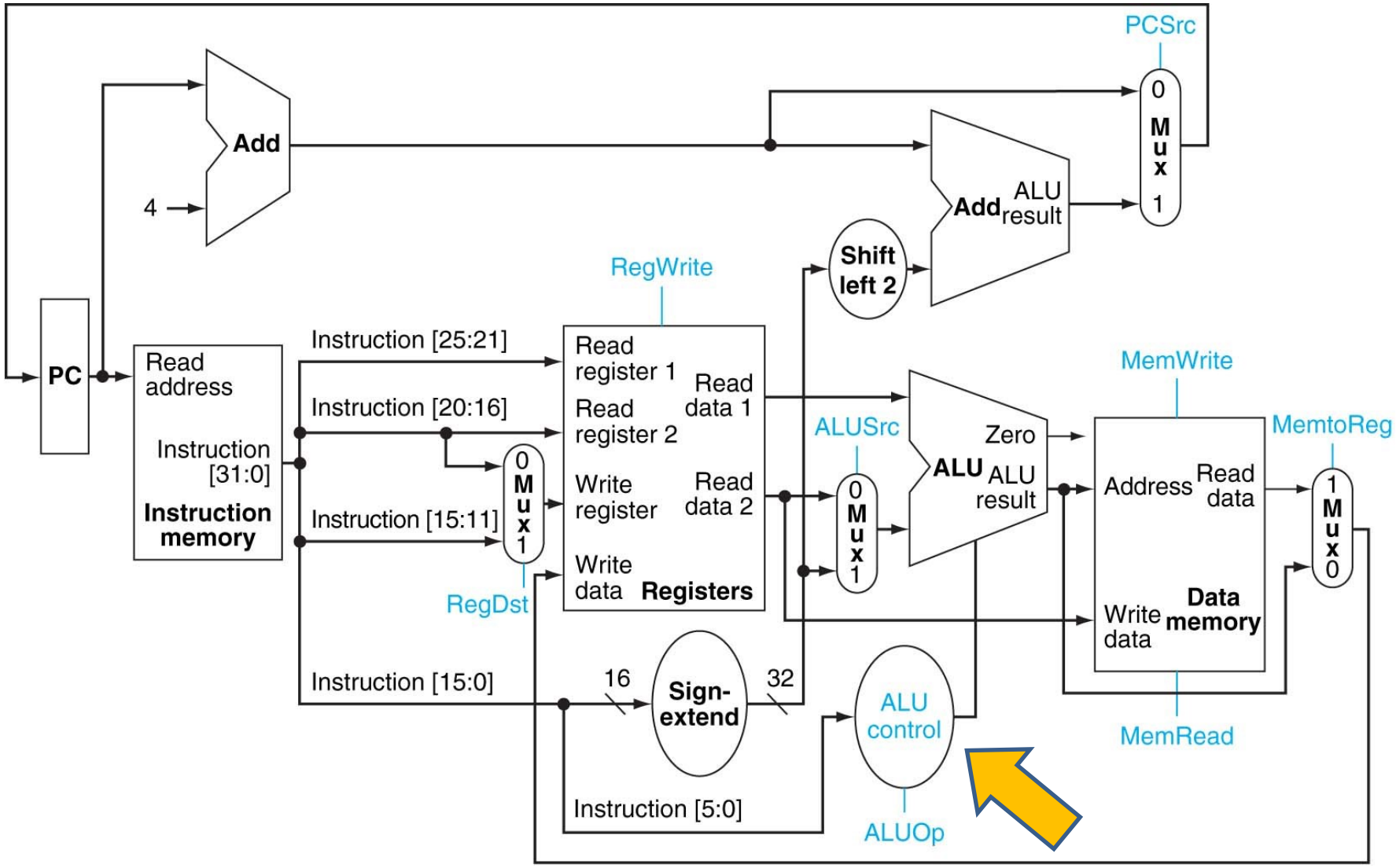
MIPS Básico


PCSrc
Add
4
0
M
u
x
1
Add
ALU result
Shift left 2
RegWrite
PC
Read address
Instruction [31:0]
Instruction memory
Instruction [25:21]
Instruction [20:16]
Instruction [15:11]
Instruction [15:0]
Instruction [5:0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
RegDst
ALUSrc
Zero
ALU
ALU result
MemWrite
MemtoReg
Address
Read data
Write data
Data memory
MemRead
16
Sign-extend
32
ALU control
ALUOp

