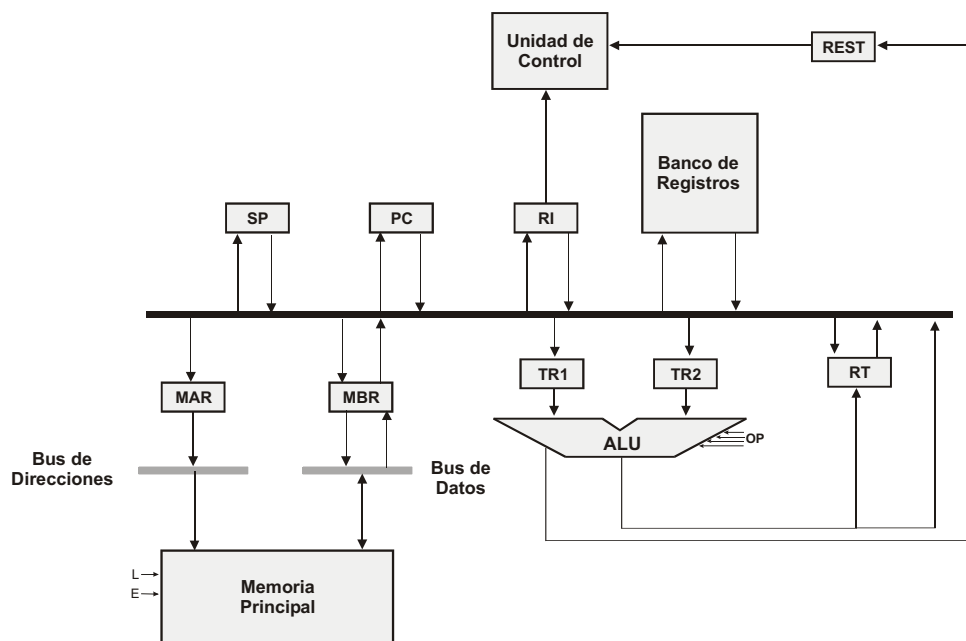


Unidad de control y diseño del procesador

Ejercicios resueltos

Ejercicio 1. La siguiente figura muestra la estructura de un computador con una CPU de 32 bits y un banco de 16 registros generales (R1 a R16). MAR es el registro de direcciones de memoria y MBR el registro de datos de memoria. El registro RT es un registro temporal transparente al usuario. TR1 y TR2 también son registros transparentes al usuario. La ALU permite realizar cuatro operaciones: suma, resta, incremento en 1 y decremento en 1. Este computador representa los números enteros en complemento a dos y los números en coma flotante según el estándar IEEE 754 de precisión simple. Suponga que se requiere un ciclo para acceder a memoria y que esta se accede a nivel de palabra y que la decodificación requiere un ciclo.



Responda a las siguientes preguntas, considerando que las instrucciones ocupan una palabra.

- Micro programe a nivel de operaciones elementales la instrucción STORE Rk, /dirección. ¿Cuántos ciclos se necesitan para ejecutar esta instrucción en este computador? Esta instrucción almacena el contenido del registro Rk en la posición de memoria dada por /dirección.
- Micro programe a nivel de operaciones elementales la instrucción JMP (Rk). ¿Cuántos ciclos se necesitan para ejecutar esta instrucción en este computador? Esta instrucción salta a la dirección de memoria almacenada en Rk.
- Micro programe a nivel de operaciones elementales la instrucción ADD R2, [#700[R7++]]. Esta instrucción suma el contenido del registro R2 con el contenido de una posición de memoria y deja el resultado en R2. Los [] representan direccionamiento indirecto.

Solución:

- a) Instrucción STORE Rk, /dirección

c1: MAR \leftarrow PC
c2: MBR \leftarrow MP
TR1 \leftarrow PC
c3: PC \leftarrow TR1 + 1
c4: R1 \leftarrow MBR
c5: decodificación
c6: MBR \leftarrow RK
c7: MAR \leftarrow RI(dirección)
c8: MP \leftarrow MBR

Observe que para poder incrementar el contador del programa ($PC \leftarrow PC + 1$) en el ciclo de captación de la instrucción, es necesario llevarlo previamente al registro transparente al usuario TR1 (o bien TR2) y a continuación incrementarlo y almacenarlo en el PC. Para ello se requieren las dos siguientes operaciones elementales:

$TR1 \leftarrow PC$
 $PC \leftarrow TR1 + 1$

b) Instrucción JMP (Rk)

