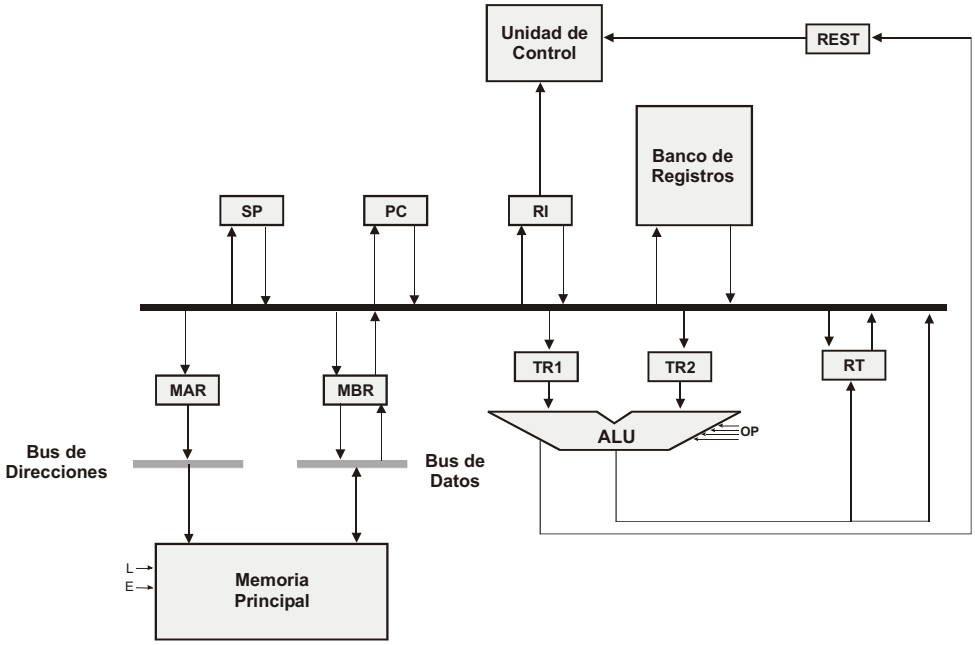
Processor design and control unit
Solved exercises

Exercise 1. The following figure shows the structure of a computer with a 32-bit CPU and a register file of 16 general registers (R1 to R16). MAR is the memory address register and MBR is the memory data register. The RT register is a temporary register that is transparent to the user. TR1 and TR2 are also transparent registers to the user. The ALU allows you to perform four operations: addition, subtraction, increment by one, and decrease by one. This computer represents integers in two's complement and numbers in floating-point according to the IEEE 754 standard of simple precision. Suppose that a cycle is required to access memory and that this is accessed at the word level and that decoding requires one cycle.



Answer the following questions, considering that instructions take up a word.

a) Microprogram at the elementary operation level the instruction "STORE Rk, /address".
   How many cycles are needed to execute this instruction on this computer?
   This instruction stores the contents of the Rk register in the memory position given by /address.
b) Microprogram the instruction "JMP (Rk)" at the elementary operations level.
   How many cycles do you need to execute this instruction on this computer?
   This instruction jumps to the memory address stored in Rk.

Solution:

a) Instruction: STORE Rk, /address

c1: MAR ← PC
c2: MBR ← Memory[MAR]
    TR1 ← PC
c3: PC ← TR1 + 1
c4 : R1 ← MBR
c5: decoding (jump to co2µaddr)
c6: MBR ← RK
c7: MAR ← RI(address)
c8: Memory[MAR] ←MBR

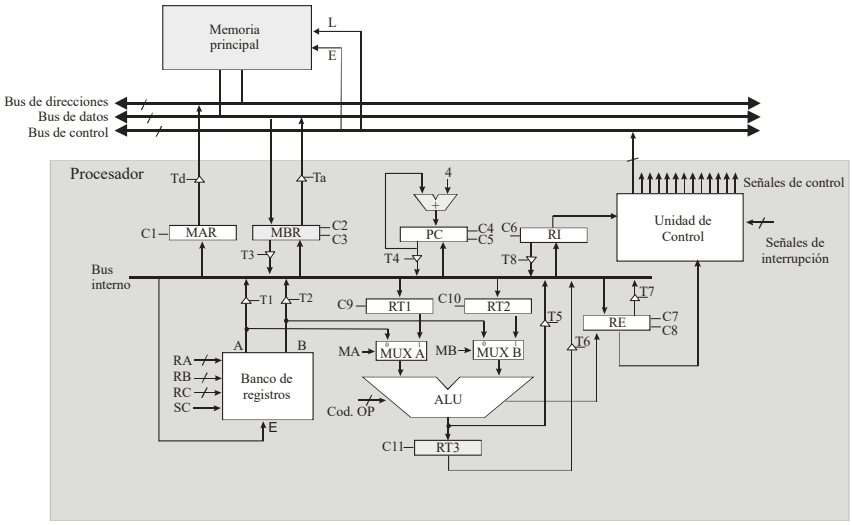
Note that in order to increase the program counter (PC = PC + 1) in the instruction fetch, it is necessary to take it prior to the transparent registry to the TR1 user (or TR2) and then increase it and store it on the PC. This requires the following two elementary operations:

TR1 ← PC
PC ← TR1 + 1

b) Instruction: JMP (Rk)

c1: MAR ← PC
c2: MBR ← Memory[MAR]
TR1 ← PC
c3: PC ← TR1 + 1
c4 : RI ←MBR
c5: decoding (jump to co2µaddr)
c6: PC ← RK

Exercise 2. Consider the following 32-bit processor schema. The register file includes 32 registers. Consider that the computer uses one clock cycle to perform the decoding of the instruction and that it connects to a memory that allows a read and a write operation to be performed in one cycle. The processor can execute 240 machine instructions.



It is requested:

a) What is the instruction register and what is it for? What is the status register and what is it for?
b) Consider the following hypothetical instruction: addr R1, R2, (R3), where R1, R2, and R3 represent registers and (R3) represents indirect register addressing. Indicate a possible format for the instruction.
c) Indicate the elementary operations and control signals required to execute the above instruction, bearing in mind that this instruction adds the contents of the R2 register to the contents of the memory position that is stored in the R3 register. The result is stored in R1.

Solution:

a) The instruction register stores the instruction to be executed. The status register stores information about the status of the program in progress.
b) As the computer executes 240 instructions, 8 bits are needed to represent the operation code (2^8 = 256). As the computer has 32 registers, 5 bits are needed to represent a register. The instruction occupies one word and its format is:

Cód. de operación	R1	R2	R3	
8 bits	5	5	5	9

c) Below are the elementary operations:

Cycle 0: MAR ← PC	Signals: T4, C3
Cycle 1: MBR ← Mem[MAR]	Signals: Td, L, C2
PC ← PC + 4	C4
Cycle 2: RI ← MBR	signals: T3, C6
Cycle 3: Decoding (jump to co2μaddr)	
Cycle 4: MAR ← R3	Signals: RA= identifier of R3, T1, C1
Cycle 5: MBR ← Mem[MAR]	Signals: Td, L, C2
Cycle 6: RT1 ← MBR	Signals: T3, C9
Cycle 7: R1← R2 + RT1	RB = id. of R2, MA = 1, MB = 0, Op.Cod. = sum, T5, Rc = id. of R1, SC, C8 (to update the RE)

Exercise 3. Consider a 32-bit processor similar to the one in exercise 2, that executes the RISC-V_{32} machine instructions. This computer uses a vectorized interrupt scheme. The interrupt vector table (IVT) consists of 256 elements and the IVT is stored from memory address 0.
It is requested:

a) Consider that a particular interrupt treatment routine is stored in the memory address $0x001F$ and that the interrupt vector corresponding to this interrupt is 3. Indicate the instruction or machine instructions necessary to store the address of the treatment routine in the entry corresponding to this interruption.
b) Consider the TRAP vector machine instruction, which triggers an operating system service. In this instruction, vector indicates the input of the interrupt vector table that includes the routine treatment direction. Consider that the stack pointer is the R1 register. Indicate the elementary operations required to execute this instruction.
c) Indicate the elementary operations required to execute the RETI machine instruction, which returns execution to the program previously suspended by the TRAP instruction.