Controle da ULA

INSTRUÇÃO	ALU CONTROL
AND – E lógico	0000
OR – Ou lógico	0001
ADD – Soma	0010
SUB – Subtração	0110
SLT – Set on Less Than (menor que)	0111

Unidade Execute

```
PROCESS ( ALU_ctl, Ainput, Binput )   // Gerar ALU_ctl
        BEGIN
        -- Seleciona a operação da ALU
        CASE ALU_ctl IS
                -- Operação E lógico
                WHEN "000“ => ALU_output_mux <= Ainput AND Binput;
                -- Operação OU lógico
                WHEN "001“ => ALU_output_mux <= Ainput OR Binput;
                -- Operação de Soma
                WHEN "010“ => ALU_output_mux <= Ainput + Binput;
                -- Operação de Subtração
                WHEN "110" => ALU_output_mux <= Ainput - Binput;
                -- Operação SLT
                WHEN "111“ => ALU_output_mux <= Ainput - Binput ;
                WHEN OTHERS => ALU_output_mux <= X"00000000" ;
        END CASE;
END PROCESS;
```

Ações da ULA por Instrução

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

Gerado pela Unidade de Controle

Bits úteis do campo *Function* de instruções R-format

Sinais de controle da ULA para seleção de operação

Unidade de Controle

```
Port ( …
        ALUop : OUT   STD_LOGIC_VECTOR( 1 DOWNTO 0 );
        …);
…
Architecture …
...
        ALUOp( 1 )      <=  R_format;
        ALUOp( 0 )      <=  Beq;
...
```

Ações da ULA por Instrução

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

Gerado pela Unidade de Controle

Bits úteis do campo *Function* de instruções R-format

Sinais de controle da ULA para seleção de operação

Resumo de ALU Control

ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
0	1	X	X	X	X	X	X	0110
1	0	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	0	X	X	0	1	0	0	0000
1	0	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Ajustar nomes corretos e entradas adicionais no código da unidade EXECUTE:

ALU_ctl(2) <= ALUOp0 OR (ALUOp1 AND F1);

Resumo de ALU Control

ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
0	1	X	X	X	X	X	X	0110
1	0	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	0	X	X	0	1	0	0	0000
1	0	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Ajustar nomes corretos e entradas adicionais no código da unidade EXECUTE:

ALU_ctl(2) <= ALUOp0 OR (ALUOp1 AND F1);
ALU_ctl(1) <= NOT (ALUOp1 AND F2);