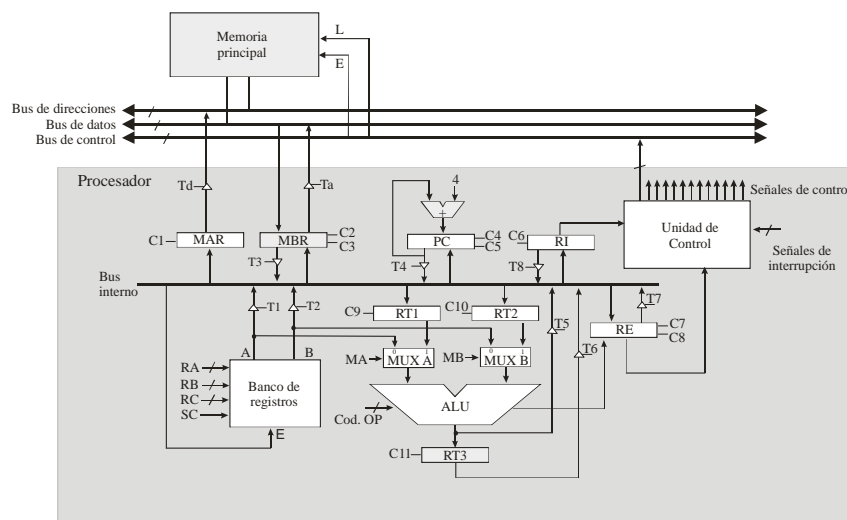
c1: $MAR \leftarrow PC$
c2: $MBR \leftarrow MP$
 $TR1 \leftarrow PC$
c3: $PC \leftarrow TR1 + 1$
c4: $RI \leftarrow MBR$
c5: decodificación
c6: $PC \leftarrow RK$

c) ADD R2, [#700[R7++]]

En esta instrucción se utiliza direccionamiento indirecto relativo, por lo que se deben hacer dos accesos a Memoria Principal para obtener el segundo operando de la suma.

c0: $MAR \leftarrow PC$
c1: $MBR \leftarrow MP$
 $TR1 \leftarrow PC$
c2: $PC \leftarrow TR1 + 1$
c3: $RI \leftarrow MBR$
c4: decodificación
c5: $TR1 \leftarrow RI(700)$
c6: $TR2 \leftarrow R7$
c7: $MAR \leftarrow TR1 + TR2$
c8: $MBR \leftarrow MP$
c9: $RT1 \leftarrow MBR$
c10: $TR2 \leftarrow R2$
c11: $R2 \leftarrow TR1 + TR2$
c12: $RT1 \leftarrow R7$
c13: $R7 \leftarrow RT1 + 1$

Ejercicio 2. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo. El procesador es capaz de ejecutar 240 instrucciones máquina.

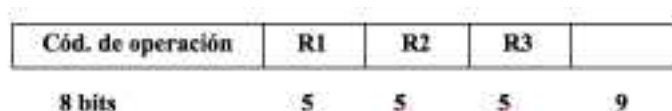


Se pide:

- ¿Qué es y para qué sirve el registro de instrucción? ¿Qué es y para qué sirve el registro de estado?
- Considere la siguiente instrucción hipotética: `addr R1, R2, (R3)`, donde R1, R2 y R3 representan registros y (R3) representa direccionamiento indirecto de registro. Indique un posible formato para la instrucción.
- Indique las operaciones elementales y las señales de control necesarias para ejecutar la instrucción anterior, teniendo en cuenta que esta instrucción suma el contenido del registro R2 con el contenido de la posición de memoria que se encuentra almacenada en el registro R3. El resultado se almacena en R1.

Solución:

- El registro de instrucción almacena la instrucción a ejecutar. El registro de estado almacena información sobre el estado del programa en curso.
- Como el computador ejecuta 240 instrucciones, se necesitan 8 bits para representar el código de operación ($2^8 = 256$). Como el computador dispone de 32 registros, se necesitan 5 bits para representar un registro. La instrucción ocupa una palabra y su formato es:



- A continuación se muestran las operaciones elementales:

Ciclo 0: $MAR \leftarrow PC$	Señales: T4, C3
Ciclo 1: $MBR \leftarrow MP$	Señales: Td, L, C2
$PC \leftarrow PC + 4$	C4
Ciclo 2: $RI \leftarrow MBR$	Señales: T3, C6
Ciclo 3: Decodificación	
Ciclo 4: $MAR \leftarrow R3$	Señales: RA= identificador de R3, T1, C1
Ciclo 5: $MBR \leftarrow MP$	Señales: Td, L, C2
Ciclo 6: $RT1 \leftarrow MBR$	T3, C9
Ciclo 7: $R1 \leftarrow R2 + RT1$	RB = identif. de R2, MA = 1, MB = 0, Cod. Op = suma, T5, Rc = identif de R1, SC, C8 (para actualizar el RE)

