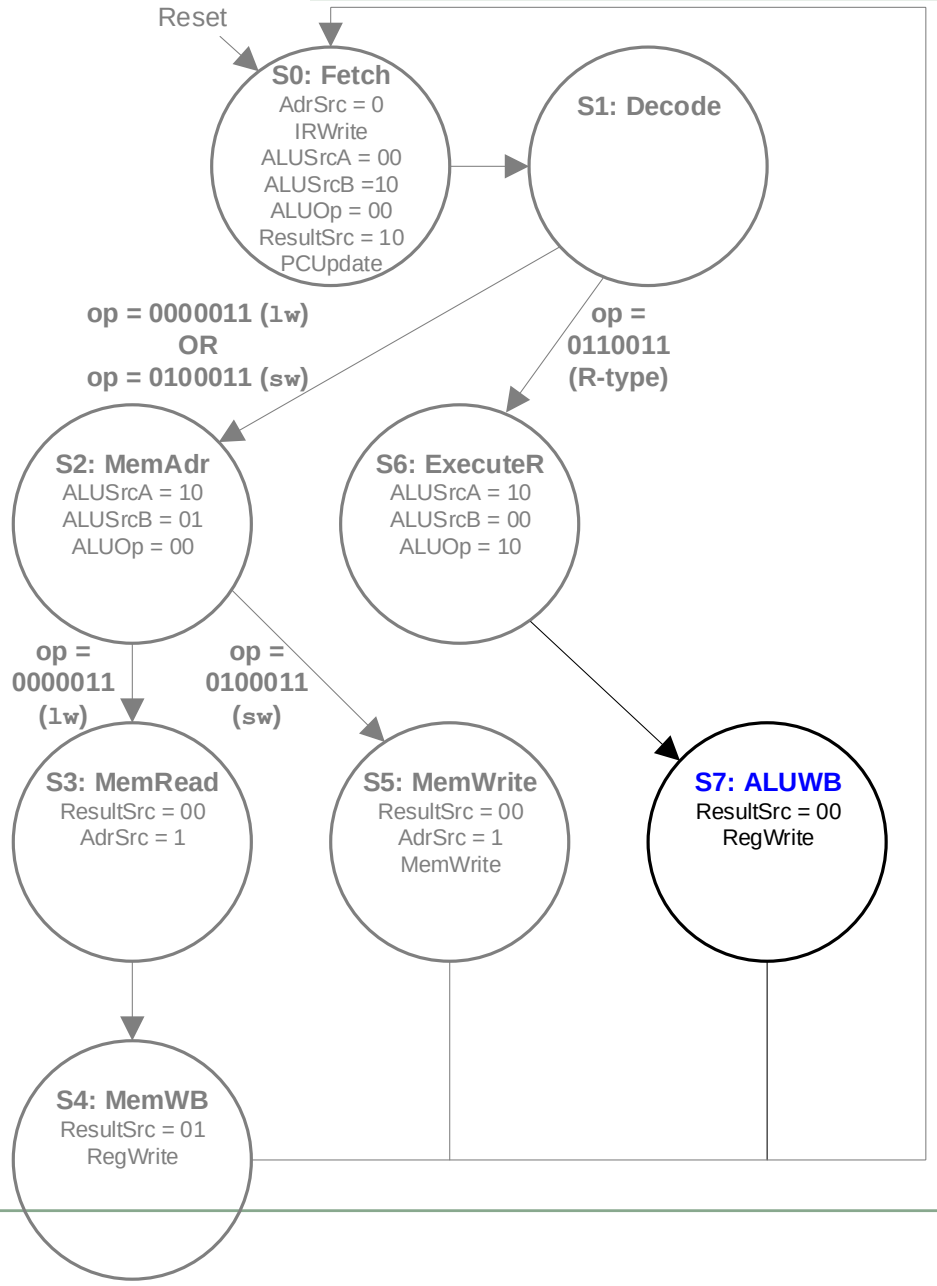
Datapath Multiciclo

- Instrução Tipo R



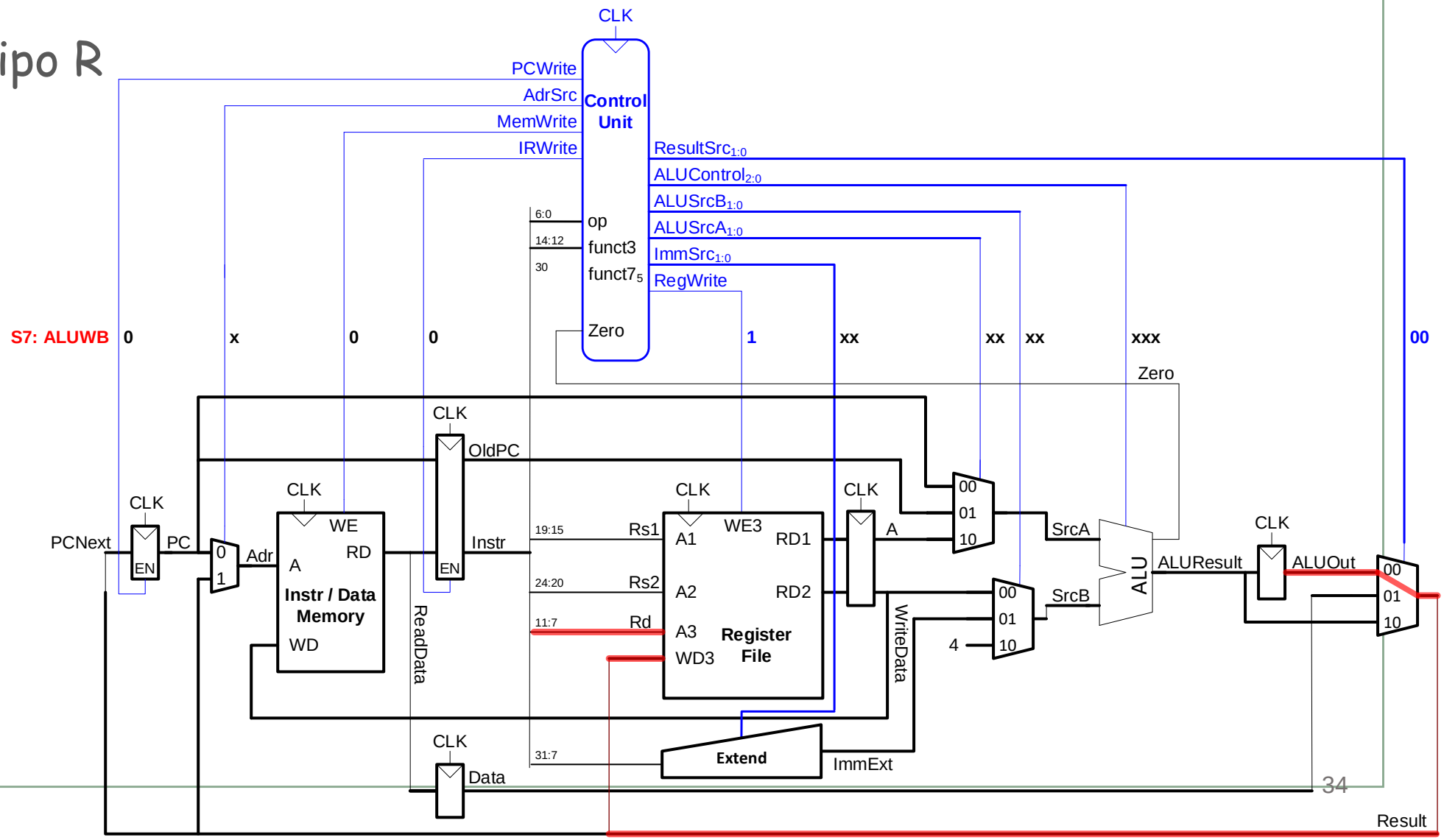
Datapath Multiciclo

- Instrução Tipo R



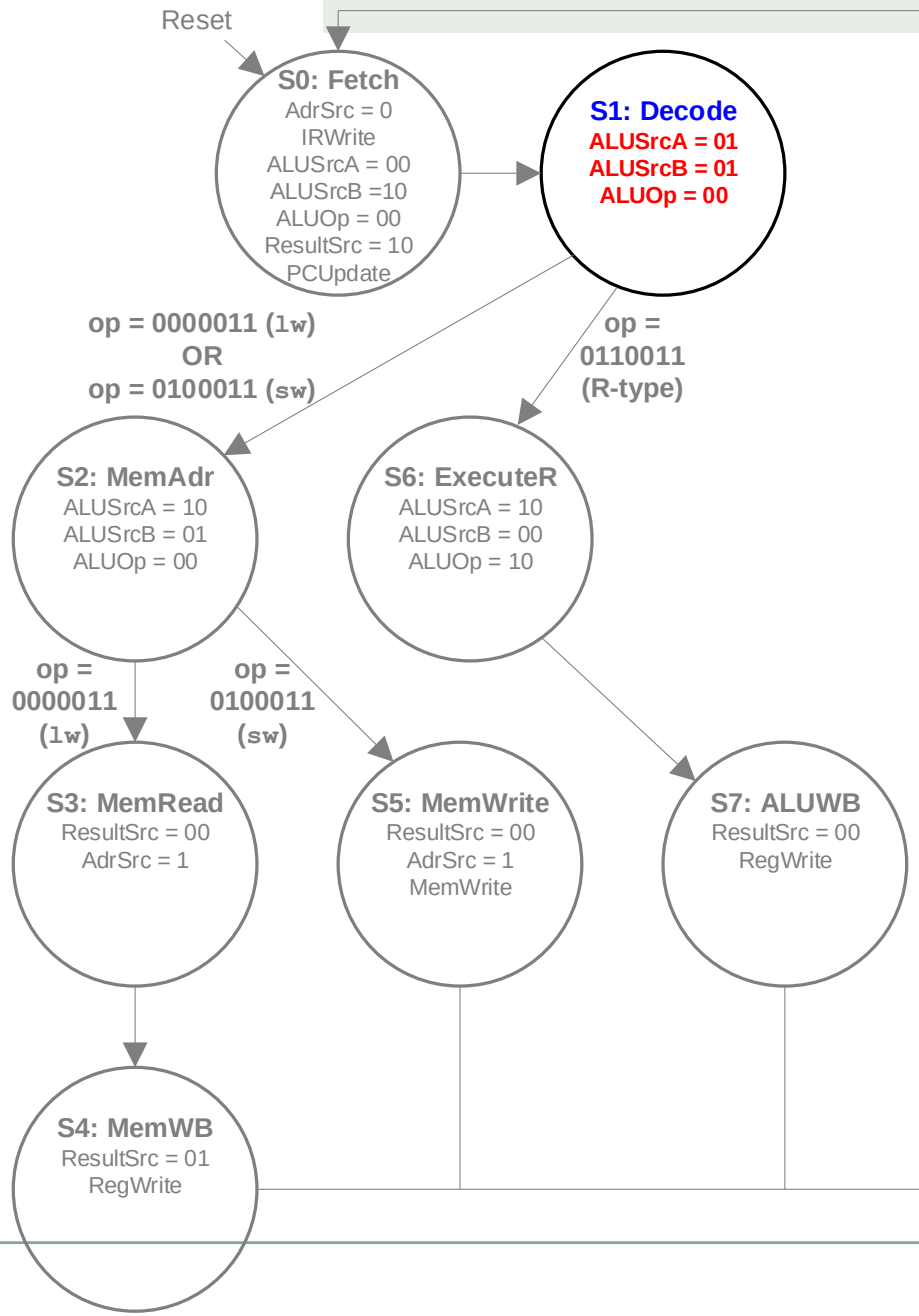