Solution:

a) The necessary instructions are as follows:

```
la    $t0, 0x001F
sw    $t0, 12(x0)
```

b) Below are the necessary elementary operations:

C0: SP, MAR ← SP – 4
C1: MBR ← PC
C2: Mem[MAR] ← MBR
C3 SP, MAR ← SP – 4
C4: MBR ← RE
C5: Mem[MAR] ← MBR
C6: RT1 ← R0
C7: RT2 ← RI(vector)
C8: RT1 ← RT1 + 4
C9: MAR ← RT1 * RT2
C10: MBR ← Mem[MAR]
C11: PC ← MBR

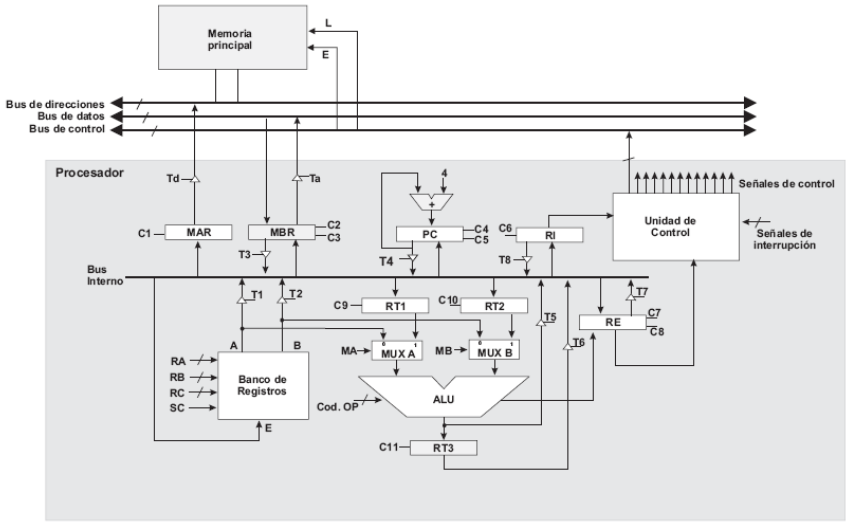
During these elementary operations, the contents of the program counter and state register are saved in the stack.

c) The elementary operations, excluding the *fetch* cycle necessary for the execution of the RTI statement are as follows:

C0: SP, MAR ← SP + 4
C1: MBR ← Mem[MAR]
C2: RE ← MBR
C3: SP, MAR ← SP + 4
C4: MBR ← Mem[MAR]

C5: PC ← MBR

Exercise 4. Consider a 32-bit processor similar to the one shown in the figure that executes the RISC-V_{32} machine instructions. The architecture has 250 different instructions, and the register file has 128 different registers. Memory reading and writing is done in a single clock cycle.



One instruction of this architecture is `TRIPLEADD`, which has the following syntax:

```
TRIPLEADD RD RO1 RO2 dir
```

This instruction adds three numbers by means of the following operation: RD ← (RO1 + RO2) + MP(dir)

This operation consists of two sums: first, it adds the contents of the `RO1` register with that of the `RO2` register. The result is in turn added with the contents of the `dir` memory position. The result of the sum is stored in the RD register. These three registers are generic from the register file while `dir` represents the memory address of a word.

It is requested:
a) Enter the format of the previous instruction.
b) Indicate the elementary operations required to execute the above instruction.

Solution:

a) The format of the instruction.

There are 250 different instructions, so the operation code will require 9 bits.
There are 126 registers, so register fields will require 7 bits.
The memory address occupies 32 bits.

The format, therefore, is as follows:

1st word (32-bit)

CO(9)	R(7)	R(7)	R(7)	Unused(2)

2nd word (32-bit)

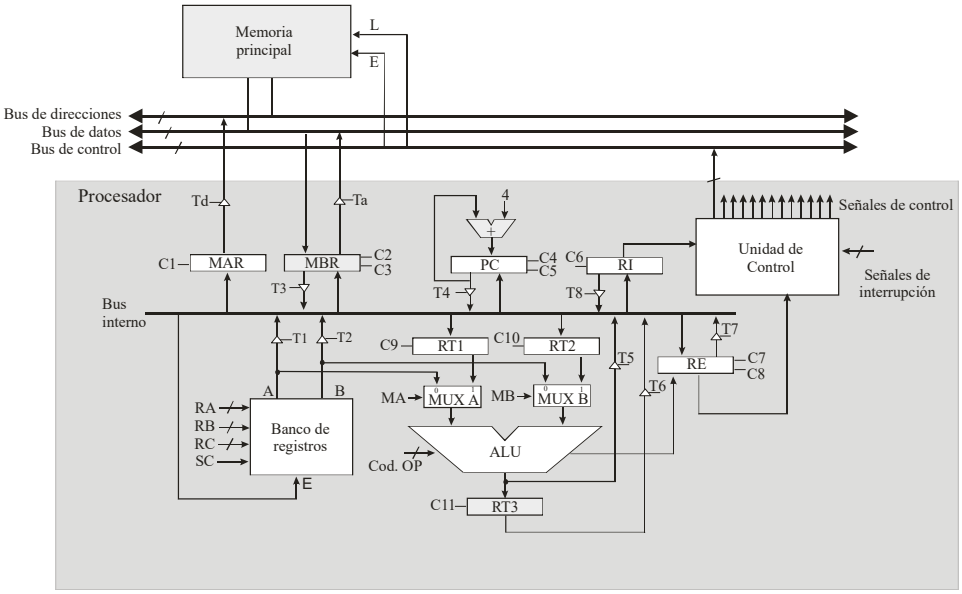
Dir(32)

b) Elementary operations and control signals

c1: MAR ← PC (T4, C1)
c2: PC ← PC + 4, MBR ← Mem[MAR] (Td, L, C4, C2)

c3: RI ← MBR (T3, C6)
c4: Decoding (jump to co2µaddr)
c5: MAR ← PC (T4, C1)
c6: PC ← PC + 4, MBR ← Mem[MAR] (Td, L, C4, C2)
c7: MAR ← MBR (T3, C1)
c8: MBR ← Mem[MAR] (Td, L, C2)
c10: RT1 ← MBR (T3, C9)
c11: RT2 ← R5 + R7 (RA=5,RB=7,MUX A=0, MUX B=0, Op.Cod.=ADD, T5,C10)
c12: R2 ← RT1 + RT2 (MUX A=1, MUX B=1, Op.Cod.=ADD, T5, RC=2, SC)

Exercise 5. Consider the following 32-bit processor schema. The register file includes 32 registers. Consider that the computer uses a single clock cycle to perform the decoding of the instruction and that it connects to a memory that allows a read and write operation to be performed in a single cycle.



a) What is and how is used the program counter register? What is the instruction register and what is used for? How many bits are needed for the control signals RA, RB, and C6?
b) Indicate in a reasoned way the necessary control signals that must be activated for the execution of the following elementary operation: MBR ← RE
c) Indicate the elementary operations and control signals required to execute the instruction `LOADM desp(R1), value` that copies the immediate value `value` to the memory position indicated by `desp(R1)`, bearing in mind that addressing relative to base register is used for this field. Consider that the instruction occupies one word.

Solution:

a) The program counter is the register that stores the address of the next instruction to be executed. The instruction register stores the instruction that is being executed at any given time. 5 bits are needed for RA and RB signals, since there are 32 registers, and 5 bits are needed to represent 32. For the C6 signal a single bit is needed

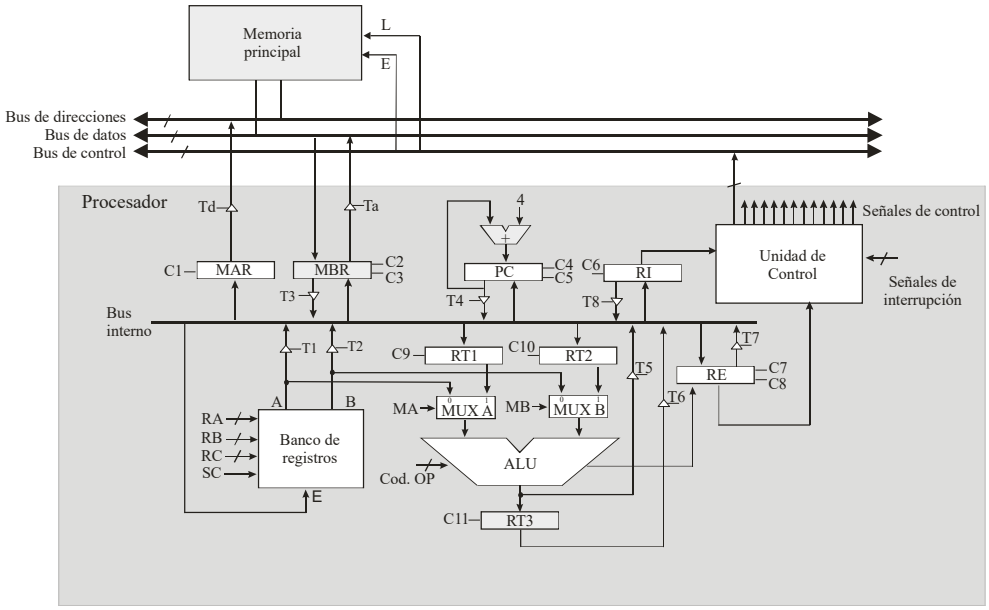
b) The T7 and C3 signals must be activated.

c) Below are the elementary operations and control signals.