Ejercicio 3. Considere un procesador de 32 bits similar al del ejercicio 2 que ejecuta las instrucciones máquina del MIPS. Este computador utiliza un esquema de interrupciones vectorizado. La tabla de vectores de interrupción consta de 256 entradas y se almacena a partir de la dirección de memoria 0. Se pide:

- Considere que una determinada rutina de tratamiento de la interrupción se almacena en la dirección de memoria `$0x001F$` y que el vector de interrupción correspondiente a esta interrupción es el 3. Indique la instrucción o instrucciones máquina necesarias para almacenar en la entrada correspondiente a esta instrucción la dirección de la rutina de tratamiento.

- b) Considere la instrucción de máquina TRAP vector, que activa un servicio del sistema operativo. En esta instrucción, vector indica la entrada de la tabla de vectores de interrupción que incluye la dirección de tratamiento de la rutina. Considere que el puntero de pila es el registro R1. Indique las operaciones elementales necesarias para ejecutar esta instrucción.
- c) Indique las operaciones elementales necesarias para ejecutar la instrucción de máquina RETI, que retorna la ejecución al programa suspendido previamente por la instrucción TRAP.

Solución:

- a) Las instrucciones necesarias son las siguientes:

```
lw    $t0, 0x001F
sw    $t0, 12
```

- b) A continuación se muestran las operaciones elementales necesarias:

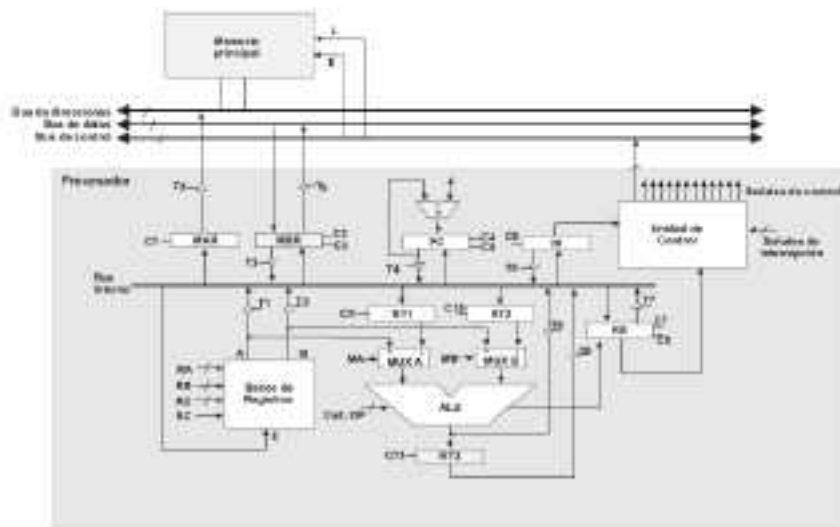
```
C0:    SP, MAR ← Sp - 4
C1:    MBR ← PC
C2:    MB ← MBR
C3:    SP, MAR ← SP - 4
C4:    MBR ← RE
C5:    MP ← MBR
C6:    RT1 ← R0
C7:    RT2 ← RI(vector)
C8:    RT1 ← RT1 + 4
C9:    MAR ← RT1 * RT2
C10:   MBR ← MP
C11:   PC ← MBR
```

Durante estas operaciones elementales se guardan en la pila el contenido del contador de programa y del registro de estado.

- c) Las operaciones elementales, excluyendo el ciclo de *fetch* necesarias para la ejecución de la instrucción RTI son las siguientes:

```
C0:    SP, MAR ← SP + 4
C1:    MBR ← MP
C2:    RE ← MBR
C3:    SP, MAR ← SP + 4
C4:    MBR ← MP
C5:    PC ← MBR
```

Ejercicio 4. Considere un procesador de 32 bits similar al mostrado en la figura que ejecuta las instrucciones máquina del MIPS. La arquitectura tiene 250 instrucciones distintas y el Banco de Registros tiene 128 registros distintos. La lectura y escritura de memoria se realiza en un solo ciclo.