Datapath Multiciclo

- Instrução Tipo R



Datapath Multiciclo

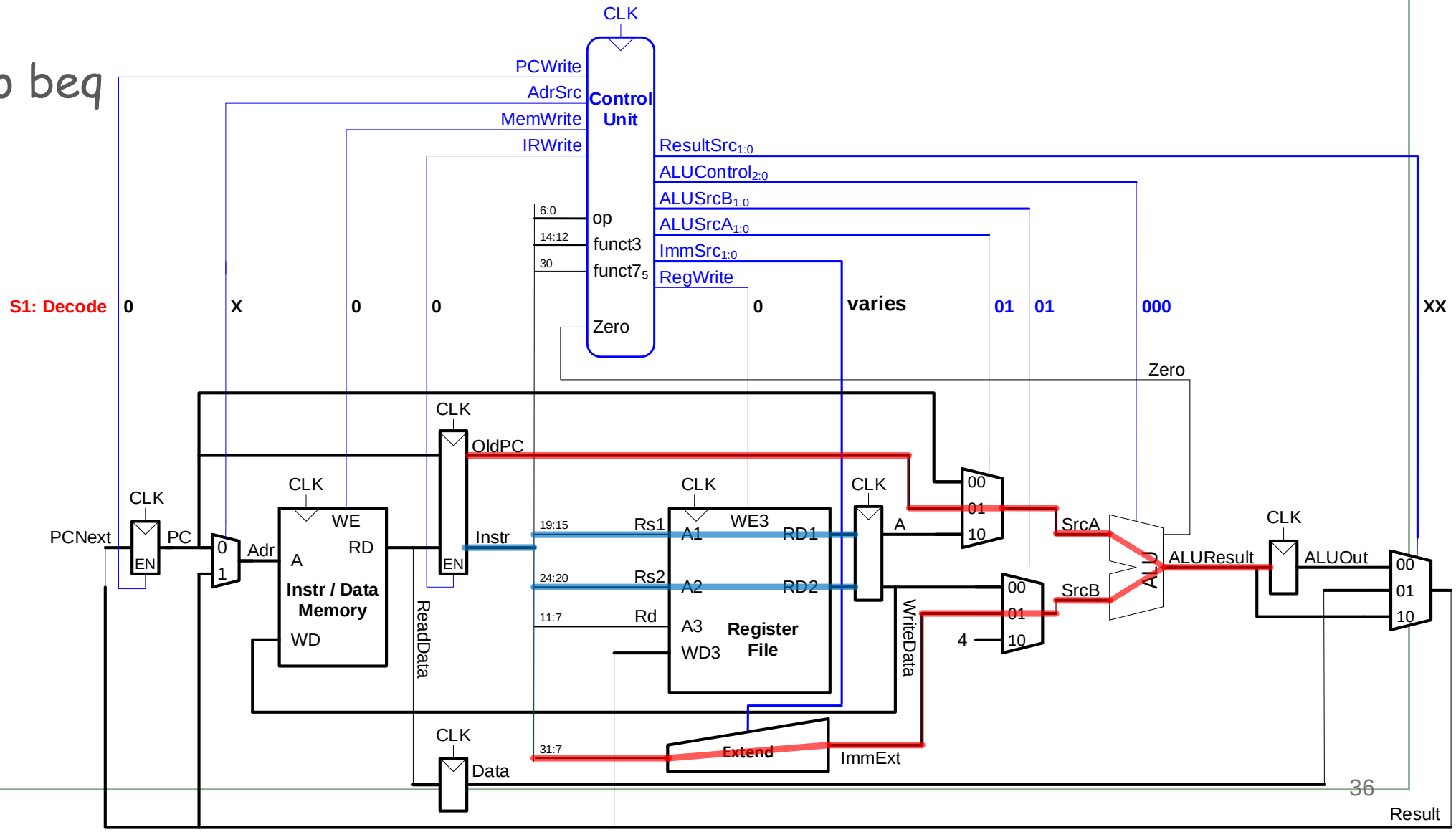
- Instrução beq



Ler Registrador e Calcula o endereço de salto (PC+imm)