Cycle	Elemental operation	Control signal
C1	MAR ← PC	T4, C1
C2	MBR ← Mem[MAR]	
	PC ← PC + 4	L, Td, C2, C4
C3	RI ← MBR	T3, C6

C4 Decoding (jump to co2μaddr)
C5 TR1 ← RI(desp) T8, C9
C6 MAR ← R1 + TR1 RB = <addr. of R1>, MA = 1, MB = 0, OpCod=ADD, T5, C1
C7 MBR ← RI(value) T8, C3
C8 MP[MAR] ← MBR Ta, E

Exercise 6. Consider the following 32-bit processor schema. The register file includes 32 records. Consider that the computer uses a clock cycle to perform the decoding of the instruction and that it connects to a memory that allows a read and write operation to be performed in a cycle.



This computer has the RISC-V$_{32}$ instruction set. Indicate the elementary operations and control signals (including fetch) required to execute the following instructions:

a) `addi t0, t1, 10`
b) `lw t1, 0(t2)`
c) `sw t1, 120(t2)`

Solution:

a) The following table shows the elemental operations and the control signals for the addi.

Cycle	Elementary operation	Activated control signals
C1	MAR ← PC	T4, C1
C2	MBR ← Mem[MAR], PC ← PC + 4	L, Td, C2, C4
C3	RI ← MBR	T3, C6
S4	Decoding	
S5	TR2 ← RI(10)	T8, C10
C6	$t0 ← $t1 + TR2	RA=<addr. of $t1> MA=0, MB=1, OpCod = ADD, T5, RC=<addr. of $t0>, SC

b) The following table shows the elementary operations and control signals for the lw instruction.

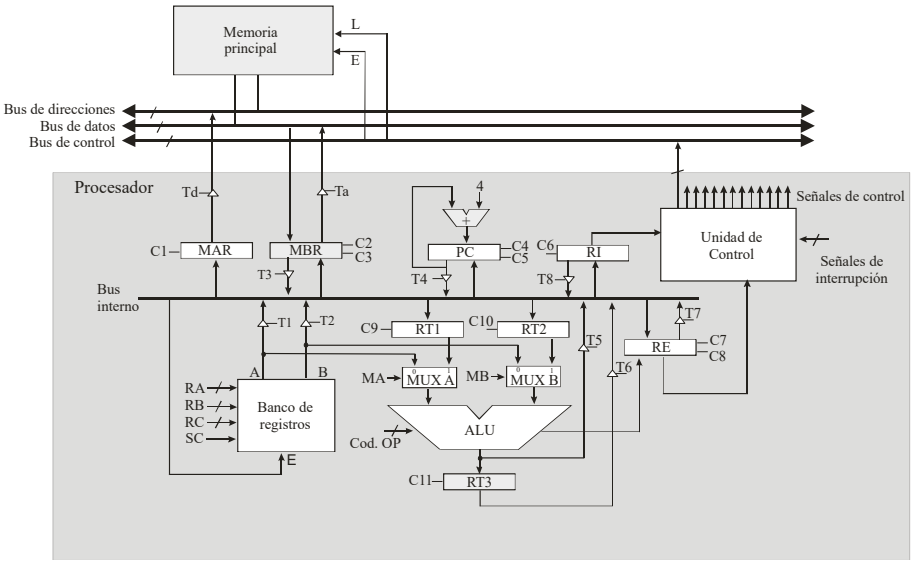
Cycle	Elementary operation	Activated control signals

C1	MAR ← PC	T4, C1
C2	MBR ← Mem[MAR], PC ← PC + 4	L, Td, C2, C4
C3	RI ← MBR	T3, C6
C4	Decoding	
C5	MAR ← $t2,	RA=<addr. of $t2> T1, C1
C6	MBR ← MP	Td, L, C2
C7	$t1 ← MBR	T3, SC, RA = <addr. of $t1>

c) Below are the elementary operations and signals for the sw instruction.

Cycle	Elementary operation	Activated control signals
C1	MAR ← PC	T4, C1
C2	MBR ← M[MAR]	Td, L, C2
C3	RI ←MBR, PC <- PC + 4	T3, C6, C4
C4	Decoding	
C5	RT2 ← RI(120)	T8, C10
C6	RT1 ← $t2	RA=$t2, T1, C9
C7	MAR ←RT1 + RT2	MA=1, MB=1, Op.Cod.=+, T5, C1
C8	MBR ← $t1	RA=$t1, C3
C9	MP[MAR] ← MBR	Ta, Td, E
C10	Jump to fetch	

Exercise 7. Consider the following 32-bit processor schema. The register file includes 32 registers. Consider that the computer uses one clock cycle to perform the decoding of the instruction and that it connects to a memory that allows a read and a write operation to be performed in one cycle.



This computer has the RISC-V_{32} instruction set. It is requested:

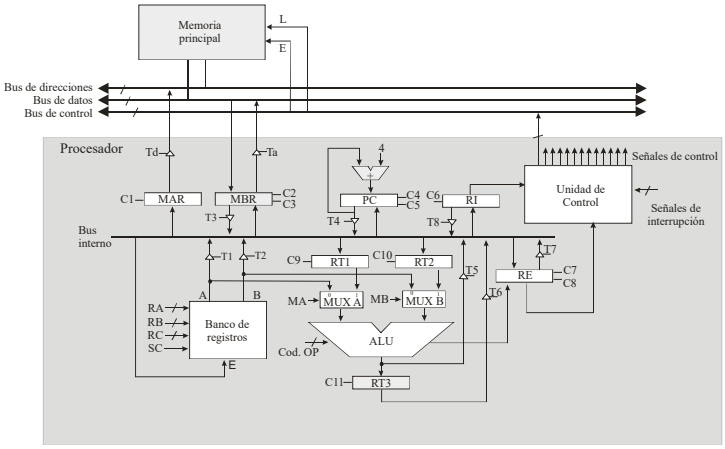
a) Indicate the control signals necessary to be able to perform the elementary operation PC ← R7, R7 being R7 the register from the register file with number 7.
b) If the T3 and C10 signals are activated during a clock cycle, indicate which elementary operation is being performed.

c) Indicate the elementary operations and control signals (including *fetch)* required to execute the instruction `addi t0, t1, 10`.

Solution:

a) Signals C5, T1 and RA = 00111 (register 7) must be activated
b) This is the operation: RT2 ← MBR
c) Cycle 0: MAR ← PC Signals: T4, C3
Cycle 1: MBR ← Mem[MAR] Signals: Td, L, C2
PC ← PC + 4 C4
Cycle 2: RI ← MBR Signals: T3, C6
Cycle 3: Decode (jump to co2µaddr)
Cycle 4: RT1 ← RI(10) Signals: T8, C9
Cycle 5: $t0 ← RT1 + $t1 Signals: MA=1, Op.Cod.=add, RB=<$t1>, T5, RC=<$t0>, SC, C8

Exercise 8. Consider the following 32-bit processor schema. The register file includes 32 registers. Consider that the computer uses a clock cycle to perform the decoding of the instruction and that it connects to a memory that allows a read and a write operation to be performed in one cycle.