Una instrucción de esta arquitectura es TRIPLEADD, la cual tiene la siguiente sintaxis:

TRIPLEADD RD RO1 RO2 dir

Esta instrucción suma tres números mediante la siguiente operación: $RD \leftarrow (RO1 + RO2) + MP(dir)$

Dicha operación consta de dos sumas: primero suma el contenido del registro RO1 con el del registro RO2. El resultado es a su vez sumado con el contenido de la posición de memoria dir. El resultado final de la suma se almacena en el registro RD. Estos tres registros son registros genéricos del Banco de Registros mientras que dir representa la dirección de memoria de una palabra. Se pide:

- Indique el formato de la instrucción anterior.
- Indique las operaciones elementales necesarias para ejecutar la instrucción anterior.

Solución:

- Formato de la instrucción.

Existen 250 instrucciones distintas, por lo que el código de operación requerirá de 9 bits.
Existen 126 registros, por lo que los campos de registros requerirán de 7 bits.
La dirección de memoria ocupa 32 bits.

El formato, por tanto, es el siguiente:

1ª palabra (32 bits)

CO(9)	R(7)	R(7)	R(7)	Sin usar(2)
-------	------	------	------	-------------

2ª Palabra (32 bits)

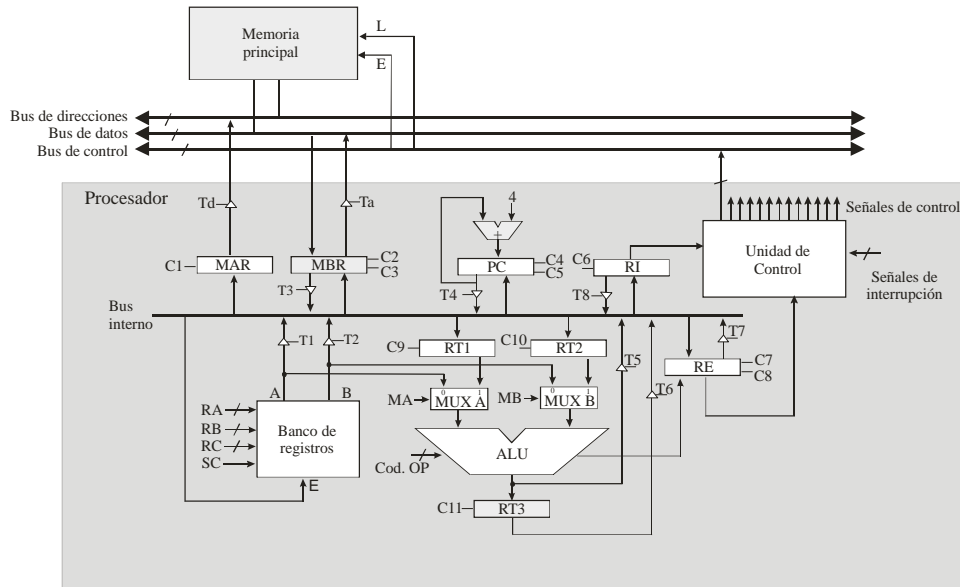
Dir(32)

- Operaciones elementales y señales de control

c1: $MAR \leftarrow PC$	(T4, C1)
c2: Lectura, $PC \leftarrow PC + 4$, $MBR \leftarrow MP$	(Td, L) (C4) (C2)
c3: $RI \leftarrow MBR$	(T3, C6)
c4: Decodificación	
c5: $MAR \leftarrow PC$	(T4, C1)
c6: Lectura, $PC \leftarrow PC + 4$, $MBR \leftarrow MP$	(Td, L) (C4) (C2)

c7: $MAR \leftarrow MBR$ (T3,C1)
c8: Lectura (Td, L)
c10: $RT1 \leftarrow MBR$ (T3,C9)
c11: $RT2 \leftarrow R5 + R7$ (RA=5,RB=7,MUX A=0, MUX B=0, Cod. Op=ADD) (T5,C10)
c12: $R2 \leftarrow RT1 + RT2$ (MUX A=1, MUX B=1, Cod. Op=ADD) (T5, RC=2, SC)

Ejercicio 5. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo.