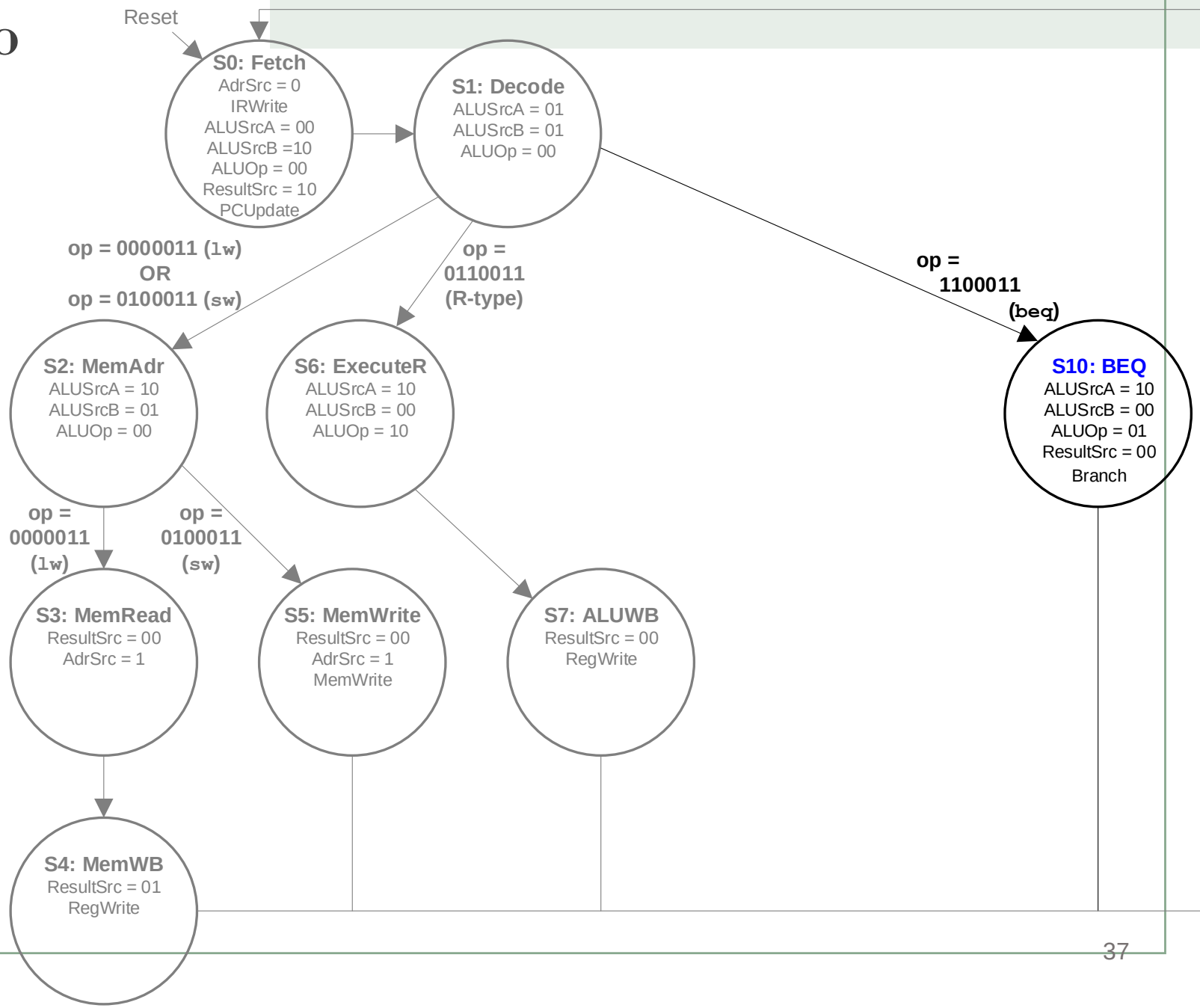
Datapath Multiciclo

- Instrução beq



Datapath Multiciclo

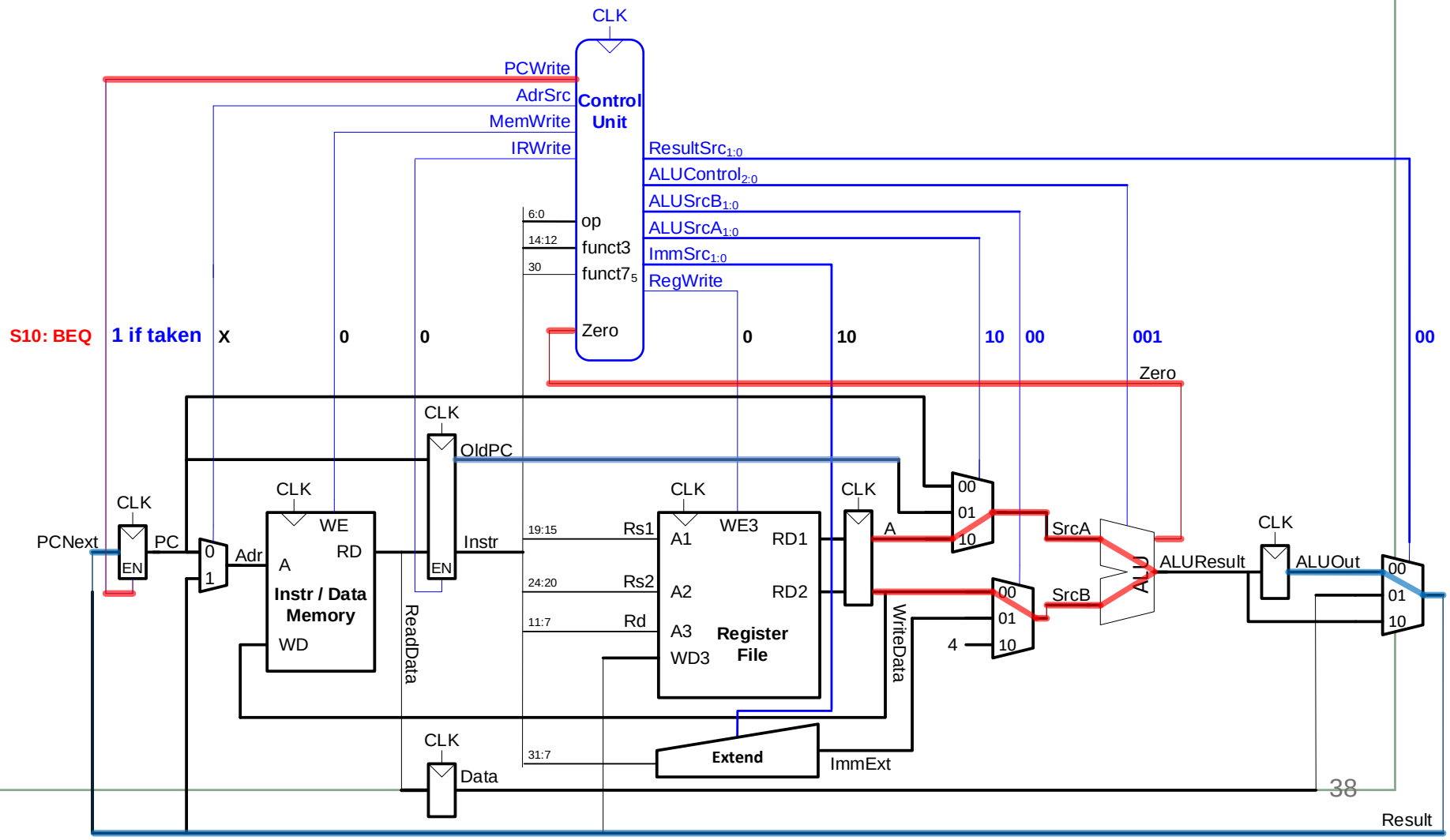
- Instrução beq



Compara registradores e
Envia o PC Alvo (ALUOut) para o PCNext