This computer has the RISC-V_{32} instruction set.
It is requested:

a) Indicate the control signals necessary to be able to perform the elementary operation RT1 ← t2, being t2 the register number 2 of the register file.
b) Indicate the elementary operations and control signals (including *fetch)* required to execute the hypothetical instruction `addm t1, (t2)`. This instruction adds the contents of the register `$t1` with the contents of the memory position, which is stored in `t2`. The result is stored in the `t1` register. That is: `t1 ← t1 + MP[t2]`

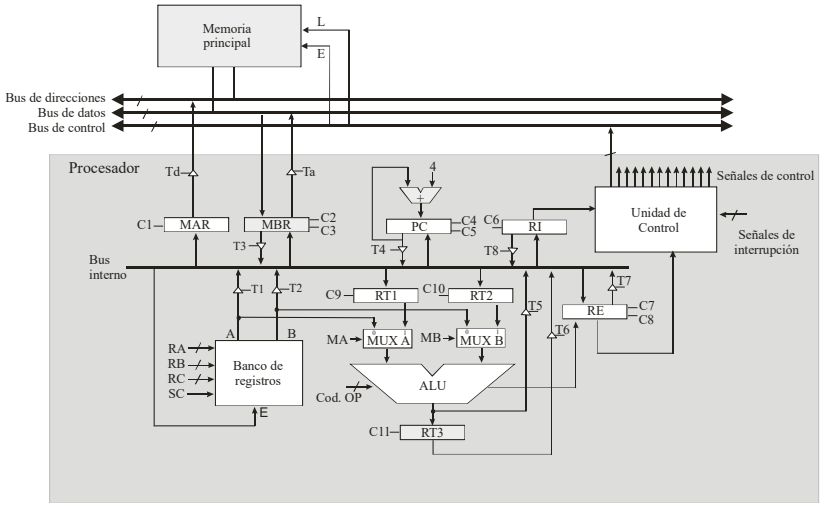
Solution:

a) Activate the signals C9, T1, RA = 00010
b)

Cycle	Op. Elementary	Control signals
Cycle 0:	MAR ← PC	T4, C3
Cycle 1:	MBR ← Mem[MAR]	Td, L, C2
	PC ← PC+4	C4
Cycle 2:	RI ← MBR	T3, C6
Cycle 3:	Decoding	
Cycle 4:	MAR ← t2	RA= <identifier of t2>, T1, C1
Cycle 5:	MP ← MBR	L
Cycle 6:	RT1 ← MBR	T3, C9
Cycle 7:	t1 ← RT1 + t1	MA = 1, RB =<id. of t1 register>, Op.Cod.=sum, T5, C8,

		RC =<id. of t1>, SC registry

Exercise 9. Consider a 32-bit computer, which has a register file of 32 registers, which is connected to a memory, which is addressed at bytes level and requires a cycle for a read and a write operation. The computer has an instruction set with 130 machine instructions. The arithmetic-logical unit can perform arithmetic and logical operations (adding, subtracting, multiplying, dividing, increasing, decreasing by one, etc.). The processor employs a cycle for decoding the instruction. Its structure is as follows:



Consider the following two instructions:

`moveM dir1, dir2` This instruction moves the contents of the `dir2` memory position to the `dir1` memory position.

`moveR R1, R2` This instruction moves the contents of the `R2` register to the `R1` register (`R1` and `R2` are the register 1 and 2 of the register file).

It is requested:

a) Indicate, in a reasoned manner, the format of the above instructions.
b) Indicate, depending on the format defined in paragraph a, the elementary operations and control signals necessary for the execution of the above instructions. Also include the fetch cycle.

Solution:

a) Because the computer has 130 instructions, it takes 8 bits to encode an instruction. The computer is 32-bit, so it takes 32 bits to encode a memory address. This instruction, therefore, needs three words for its encoding. Its content is as follows:

CO (8-bit)		**Dir 1 (32-bit)**	**Dir 2 (32-bit)**

The format of the second instruction would be:

CO (8-bit)	**R1 (5-bit)**	**R2 (5-bit)**	

a) First instruction:

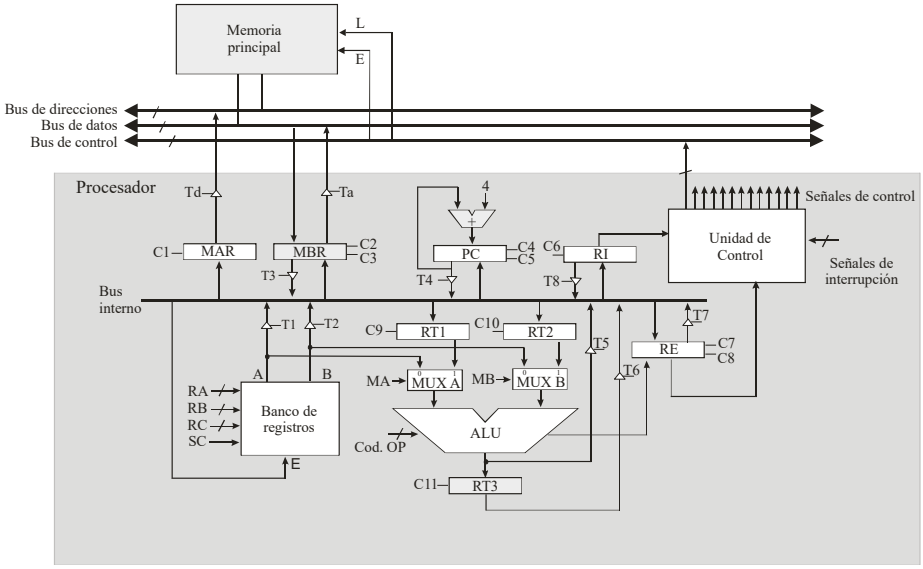
Cycle	**Elementary operations**	**Control signals**
C0	MAR ← PC	T4, C1

C1	MBR ← M[MAR], PC ← PC + 4	Td, L, C2, C4
C2	RI ← MBR	T3, C6
C3	Decoding	
C4	MAR ← PC	T4, C1
C5	MBR ← M[MAR], PC ← PC +4	Td, L, C2, C4
C6	RT1 ← MBR, RT2 ← MBR	T3, C9, C10
C7	MAR ← PC,	T4, C1
C8	MBR ← M[MAR], PC ← PC+ 4	Td, L, C2, C4
C9	MAR ← MBR	T3, C1
C10	MBR ← M[MAR]	Td, L, C2
C11	MAR ← RT1 OR RT2	MA, MB, T5, C1, Op.Cod. = OR
C12	M[MAR] ← MBR	Td, Ta, E

Second instruction:

Cycle	Elementary operations	Control signals
C0	MAR ← PC	T4, C1
C1	MBR ← M[MAR], PC ← PC + 4	Td, L, C2, C4
C2	RI ← MBR	T3, C6
C3	Decoding	
C4’	R1 ← R2	RA = 00001, RC = 00010, T1, SC

Exercise10. Consider a **32-bit** computer, which has a register file of **32 registers,** which is connected to a memory, which is addressed **by bytes** and requires **one cycle for a read and a write operation.** The computer has an instruction set with **110 machine instructions.** The arithmetic-logical unit can perform arithmetic and logical operations (adding, subtracting, multiplying, dividing, increasing, decreasing by one, etc.). The processor employs a cycle for decoding the instruction. Its structure is as follows:



Consider the following two instructions:

PUTS R1, R2	this instruction inserts two words at the top of the stack first the contents of the r1 register and above the contents of the r2 register
ADDS	This instruction extracts from the stack the contents of the first two stacked words. Perform the sum of these two values and insert the result at the top of the stack
POP R1	this function extracts an element (word from the stack) and stores it in r1

Considering that the stack pointer register is the R30 register and that the logical arithmetic unit has, among others, an operation that allows adding 4 to the data that enters the ALU by the MUX A and another that allows subtracting 4 from the data that enters the ALU by the MUX A. It is requested:

a) Indicate, in a reasoned manner, the format of the above instructions.
b) Indicate, depending on the format defined in a), the elementary operations and control signals necessary for the execution of the instruction PUTS. Also include the *fetch* cycle.
c) Using the three instructions above, write a program fragment whose behavior is similar to that performed by the RISC-V_{32} instruction `add R1, R2, R3`.