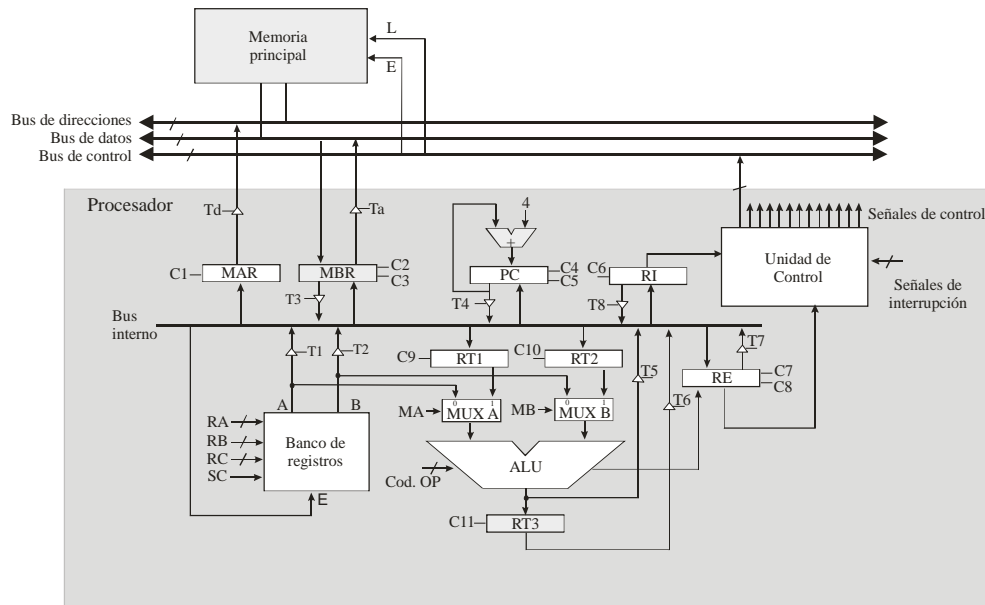
- ¿Qué es y para qué sirve el registro contador de programa? ¿Qué es y para qué sirve el registro de instrucción? ¿Cuántos bits se necesitan para las señales de control RA, RB y C6?
- Indique de forma razonada las señales de control necesarios que hay que activar para la ejecución de la siguiente operación elemental: $MBR \leftarrow RE$
- Indique las operaciones elementales y las señales de control necesarias para ejecutar la instrucción $LOADM\ desp(R1),\ valor$ que copia el valor inmediato valor en la posición de memoria indicada por $desp(R1)$, teniendo en cuenta que se utiliza para este campo direccionamiento relativo a registro base. Considere que la instrucción ocupa una palabra.

Solución:

- El contador de programa es el registro que almacena la dirección de la siguiente instrucción a ejecutar. El registro de instrucción es el registro que almacena la instrucción que se está ejecutando en un momento dado. Se necesitan 5 bits para las señales RA y RB, puesto que hay 32 registros y se necesitan 5 bits para representar 32. Para la señal C6 se necesita un único bit
- Hay que activar las señales T7 y C3.
- A continuación se muestran las operaciones elementales y las señales de control.

Ciclo	Operación elemental	Señales de control activadas
C1	$MAR \leftarrow PC$	T4, C1
C2	$MBR \leftarrow MP$	
	$PC \leftarrow PC + 4$	L, Td, C2, C4
C3	$RI \leftarrow MBR$	T3, C6
C4	Decodificación	
C5	$RT1 \leftarrow RI(desp)$	T8, C9
C6	$MAR \leftarrow R1 + RT1$	RB = <dir de R1>, MA = 1, MB = 0, Cod op = SUMAR, T5, C1
C7	$MBR \leftarrow RI(valor)$	T8, C3
C8	$MP \leftarrow MBR$	Ta, E

Ejercicio 6. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo.



Este computador dispone del juego de instrucciones del MIPS32. Indique las operaciones elementales y las señales de control (incluyendo el fetch) necesarias para ejecutar las siguientes instrucciones:

- `addi $t0, $t1, 10`
- `lw $t1, ($t2)`
- `sw $t1, 120($t2)`

Solución:

- En la siguiente tabla se muestran las operaciones elementales y las señales de control para la instrucción `addi`.

Ciclo	Operación elemental	Señales de control activadas
C1	$MAR \leftarrow PC$	T4, C1
C2	$MBR \leftarrow MP$, $PC \leftarrow PC + 4$	L, Td, C2, C4
C3	$RI \leftarrow MBR$	T3, C6
C4	Decodificación	
C5	$RT2 \leftarrow RI(10)$	T8, C10
C6	$\$t0 \leftarrow \$t1 + RT2$	RA = <dir de \$t1> MA = 0 MB = 1 Cod op = SUMAR T5 RC = <dir de \$t0> SC

- En la siguiente tabla se muestran las operaciones elementales y las señales de control para la instrucción `lw`.