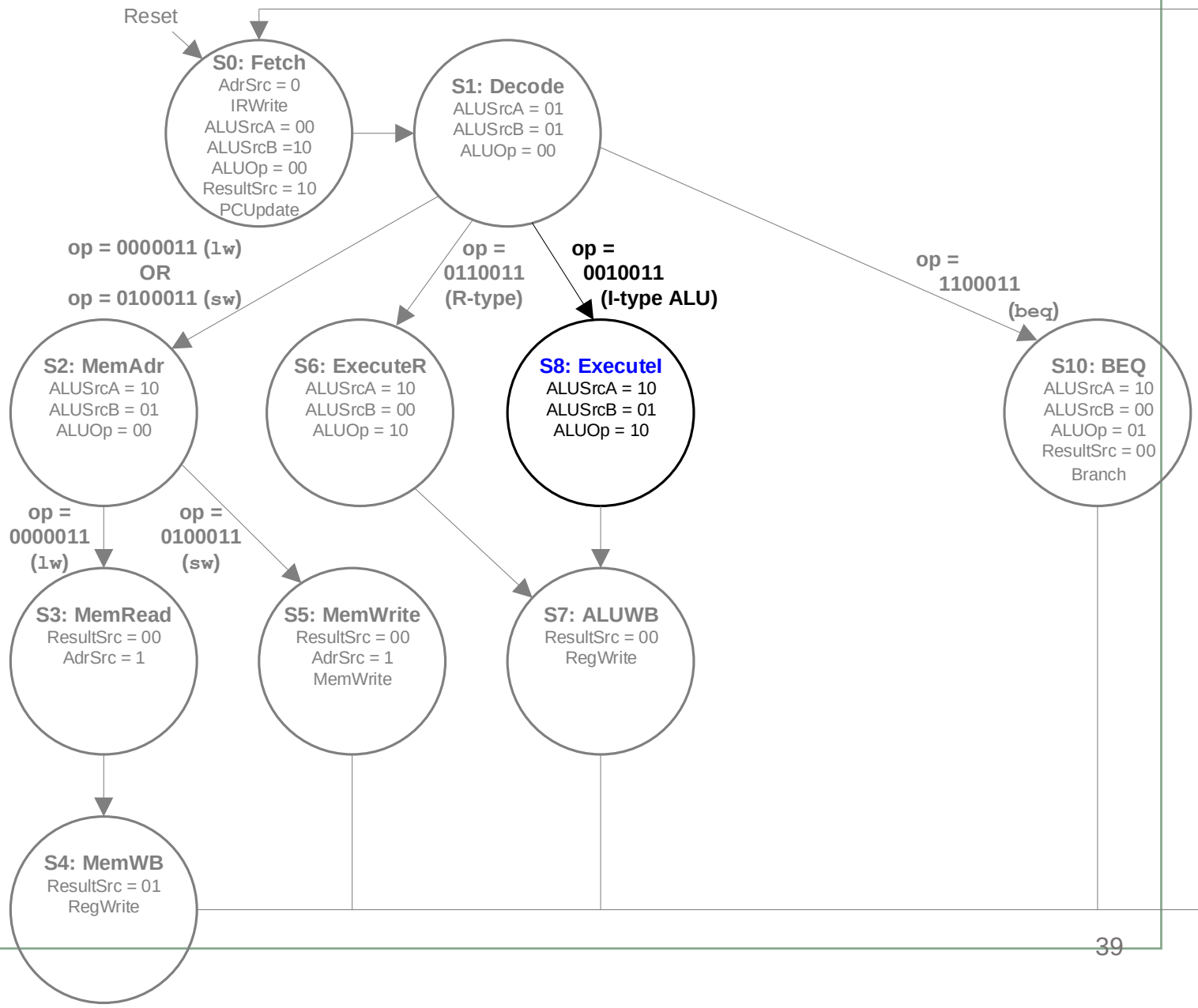
Datapath Multiciclo

- Instrução beq



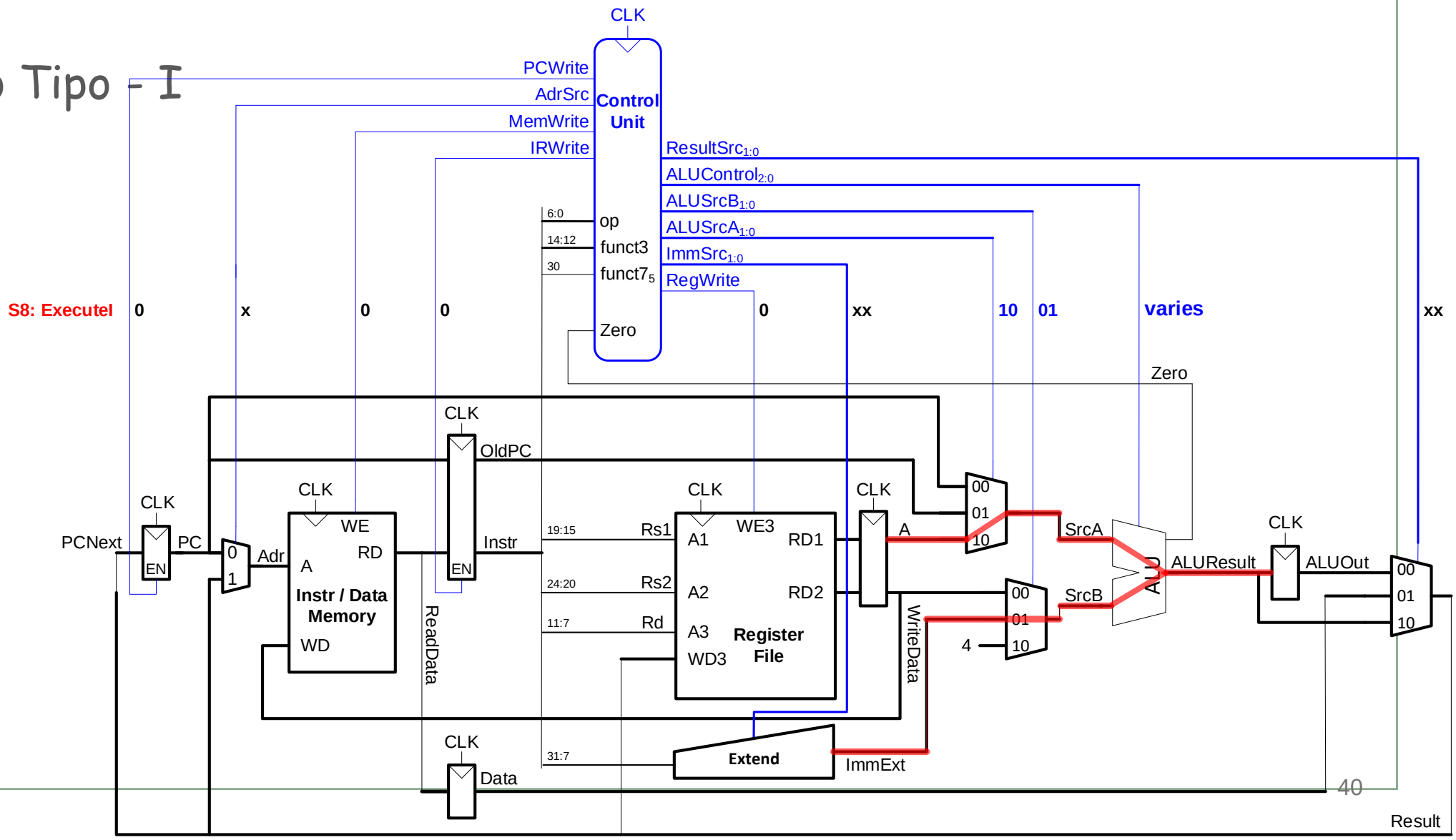
Datapath Multiciclo

- Instrução Tipo I



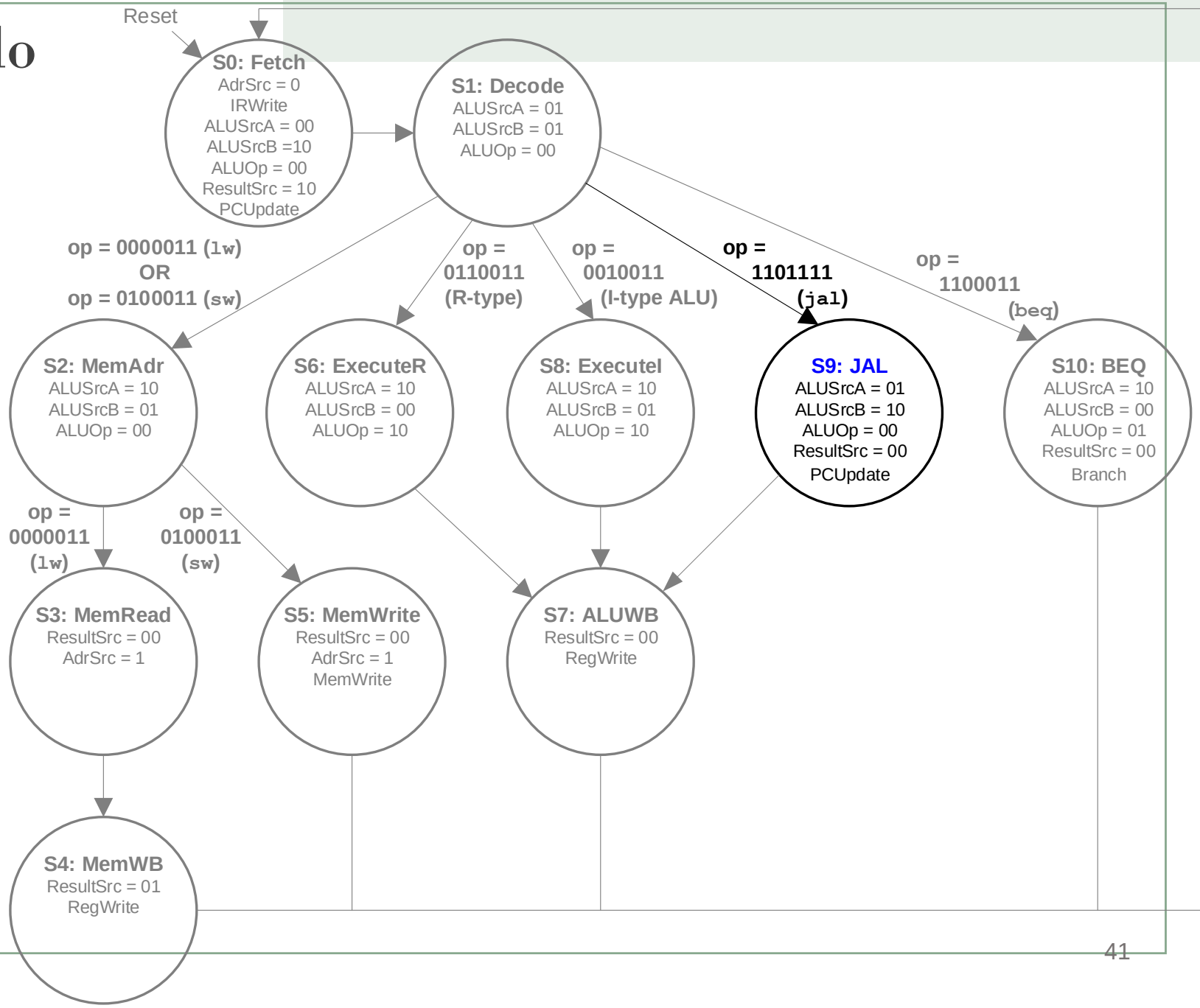
Datapath Multiciclo

- Instrução Tipo - I



Datapath Multiciclo