Solution:

a) Because the computer has 110 instructions, it takes 7 bits to encode an instruction. The computer is 32-bit, so it takes 32 bits to encode a memory address. This instruction, therefore, needs three words for its encoding. Its content is as follows:
The format of the PUTS statement is:

CO (7-bit)	**R1 (5-bit)**	**R2 (5-bit)**	Unused (15-bit)

The format of the second instruction would be:

CO (7-bit)	Unused (25-bit)

The format of the third instruction is:

CO (7-bit)	**R1 (5-bit)**	Unused (20-bit)

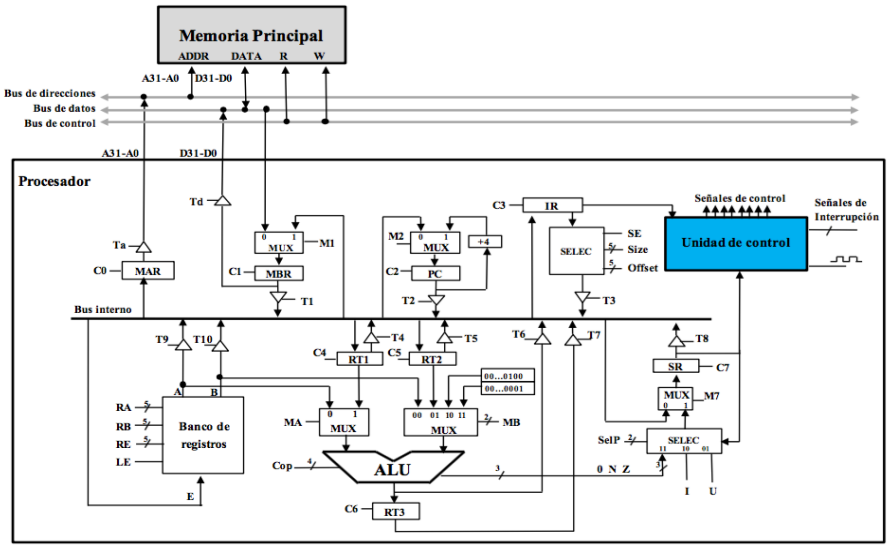
b) PUTS statement:

Cycle	**Elementary operations**	**Control signals**
C0	MAR ← PC	T4, C1
C1	MBR ← M[MAR], PC ← PC + 4	Td, L, C2, C4
C2	RI ← MBR	T3, C6
C3	Decoding	
C4	R30 ← R30 – 4	RA = 11110 (30 in binary) MA = 0 Cod. Op = subtract 4 T5, RC = 11110, SC
C5	MAR ← R30	RA = 11110 (30 in binary), T1, C1
C6	MBR ← R1	RA = 00001, T1, C3
C7	M[MAR] ← MBR	Td, Ta, E
C7	R30 ← R30 – 4	RA = 11110 (30 in binary) MA = 0 Cod. Op = subtract 4 T5, RC = 11110, SC
C8	MAR ← R30	RA = 11110 (30 in binary), T1, C1
C9	MBR ← R2	RA = 00010, T1, C3
C10	M[MAR] ← MBR	Td, Ta, E

c) The necessary fragment would be:

```
PUTS R1, R2
ADDS
POP R1
```

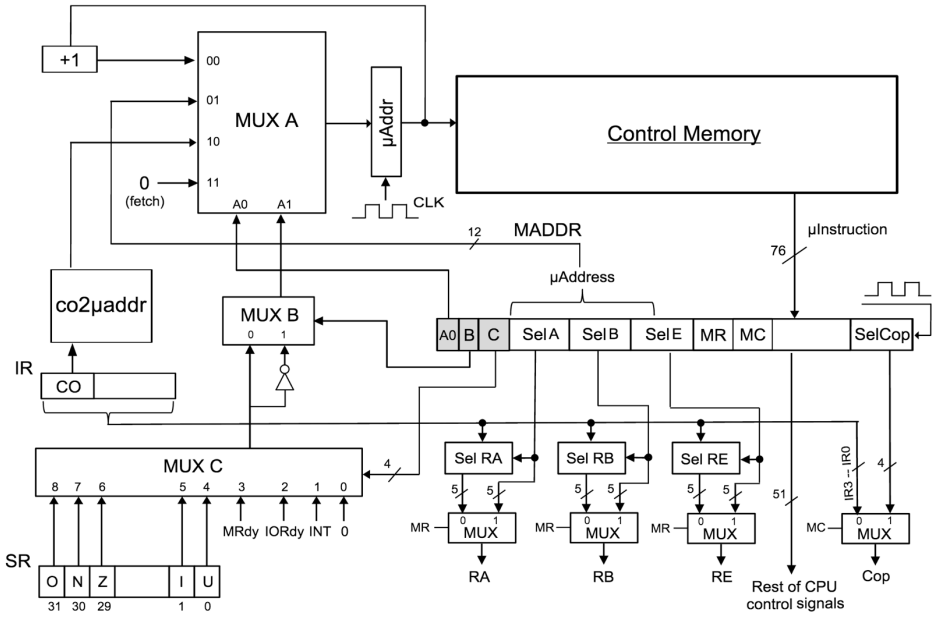
Exercise 11. Given the processor with the following structure:



And with the following characteristics:

- A register file of 32 32-bit registers, capable of processing 127 different instructions.
- An ALU with 16 arithmetic-logical operations that include addition, subtraction, multiplication, the and, or, not, xor at the bit level, the right and left rotation of a bit, and the logical and arithmetic displacements to the right, and the logical displacement to the left.
- Byte-level addressable memory and one single clock cycle for reading or writing.

This processor has a Control Unit represented by the following figure:



Briefly and justifiably responds to the following questions:

a) Analyzing the diagram of the Control Unit, what control technique do you employ? What type of sequencing do you use?

b) Given the two-word statement `li.w R value` that stores "value" in the register "R", indicate an instruction format for it so that the first word contains the register "R" and the second word is "value" as the 32-bit binary number.

c) Specify the elementary operations and all the necessary control signals, both from the control unit and the processor, to execute the machine instruction `li.w R value` described above. Include the *fetch* cycle.

NOTE: Assume that R29 acts as a stack pointer, that the stack pointer points to stack top, and that the stack grows toward decreasing memory directions.

Solution:

a) This Control Unit has a microprogrammed control technique and implicit sequencing.

b) A possible format for the instruction described would be:

First word:

Instruction code **(7-bit)**	Register **(5-bit)**	

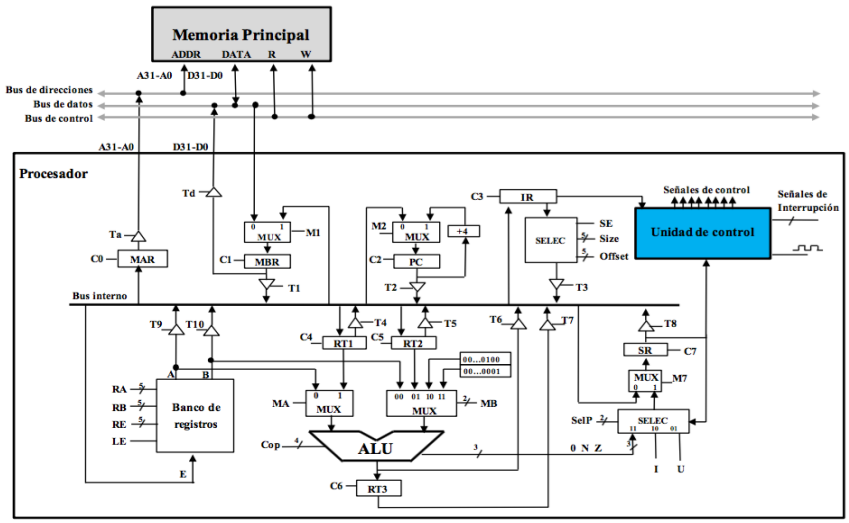
Second word:

Value **(32-bit)**

c) Elementary operations and control signals:

Elementary operations	**Control signals**
Fetch	
MAR ← PC	T2, C0
MBR ← Mem[MAR]	TA, R, BW=11, C1=1
PC ← PC+4, IR <- MBR	M2, C2, T1, C3
Jump to co2µaddr	A0, B=0, C=0
li.w R value	
MAR ← PC, PC <- PC+4	T2, C0, M2=1, C2
MBR ← Mem[MAR]	TA=1, R=1, BW=11, C1=1
R ← MBR, jump to fetch	T1=1, LE=1, MR=0, SELE=10100, A0=1, B=1, C=0

Exercise 12. Given the processor with the following structure:

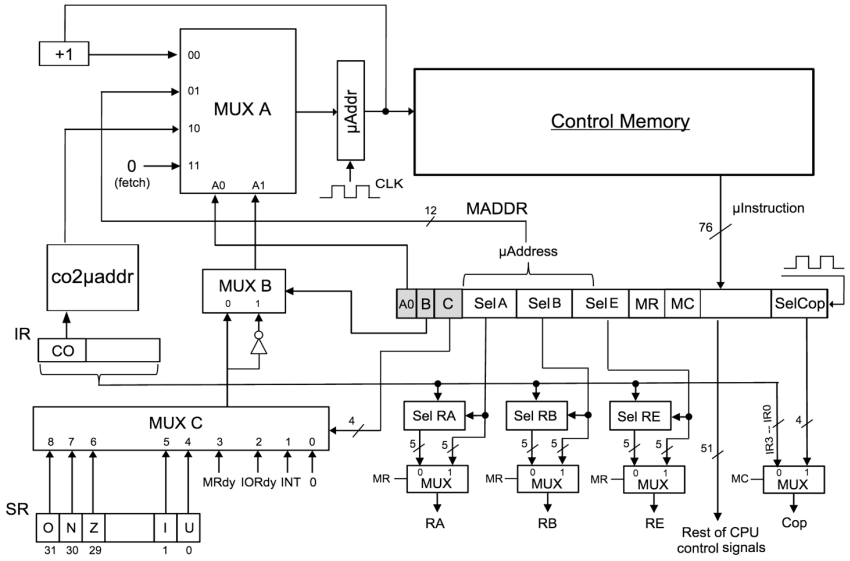


And with the following characteristics:

- A register file with 32 registers of 32 bits, capable of processing 63 different instructions.
- An ALU with 16 arithmetic-logical operations that include addition, subtraction, multiplication, the and, or, not, xor at the bit level, the right and left rotation of a bit, and the logical and arithmetic displacements to the right, and the logical displacement to the left.