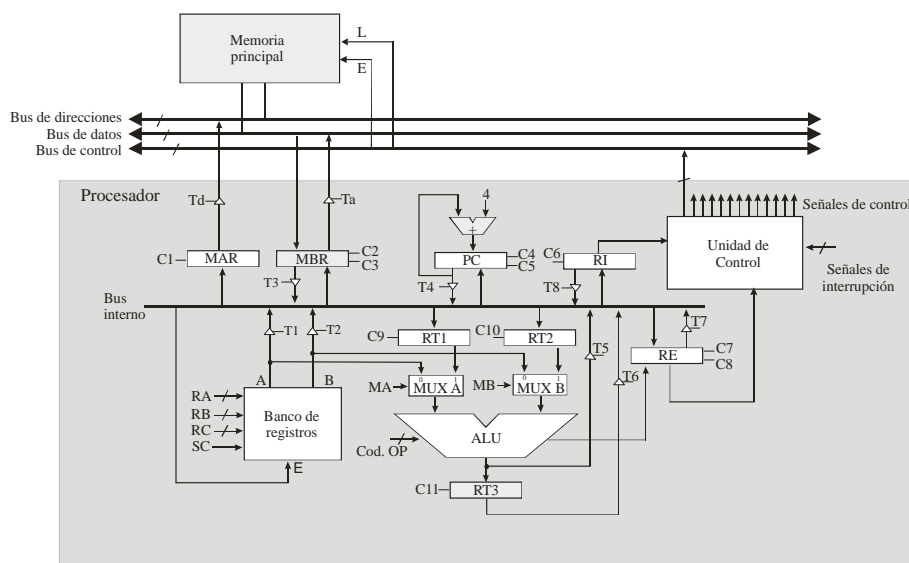
Ciclo	Operación elemental	Señales de control activadas
C1	$MAR \leftarrow PC$	T4, C1
C2	$MBR \leftarrow MP$, $PC \leftarrow PC + 4$	L, Td, C2, C4
C3	$RI \leftarrow MBR$	T3, C6

C4	Decodificación	
C5	$MAR \leftarrow \$t2,$	$RA = \langle \text{dir de } \$t2 \rangle$ T1, C1
C6	$MBR \leftarrow MP$	Td, L, C2
C7	$\$t1 \leftarrow MBR$	T3, SC, $RA = \langle \text{dir de } \$t1 \rangle$

c) A continuación se muestran las operaciones elementales y señales para la instrucción sw.

Ciclo	Operación elemental	Señales de control activadas
C1	$MAR \leftarrow PC$	T4, C1
C2	$MBR \leftarrow M[MAR]$	Td, L, C2
C3	$RI \leftarrow MBR, PC \leftarrow PC + 4$	T3, C6, C4
C4	Decodificación	
C5	$RT2 \leftarrow RI(120)$	T8, C10
C6	$RT1 \leftarrow \$t2$	$RA = \$t2, T1, C9$
C7	$MAR \leftarrow RT1 + RT2$	$MA=1, MB=1, \text{Cod. OP}=+, T5, C1$
C8	$MBR \leftarrow \$t1$	$RA = \$t1, C3$
C9	$MP[MAR] \leftarrow MBR$	Ta, Td, E
C10	Salto a fetch	

Ejercicio 7. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo.



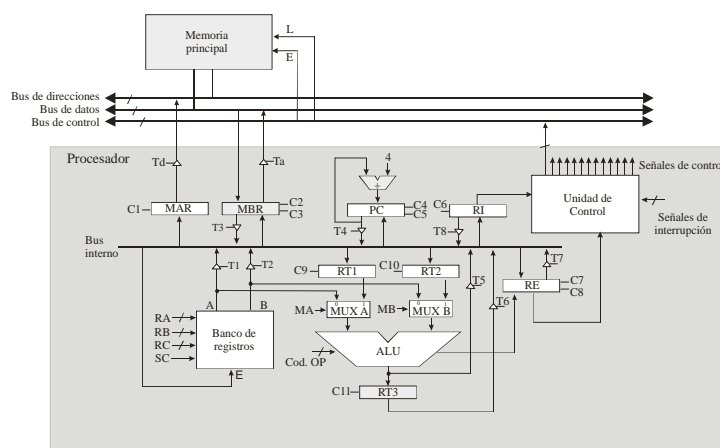
Este computador dispone del juego de instrucciones del MIPS32. Se pide:

- Indique las señales de control necesarias para poder realizar la operación elemental $PC \leftarrow R7$, siendo R7 el registro del banco de registros cuyo número es el 7.
- Si durante un ciclo de reloj se activan las señales T3 y C10, indique qué operación elemental se está realizando.
- Indique las operaciones elementales y las señales de control (incluyendo el *fetch*) necesarias para ejecutar la instrucción `addi $t0, $t1, 10`.