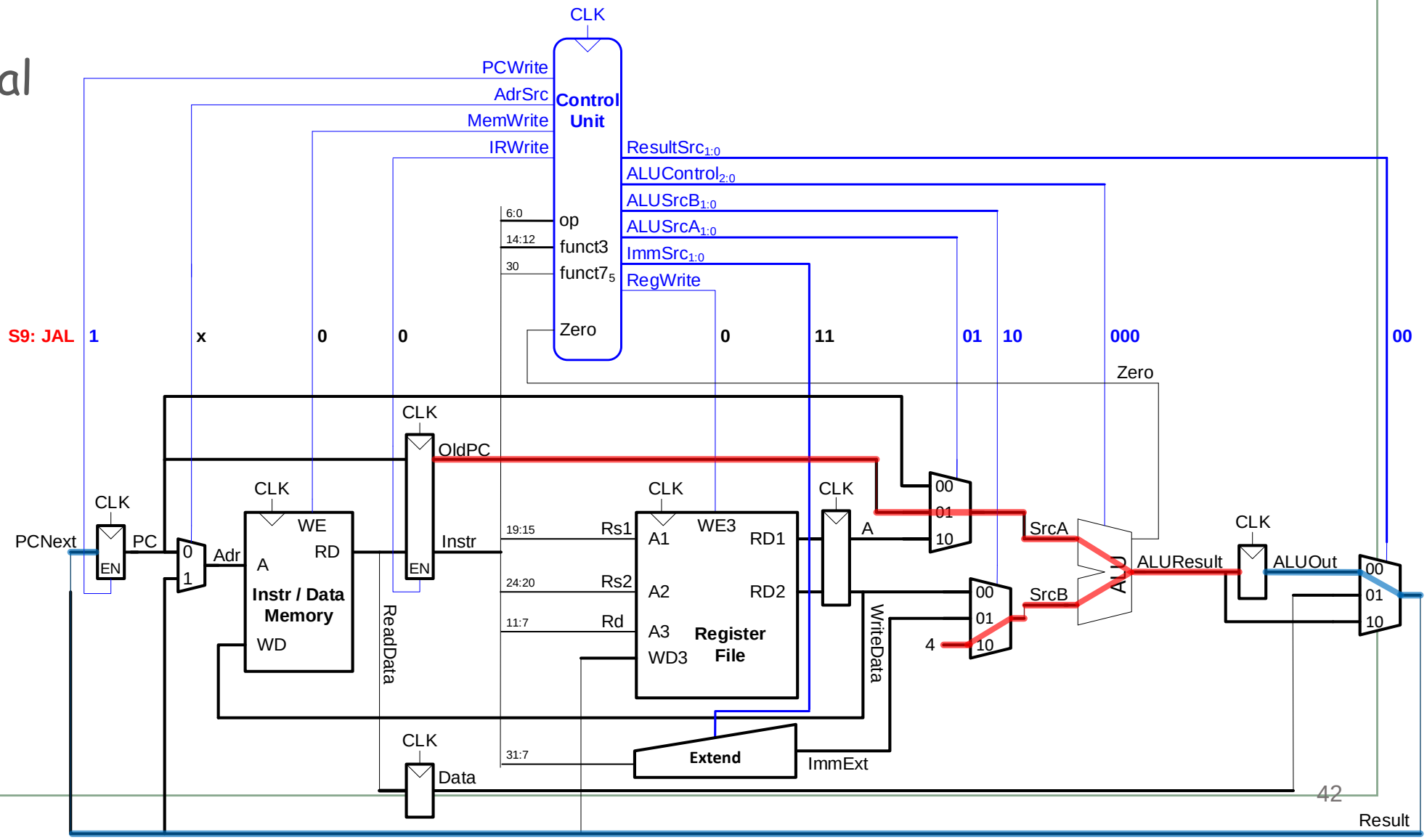
- Instrução Tipo jal



Calcula PC + 4 e
Envia endereço alvo (ALUOut) para PCNext

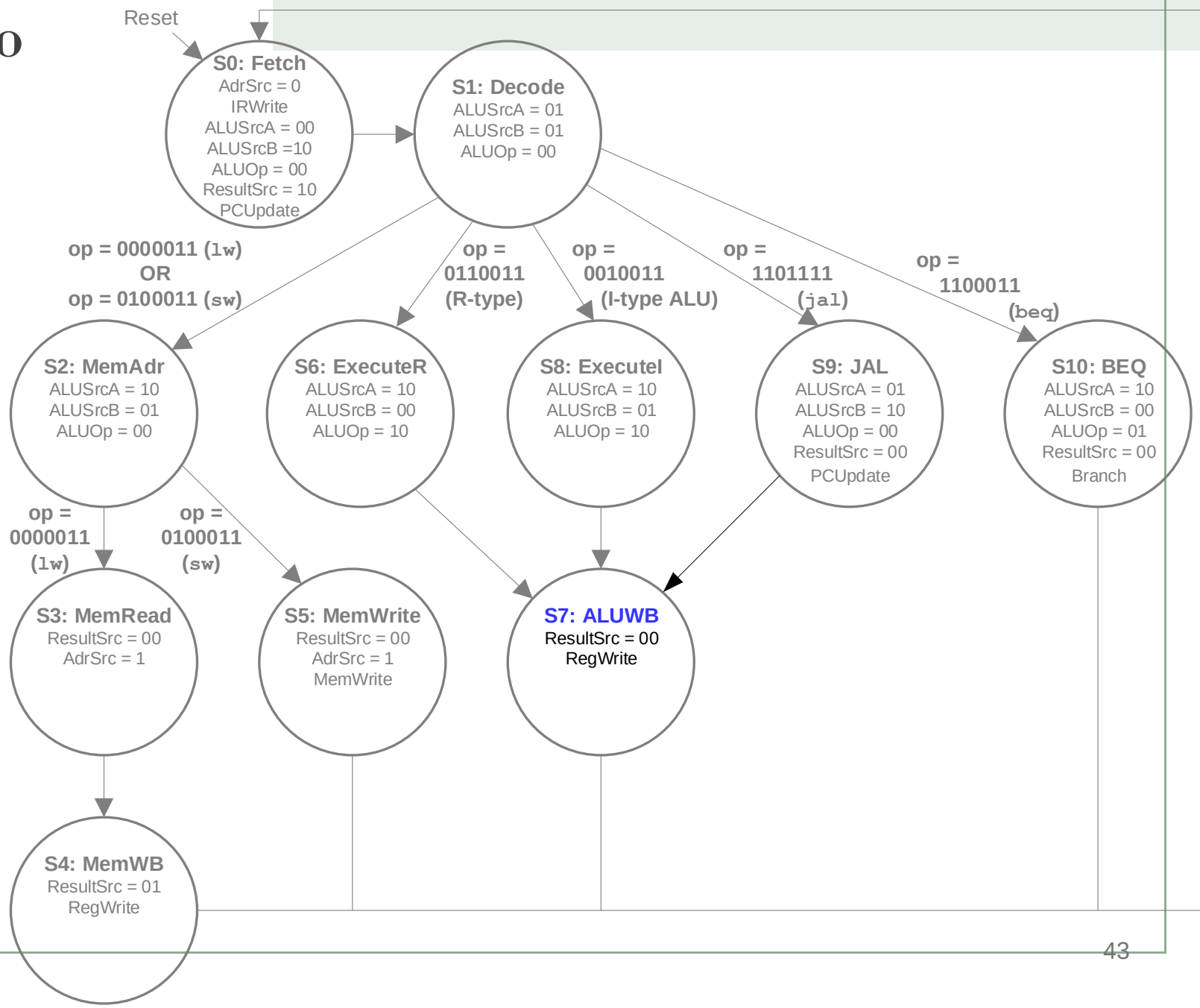
Datapath Multiciclo

- Instrução jal



Datapath Multiciclo