- Byte-level addressable memory and one single clock cycle for reading or writing.

This processor has a Control Unit represented by the following figure:



Respond briefly and justifiably to the following questions:

a) When implementing a Control Unit, what control techniques is it possible to use? What type of sequencing is possible to use in a Control Unit?

b) Consider the instruction `add.w R1 R2 value` that stores in the register "R1" the sum of the contents of "R2" and "value". This statement adds numbers in complement to two. For the immediate value "value" the same format is used as the one stored in the R1 and R2 register. Please provide a valid format for that instruction.

c) Specify the elementary operations and control signals required to execute the `add.w R1 R2 value` machine instruction described above. Please include the *fetch* cycle.

Assume that R29 acts as a stack pointer, that the stack pointer points to stack top, and that the stack grows toward decreasing memory directions.

Solution:

a) The Control Unit can use **microprogrammed** logic or **wired** logic. Regarding sequencing, it can be **implicit or explicit.**

b) A possible format for the instruction described would be:

First word:

Instruction code **(6-bit)**	Register **(5-bit)**	Register **(5-bit)**	Not-used (**16-bit**)

Second word:

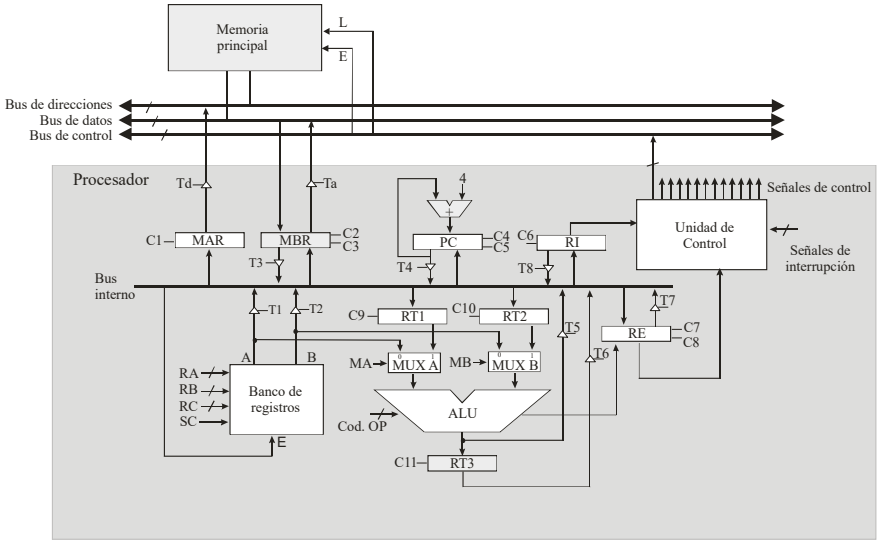
Value **(32-bit)**

c) Elementary operations and control signals:

Elementary operations	**Control signals**
Fetch	
MAR <- PC	T2, C0
MBR <- Mem[MAR]	TA, R, BW=11, C1=1
PC <- PC+4, IR <- MBR	M2, C2, T1, C3
Jump to co2µaddr	A0, B=0, C=0

li.w R1 R2 value	
MAR <- PC, PC <- PC+4	T2, C0, M2=1, C2
MBR <- Mem[MAR]	TA=1, R=1, BW=11, C1=1
RT1 <- MBR	T1, C4
RT2 <- IR(R2)	SE=0, Size=5, Offset=16, T3, C5
R1 <- RT1+RT2, jump to fetch	T1=1, LE=1, MR=0, SELE=10101, A0=1, B=1, C=0

Exercise 13. Consider a 32-bit computer, which has a register file of 32 registers, which is connected to a memory, which is addressed by bytes and requires a cycle for read and write operations. The computer has an instruction set with 120 machine instructions. The arithmetic-logical unit can perform arithmetic and logical operations (adding, subtracting, multiplying, dividing, increasing, decreasing by one, etc.). The processor employs a cycle for decoding the instruction. The R0 register has its wired value to 0, and the R31 register is used as the stack pointer register. Its structure is as follows:



Consider the following `swapSP R1` machine instruction. This instruction exchanges the content stored in the `R1` register with the content stored at the top of the stack.

It is requested:

a) Indicate, in a reasoned manner, the format of the previous statement.
b) Indicate, depending on the format defined in (a), the elementary operations and control signals necessary for the execution of the above instruction. Also include the *fetch cycle.*

Solution:

a) As the computer has 120 instructions, it takes 7 bits ($\log_2 120$ rounded up) to encode an instruction. As there are 32 registers, 5 bits are needed to identify each register ($2^5 = 32$). This instruction, therefore, needs a word for its encoding. Its content is as follows:

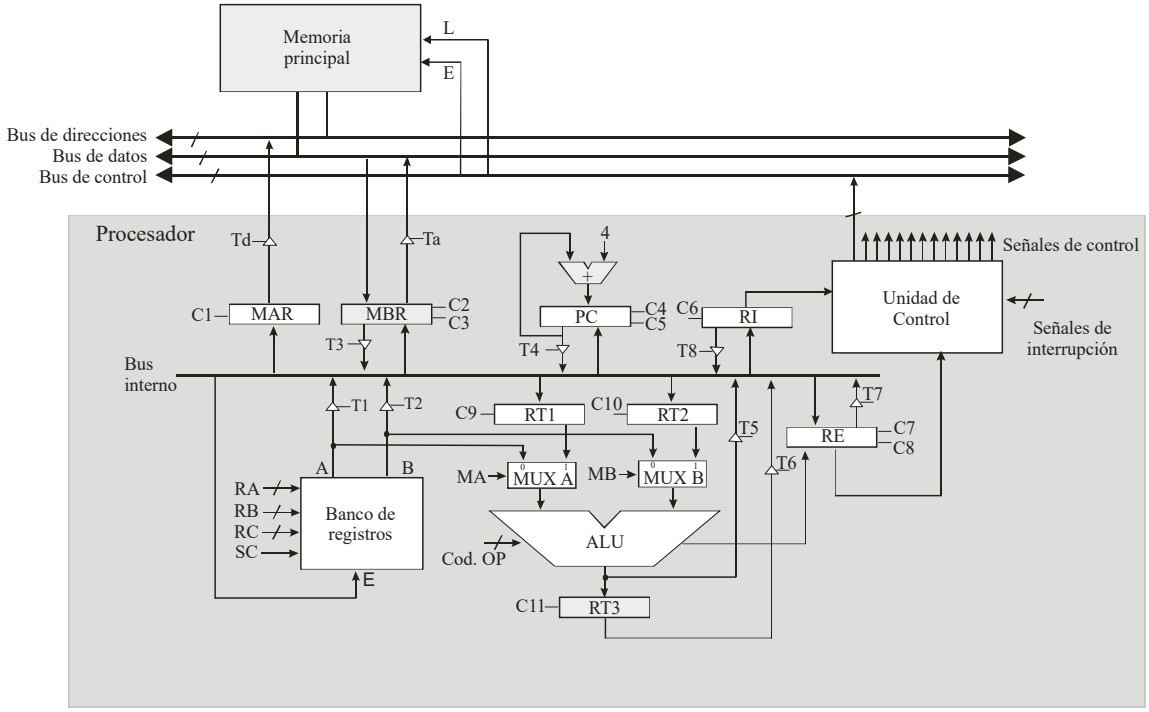
CO (7-bit)	**R1 (5-bit)**	

b) Below are the cycles, elementary operations, and control signals needed, keeping in mind that the stack pointer is the R31 register:

Cycle	**Elementary operations**	**Control signals**
C0	MAR ← PC	T4, C1
C1	MBR ← M[MAR], PC ← PC + 4	Td, L, C2, C4
C2	RI ← MBR	T3, C6

C3	Decode (Jump to co2µaddr)	
C4	MAR ← R31	RA=11110, T1, C1
C5	MBR ← M[MAR]	L, TD, TA, C2
C6	RT2 ← MBR	T3, C10
C7	MBR ← R1	RA=00001, T1, C3
C8	M[MAR] ← MBR R1 ← R0 + RT2	TD, TA, E Cod. OP=add, MB, RA=00000, T5, RC=00000, SC

Exercise 14. Consider a 32-bit computer, with a clock frequency of 2 GHz, that has a register file of 32 registers, connected to a memory addressed at byte level and that requires one cycle for a read and a write operation. The computer has an instruction set with 120 machine instructions. The arithmetic-logical unit can perform arithmetic and logical operations (adding, subtracting, multiplying, dividing, increasing, decreasing by one, etc.). The processor employs one cycle for decoding the instruction. The R0 register has its wired value to 0, and the R31 register is used as the stack pointer register. Its structure is as follows:



Consider the following machine instruction, whose operation code is `ADDSP R1, dir`. This instruction adds the content stored in the `R1` register with the content stored in the `dir` memory address and leaves the result at the top of the stack.

It is requested:

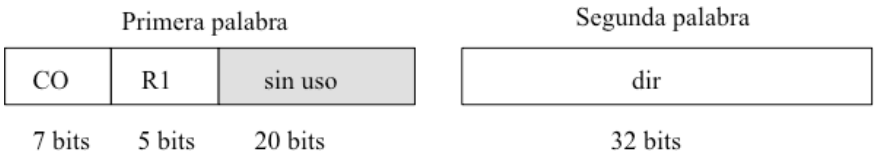
a) Indicate, in a reasoned manner, the format of the previous instruction.
b) Indicate, depending on the format defined in (a), the elementary operations and control signals necessary for the execution of the above instruction. Also include the *fetch cycle*.
c) Consider that 20% of the instructions on this computer need the same number of cycles for execution as the ADDSP instruction and that the remaining 80% need on average 7 cycles. Calculate the RISC-V_{32} value for this computer.

Solution:

a) To represent the ADDSP R1 instruction, the following bits are required:
   - 7 bits for the operation code, since with 7 bits you can identify up to 127 instructions.

- 5 bits for the register since with 5 bits you can identify the 32 registers.
- The memory addresses on that computer occupy 32 bits.

Therefore, to be able to fully represent this instruction, 5 + 7 + 32 bits are needed, that is, two words. The format will be as follows:

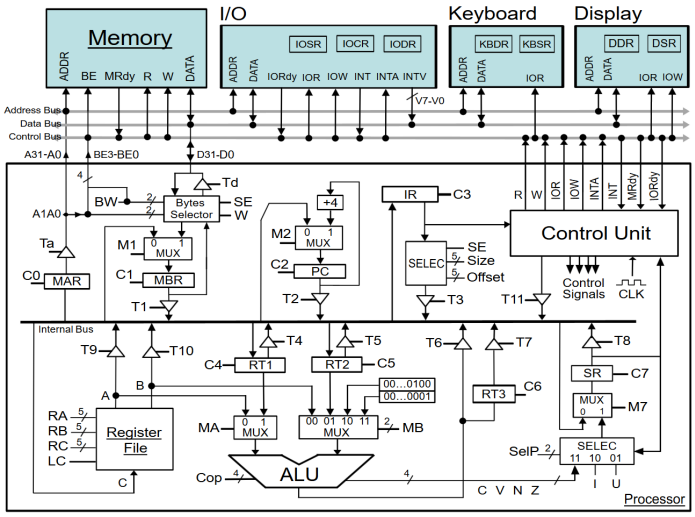


b) The elementary operations and signals would be:

PC →MAR	T4, C1 // fetch
Mem[MAR] → MBR, PC + 4 → PC	Td, L, C2, C4
MBR → RI	T3, C6
Decoding (Jump to co2µaddr)	
PC →MAR	T4, C1 // reading second word
Mem[MAR] → MBR, PC + 4 → PC	Td, L, C2, C4
MBR → MAR	T3, C1
Mem[MAR] → MBR	Td, L, C2
MBR →RT2	T3, C10
R1 + RT2 → MBR	RA=R1, MA=0, MB=1, Op.Cod.=+, T5, C3
R31 →MAR	RA=R31, T1, C1
MBR → Mem[MAR]	E, TA

c) For the execution of the above instruction 12 cycles are needed. Therefore, the average number of cycles per instruction on this computer = 0.8 * 7 + 0.2 * 12 = 8 cycles on average per instruction.
2 GHz assumes $210\times^{9}$ cycles per second, at 8 cycles per instruction on average could be executed $(210\times^{9})/8 = (200010\times^{6})/8 = 250$ MIPS.

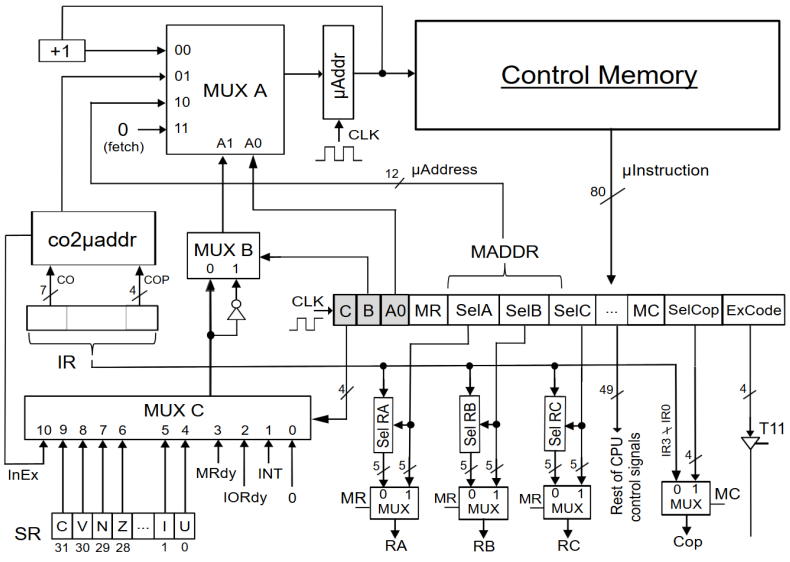
Exercise 15. Given the processor with the following structure:



And with the following characteristics:

- A register file with 32 32-bit registers, capable of processing 127 different instructions.
- An ALU with 16 arithmetic-logical operations that include addition, subtraction, multiplication, the and, or, not, xor at the bit level, the right and left rotation of a bit, and the logical and arithmetic displacements to the right, and the logical displacement to the left.
- Byte-level addressable memory and one single clock cycle for reading or writing.

This processor has a Control Unit represented by the following figure:



Respond briefly and justifiably to the following questions:

a) Analyzing the diagram of the Control Unit, what control technique do you employ? What type of sequencing do you use?

b) Consider the instruction `set.v R1, R2, address`. This instruction is a vector instruction that stores the value contained in the `R1` register in the N consecutive memory addresses from `address`. The value N is stored in the `R2` register. That is, it stores the value in `R1` in the memory positions:
`address, address +4, address +8 . . . . . address + Nx4.`
Properly specify the format of this instruction.

c) Specify the elementary operations and all the necessary control signals, both from the control unit and the processor, to execute the machine instruction described above. Include the *fetch* cycle.

Assume that R29 acts as a stack pointer, that the stack pointer points to stack top, and that the stack grows toward decreasing memory directions.