Solución:

- Han de activarse las señales C5, T1 y RA = 00111 (registro 7)
- Se trata de la operación $RT2 \leftarrow MBR$
- Ciclo 0: $MAR \leftarrow PC$ Señales: T4, C3
Ciclo 1: $MBR \leftarrow MP$ Señales: Td, L, C2
 $PC \leftarrow PC + 4$ C4
Ciclo 2: $RI \leftarrow MBR$ Señales: T3, C6
Ciclo 3: Decodificación
Ciclo 4: $RT1 \leftarrow RI(10)$ Señales: T8, C9
Ciclo 5: $\$t0 \leftarrow RT1 + \$t1$ Señales: MA = 1, Cod. OP = suma, RB = $\langle \$t1 \rangle$, T5, RC = $\langle \$t0 \rangle$, SC, C8

Ejercicio 8. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo.



Este computador dispone del juego de instrucciones del MIPS32. Se pide:

- Indique las señales de control necesarias para poder realizar la operación elemental $RT1 \leftarrow \$t2$, siendo $\$t2$ el registro número 2 del banco de registros.
- Indique las operaciones elementales y las señales de control (incluyendo el *fetch*) necesarias para ejecutar la instrucción hipotética $addm \$t1, (\$t2)$. Esta función suma el contenido del registro $\$t1$ con el contenido de la posición de memoria, que se encuentra almacena en $\$t2$. El resultado se almacena en el registro $\$t1$. Es decir: $\$t1 \leftarrow \$t1 + MP[\$t2]$

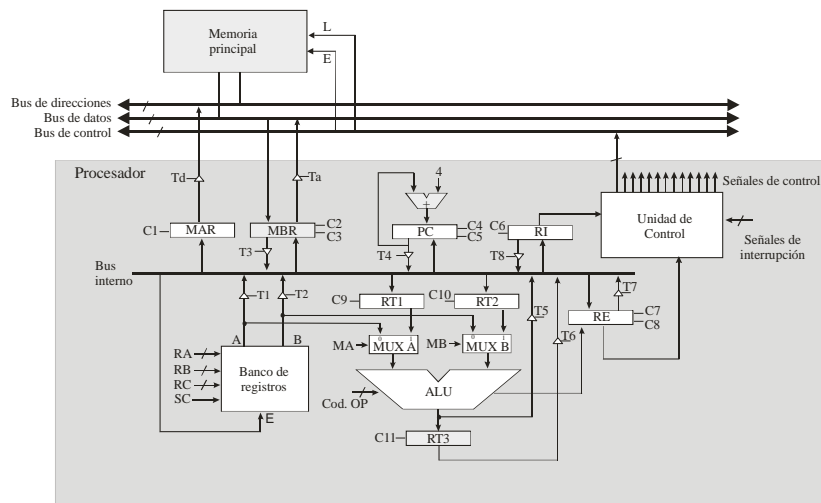
Solución:

- Hay que activar las señales C9, T1, RA = 00010
-

Ciclo	Op. Elemental	Señales de control
Ciclo 0:	$MAR \leftarrow PC$	Señales: T4, C3
Ciclo 1:	$MBR \leftarrow MP$	Señales: Td, L, C2
	$PC \leftarrow PC + 4$	C4
Ciclo 2:	$RI \leftarrow MBR$	Señales: T3, C6
Ciclo 3:	Decodificación	
Ciclo 4:	$MAR \leftarrow \$t2$	Señales: RA= identificador de $\$t2$, T1, C1
Ciclo 5:	$MP \leftarrow MBR$	L
Ciclo 6:	$RT1 \leftarrow MBR$	T3, C9
Ciclo 7:	$\$t1 \leftarrow RT1 + \$t1$	MA = 1, RB = id. Del registro $\$t1$, Cod. Op = sumar, T5, C8, RC = id. Del registro $\$t1$, SC

Ejercicio 9. Considere un computador de 32 bits, que dispone de un banco de 32 registros, que se encuentra conectado a una memoria, que se direcciona por bytes y que requiere un ciclo para las operaciones de lectura y escritura. El computador dispone de un juego de instrucciones con 130 instrucciones máquina. La unidad aritmético-lógica es capaz

de realizar operaciones aritméticas y lógicas (sumar, restar, multiplicar, dividir, incrementar, disminuir en uno, etc.). El procesador emplea un ciclo para la decodificación de la instrucción. Su estructura es la siguiente:



Considere las dos siguientes instrucciones:

`moveM dir1, dir2` Esta instrucción mueve el contenido de la posición de memoria `dir2` a la posición de memoria `dir1`.