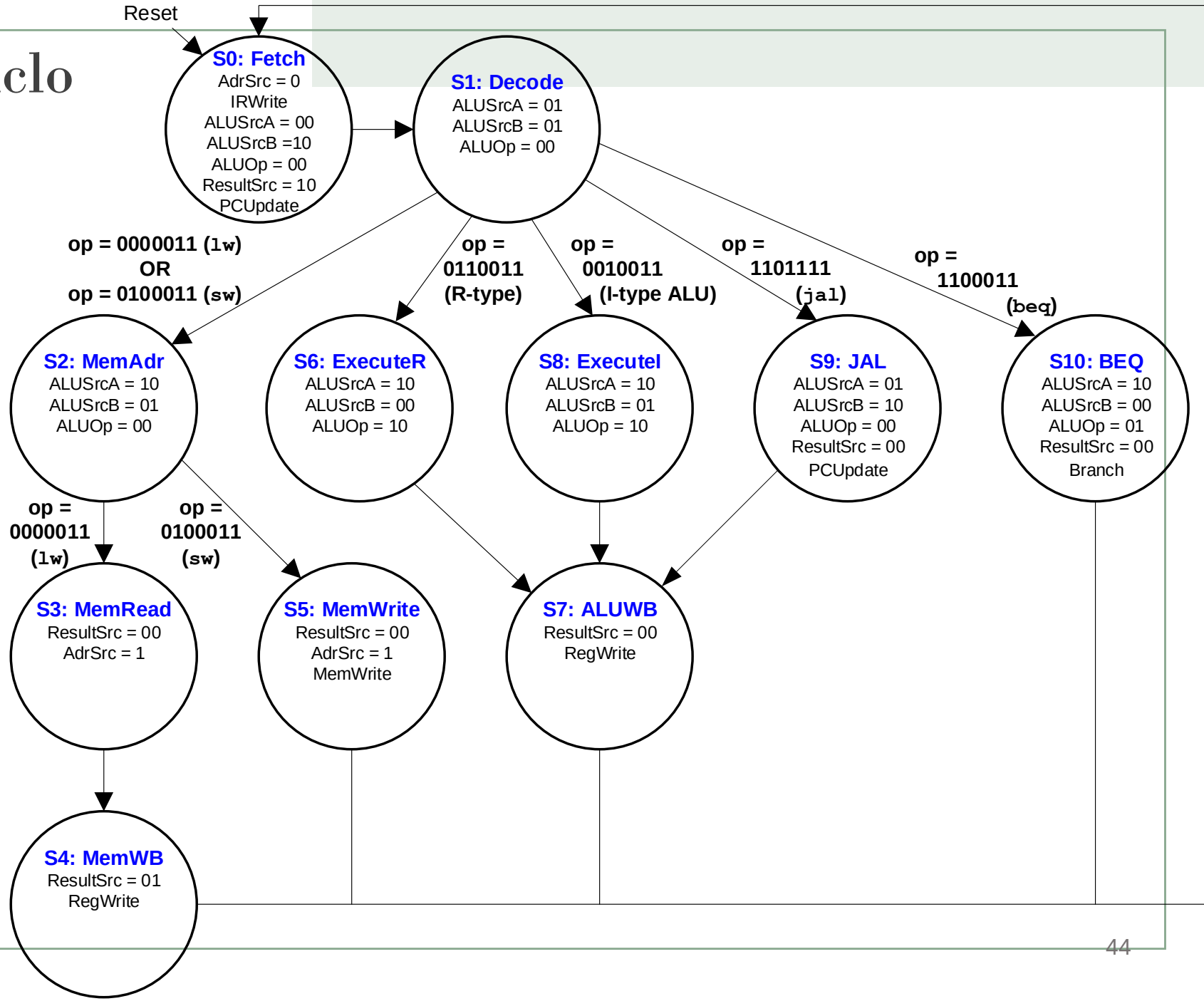
- Instrução Tipo jal



PC + 4 é escrito no **rd** em **S7: ALUWB**

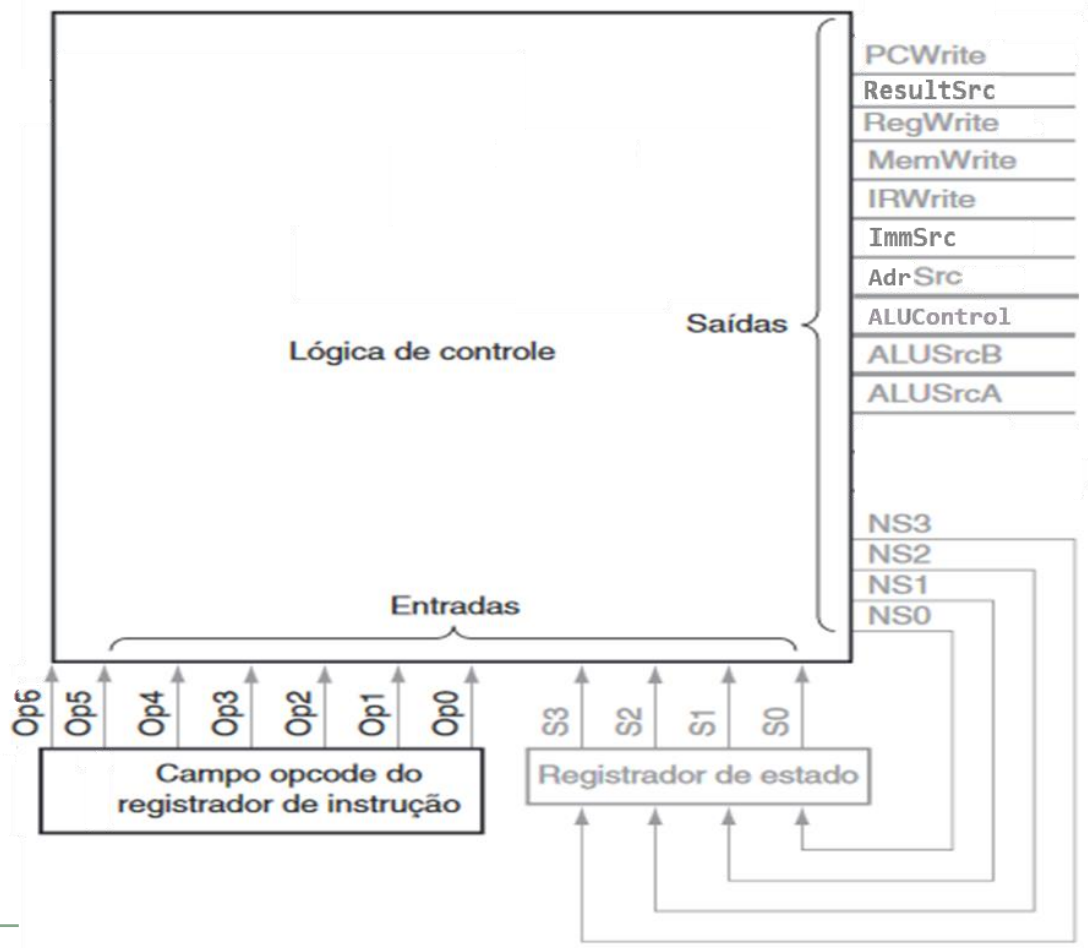
Datapath Multiciclo

State	Datapath μOp
Fetch	Instr ←Mem[PC]; PC ← PC+4
Decode	ALUOut ← PCTarget
MemAdr	ALUOut ← rs1 + imm
MemRead	Data ← Mem[ALUOut]
MemWB	rd ← Data
MemWrite	Mem[ALUOut] ← rd
ExecuteR	ALUOut ← rs1 op rs2
ExecuteI	ALUOut ← rs1 op imm
ALUWB	rd ← ALUOut
BEQ	ALUResult = rs1-rs2; if Zero, PC ← ALUOut
JAL	PC ← ALUOut; ALUOut ← PC+4