Solution:

a) This Control Unit has a **microprogrammed control** technique and **implicit sequencing.**

b) A possible format for the instruction described would be:

First word:

Instruction code **(7-bit)**	Register **(5-bit)**	Register **(5-bit)**	**(15-bit)**

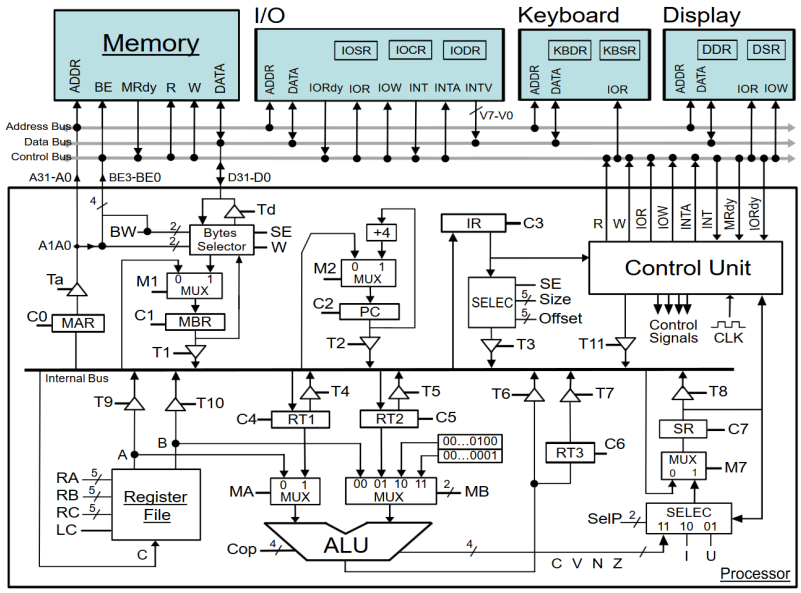
Second word:

Value **(32-bit)**

c) Elementary operations and control signals:

Cycle	Elementary operations	Control signals
Fetch		
C1	MAR ← PC	T2, C0
C2	MBR ← Mem[MAR]	TA, R, BW=11, C1, M1
C3	PC ← PC+4, IR ← MBR	M2, C2, T1, C3
S4	Decoding (Jump to co2µaddr)	A0, B=0, C=0
Execution of the instruction		
C5	MAR ← PC	T2, C0
C6	MBR ← MP[MAR], PC ← PC + 4	TA, R, BW=11, C1, M2, C2, M1
C7	RT1 ← **R2**	C4, T9, SelA=R2_offset, MR=0
C8	RT3 ← RT1 * 1	MA, MB=11, SelCop=*, SelP=11, M7, C7
C9	If (SR.Z==1) goto fetch	C=6, B=0, A0=0, MADDR=0
C10	RT2 ← MAR ← MBR	T1, C0, C5
C11	MBR ← **R1**	T9, SelA=R1_offset, MR=0, M1=0, C1
C12	MP[MAR] ← MBR	TA, TD, BW=11, W
C13	RT1 ← RT2	T5, C4
C14	RT2 ← MAR ← RT1 + 4	MA, MB=10, SelCOP=+, T6, C0, C5
C15	RT1 ← RT3	T7, C4
C16	RT3 ← RT1 – 1	MA, MB=11, SelCOP=-, C6, SelP=11, M7, C7
C17	If (NOT SR.Z==0) goto C12	C=6, B=0, A0=0, MADDR=C12
C18	Jump to fetch	A0, C=0, B

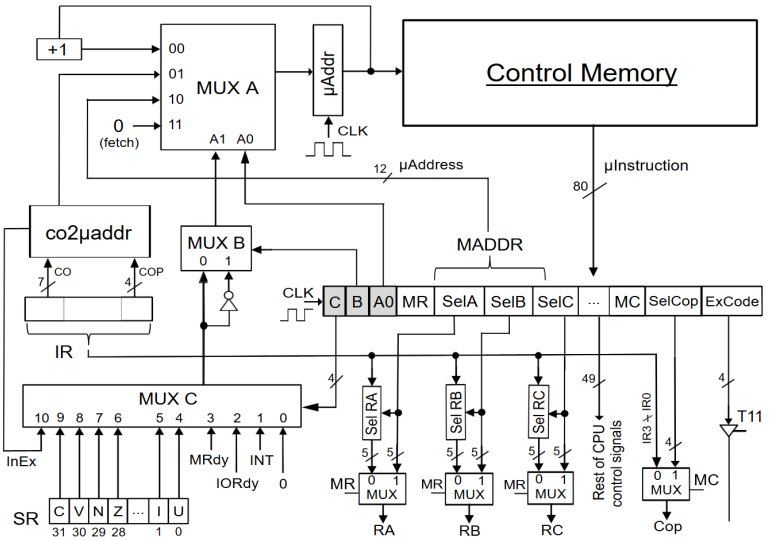
Exercise 16. Given the processor with the following structure:



And with the following characteristics:

- A register file of registers with 32 32-bit registers, capable of processing 127 different instructions.
- An ALU with 16 arithmetic-logical operations that include addition, subtraction, multiplication, the and, or, not, xor at the bit level, the right and left rotation of a bit, and the logical and arithmetic displacements to the right, and the logical displacement to the left.
- Byte-level addressable memory and one single clock cycle for reading or writing.

This processor has a Control Unit represented by the following figure:



Answer the following questions briefly and justifiably:

a) What is the advantage of the microprogrammed control technique over the wired technique? What is the advantage of implicit sequencing over explicit sequencing?

b) Consider the statement **reset_v R1, address**. This statement is a vector statement that stores the value zero in the *N* consecutive memory addresses from *address*. The *N* value is located in the *R1* register. That is, it stores the zero value in the memory positions:
`address, address +4, address +8 . . . . . address + (N-1)x4.`
Properly specify the format of this instruction.

c) Specify the elementary operations and all the necessary control signals, both from the control unit and the processor, to execute the machine instruction described above. Include the *fetch*cycle.

Assume that R29 acts as a stack pointer, that the stack pointer points to stack top, and that the stack grows toward decreasing memory directions. Also assume that the R0 register is always zero, and that the state register does not need to be saved, the instruction can modify it in any way it deems appropriate.

Solution:

a) The microprogrammed control allows you to reprogram the microcode (if a rewritable ROM is used). Implicit sequencing allows to reduce the number of bits in the control memory since most of the time the next microinstruction to execute is the next one in the control memory, and in case of jump the microdirection can overlap with other control fields.

b) A possible format for the instruction described would be:

First word:

Instruction code **(7-bit)**	Register **(5-bit)**	**(20-bit)**

Second word:

Address **(32-bit)**

c) Elementary operations and control signals:

Cycle	Elementary operations	Control signals
Fetch		
C1	MAR ← PC	T2, C0
C2	MBR ← Mem[MAR]	TA, R, BW=11, C1, M1
C3	PC ← PC+4, IR ← MBR	M2, C2, T1, C3
S4	Decode (Jump to co2µaddr)	A0, B=0, C=0
Execution of the instruction		
C5	MAR ← PC	T2, C0
C6	MBR ← Mem[MAR], PC ← PC + 4	TA, R, BW=11, M1, C1, M2, C2
C7	RT2 ← MBR	T1, C5
C8	RT1 ← **R1** * 4	SelA=**Offset_R1**, MB=10, SelCop=*, MC, T6, C4
C9	MBR ← R0	SelA=0, MR, T9, M1=0, C1
C10	RT1 ← RT1 – 4, SR ← flags	MA, MB=10, SelCop=-, MC, T6, C4, SelP=11, M7, C7
C11	If (SR.N==1) goto fetch	C=111, B=0, A0=0, MADDR=fetch
C12	MAR ← RT2 + RT1	MA, MB=01, SelCop=+, MC, T6, C0
C13	Mem[MAR] ← MBR	TA, TD, W, BW=11,
C14	Salto C10	C=0, B=1, A0=0, MADDR=C10

Exercise 17. A computer with a clock frequency of 2 GHz is available. The average number of cycles per machine instruction is 20. This computer is connected to a device that generates 100,000 interruptions per second. The interruption treatment routine executes 500 instructions. It is requested:

a) Calculate the percentage of time this computer spends treating this device.

b) Could this computer attend to all the interruptions generated by this device, if the interruption treatment routine had 2000 machine instructions? Reason your answer.

Solution:

a) The computer has a clock frequency of 2 GHz = 2 x 10^9 cycles/second.
The treatment routine executes 500 machine instructions, which means: 500 x 20 = 10000 = 10^4 clock cycles consumed for each interruption generated. As the device generates 100 000 interrupts per second, the number of cycles per second dedicated to treating this device is $10^{x\,5}$ x 10^4 = 10^9 cycles per second.

As the clock frequency is 2 x 10^9 cycles/second, the percentage of time spent dealing with the routine would be:
10^9 / (2 x 10^9) x 100 = 50%.

a) If the routine had 2000 machine instructions, it would require 2000 x 20 = 4 x 10^4 clock cycles. If 100 000 interruptions per second are generated, 10 5x^4 x 10 4=^{4} x 10 9 cycles would be necessary. As the computer only has = 2 x 10^9 cycles/second, it does not have the capacity, therefore, to be able to attend all the interruptions generated by this device.