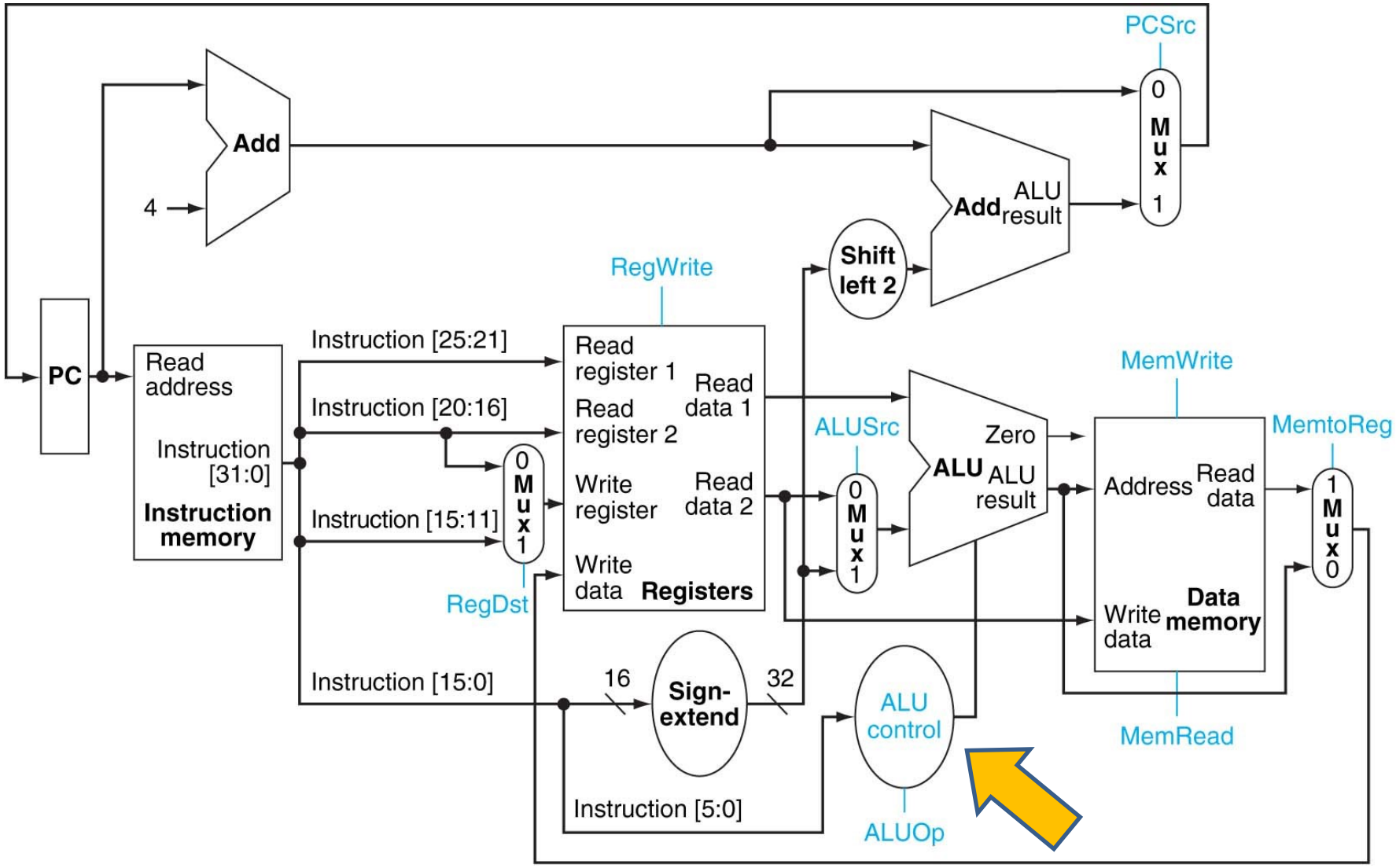
Resumo de ALU Control

ALUOp		Funct field						
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Operation
0	0	X	X	X	X	X	X	0010
0	1	X	X	X	X	X	X	0110
1	0	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	0	X	X	0	1	0	0	0000
1	0	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Ajustar nomes corretos e entradas adicionais no código da unidade EXECUTE:

```
ALU_ctl(2) <= ALUOp0 OR ( ALUOp1 AND F1);
ALU_ctl(1) <= NOT (ALUOp1  AND F2 );
ALU_ctl(0) <= ALUOp1 AND (F0 OR F3);
```

MIPS Básico


PCSrc
Add
4
0
M
u
x
1
Add
ALU result
Shift left 2
RegWrite
PC
Read address
Instruction [31:0]
Instruction memory
Instruction [25:21]
Instruction [20:16]
Instruction [15:11]
Instruction [15:0]
Instruction [5:0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
RegDst
ALUSrc
Zero
ALU
ALU result
MemWrite
MemtoReg
Address
Read data
Write data
Data memory
MemRead
16
Sign-extend
32
ALU control
ALUOp

Resumo de Alterações

- Editar a unidade EXECUTE:
  - Inserir entradas ALUOp e function;
  - Inserir lógica para ALU_ctl;
  - Editar ALU_Result para receber resultado da ULA;
  - Editar Zero para ser ativo sempre que o resultado for zero;
  - Inserir sinais internos adicionais;
  - Inserir o process de operações da ULA.
- Ajustar TLE para interligar novos sinais de controle.

Entrega

- Testar na próxima aula o experimento 5
- Entregar apenas os arquivos vhd modificados.