Datapath Multiciclo

- Máquina de Estados Moore



Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual	Opcode	Próximo Estado
0	-	1
1	‘lw’ ou ‘sw’	2
1	tipo R	6
1	‘beq’	10
1	‘jal’	9
2	‘lw’	3
2	‘sw’	5
3	-	4
4	-	0
5	-	0
6	-	7
7	-	0
9	-	7
10	-	0

Datapath Multiciclo

- Tabela de Atribuição de Estados

Estado	Flip-Flop			
	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual	Opcode	Próximo Estado
0000	-	0001
0001	lw ou sw	0010
0001	Tipo R	0110
0001	beq	1010
0001	jal	1001
0010	lw ou sw	0011
0010	sw	0101
0011	-	0100
0100	-	0000
0101	-	0000
0110	-	0111
0111	-	0000
1001	-	0111
1010	-	0000

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual				Opcode (Entrada)							Próximo Estado			
S3	S2	S1	S0	OP6	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0
0	0	0	0								0	0	0	1
0	0	0	1	0	0	0	0	0	1	1	0	0	1	0
0	0	0	1	0	1	0	0	0	1	1	0	0	1	0
0	0	0	1	0	1	1	0	0	1	1	0	1	1	0
0	0	0	1	1	1	0	0	0	1	1	1	0	1	0
0	0	0	1	1	1	0	1	1	1	1	1	0	0	1
0	0	1	0	0	0	0	0	0	1	1	0	0	1	1
0	0	1	0	0	1	0	0	0	1	1	0	1	0	1
0	0	1	1								0	1	0	0
0	1	0	0								0	0	0	0
0	1	0	1								0	0	0	0
0	1	1	0								0	1	1	1
0	1	1	1								0	0	0	0
1	0	0	1								0	1	1	1
1	0	1	0								0	0	0	0

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual				Opcode (Entrada)							Próximo Estado			
S3	S2	S1	S0	OP6	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0
0	0	0	0								0	0	0	1
0	0	0	1	0	0	0	0	0	1	1	0	0	1	0
0	0	0	1	0	1	0	0	0	1	1	0	0	1	0
0	0	0	1	0	1	1	0	0	1	1	0	1	1	0
0	0	0	1	1	1	0	0	0	1	1	1	0	1	0
0	0	0	1	1	1	0	1	1	1	1	1	0	0	1
0	0	1	0	0	0	0	0	0	1	1	0	0	1	1
0	0	1	0	0	1	0	0	0	1	1	0	1	0	1
0	0	1	1								0	1	0	0
0	1	0	0								0	0	0	0
0	1	0	1								0	0	0	0
0	1	1	0								0	1	1	1
0	1	1	1								0	0	0	0
1	0	0	1								0	1	1	1
1	0	1	0								0	0	0	0

NS3 = S3' . S2' . S1' . S0 . OP6 . OP5 . OP4' . OP3' . OP2' . OP1 . OP0 +
S3' . S2' . S1' . S0 . OP6 . OP5 . OP4' . OP3 . OP2 . OP1 . OP0

Datapath Multiciclo

- Sinais de controle (saída)

Saída	Estados Atuais
PCUpdate	S0 e S9
Branch	S10
ResultSrc1	S0
ResultSrc0	S4
RegWrite	S4 e S7
MemWrite	S5
IRWrite	S0
AdrSrc	S3 e S5
ALUOp1	S6 e S8
ALUOp0	S10
ALUSrcB1	S0 e S9
ALUSrcB0	S1 e S2
ALUSrcA1	S2, S6 e S10
ALUSrcA0	S1 e S9

Datapath Multiciclo

- Sinais de controle (saída)

Estado Atual				Opcode (Entrada)							Próximo Estado				PCUpdate	Branch	ResultSrc1	ResultSrc0	RegWrite	MemWrite	IRWrite	AdrSrc	ALUOp1	ALUOp0	ALUSrcB1	ALUSrcB0	ALUSrcA1	ALUSrcA0
S3	S2	S1	S0	OP6	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0														
0	0	0	0								0	0	0	1	1		1				1				1			
0	0	0	1	0	0	0	0	0	1	1	0	0	1	0												1		1
0	0	0	1	0	1	0	0	0	1	1	0	0	1	0												1		1
0	0	0	1	0	1	1	0	0	1	1	0	1	1	0												1		1
0	0	0	1	1	1	0	0	0	1	1	1	0	1	0												1		1
0	0	0	1	1	1	0	1	1	1	1	1	0	0	1												1		1
0	0	1	0	0	0	0	0	0	1	1	0	0	1	1												1	1	
0	0	1	0	0	1	0	0	0	1	1	0	1	0	1												1	1	
0	0	1	1								0	1	0	0								1						
0	1	0	0								0	0	0	0				1	1									
0	1	0	1								0	0	0	0						1		1						
0	1	1	0								0	1	1	1									1				1	
0	1	1	1								0	0	0	0					1									
1	0	0	1								0	1	1	1	1										1			1
1	0	1	0								0	0	0	0		1								1			1	

Datapath Multiciclo

- Sinais de controle (saída)

Estado Atual				Opcode (Entrada)							Próximo Estado				PCUpdate	Branch	ResultSrc1	ResultSrc0	RegWrite	MemWrite	IRWrite	AdrSrc	ALUOp1	ALUOp0	ALUSrcB1	ALUSrcB0	ALUSrcA1	ALUSrcA0
S3	S2	S1	S0	OP6	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0														
0	0	0	0								0	0	0	1	1		1				1				1			
0	0	0	1	0	0	0	0	0	1	1	0	0	1	0												1		1
0	0	0	1	0	1	0	0	0	1	1	0	0	1	0												1		1
0	0	0	1	0	1	1	0	0	1	1	0	1	1	0												1		1
0	0	0	1	1	1	0	0	0	1	1	1	0	1	0												1		1
0	0	0	1	1	1	0	1	1	1	1	1	0	0	1												1		1
0	0	1	0	0	0	0	0	0	1	1	0	0	1	1												1	1	
0	0	1	0	0	1	0	0	0	1	1	0	1	0	1												1	1	
0	0	1	1								0	1	0	0								1						
0	1	0	0								0	0	0	0				1	1									
0	1	0	1								0	0	0	0						1		1						
0	1	1	0								0	1	1	1									1				1	
0	1	1	1								0	0	0	0					1									
1	0	0	1								0	1	1	1	1										1			1
1	0	1	0								0	0	0	0		1								1			1	

PCUpdate = S3’·S2’·S1’·S0’ +

S3·S2’·S1·S0’

Datapath Multiciclo

- Acelera a execução de algumas instruções
  - Ex: jal não utiliza todos os estágios do datapath
  - Ex: Type-R não acessa a memória de dados

- Não acelara a execução das instruções mais complexas (ex: lw)

Datapath Multiciclo

- É possível melhorar ainda mais o desempenho?
- Sim:
  1. Processo de fabricação
  2. Exploração de Paralelismo
  3. Microarquitetura Avançada
     - Pipelining