`moveR R1, R2` Esta instrucción mueve el contenido del registro `R2` al registro `R1` (`R1` y `R2` son los registros 1 y 2 del banco de registros).

Se pide:

- Indique, de forma razonada, el formato de las instrucciones anteriores.
- Indique, en función del formato definido en el apartado a, las operaciones elementales y las señales de control necesarias para la ejecución de las instrucciones anteriores. Incluya también el ciclo de fetch.

Solución:

- Como el computador dispone de 130 instrucciones, se necesitan 8 bits para codificar una instrucción. El computador es de 32 bits, por tanto, se necesitan 32 bits para codificar una dirección de memoria. Esta instrucción, por tanto, necesita tres palabras para su codificación. Su contenido es el siguiente:

CO (8 bits)		Dir 1 (32 bits)	Dir 2 (32 bits)
--------------	--	-----------------	-----------------

El formato de la segunda instrucción sería:

CO (8 bits)	R1 (5 bits)	R2 (5 bits)	
--------------	--------------	--------------	--

- Primera instrucción:

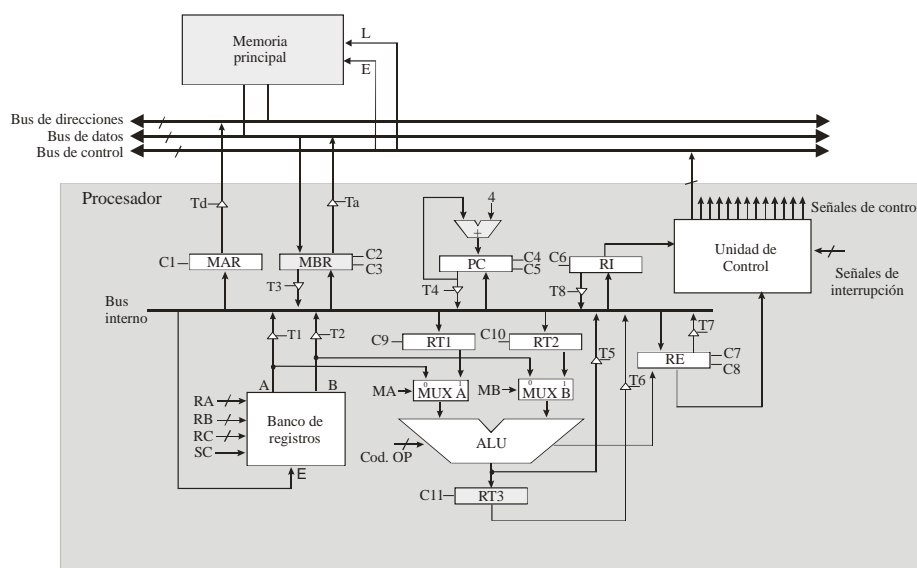
ciclo	Operaciones elementales	Señales de control
C0	$MAR \leftarrow PC$	T4, C1
C1	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4
C2	$RI \leftarrow MBR$	T3, C6
C3	Decodificación	
C4	$MAR \leftarrow PC$	T4, C1
C5	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4

C6	$RT1 \leftarrow MBR, RT2 \leftarrow MBR$	T3, C9, C10
C7	$MAR \leftarrow PC,$	T4, C1
C8	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4
C9	$MAR \leftarrow MBR$	T3, C1
C10	$MBR \leftarrow MP$	Td, L, C2
C11	$MAR \leftarrow RT1 \text{ OR } RT2$	MA, MB, T5, C1, Cod. OP = OR
C12	$MP \leftarrow MBR$	Td, Ta, E

Segunda instrucción:

Ciclo	Operaciones elementales	Señales de control
C0	$MAR \leftarrow PC$	T4, C1
C1	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4
C2	$RI \leftarrow MBR$	T3, C6
C3	Decodificación	
C4'	$R1 \leftarrow R2$	RA = 00001, RC = 00010, T1, SC

Ejercicio10. Considere un computador de **32 bits**, que dispone de un banco de **32 registros**, que se encuentra conectado a una memoria, que se **direcciona por bytes** y que requiere **un ciclo para las operaciones de lectura y escritura**. El computador dispone de un juego de instrucciones con **110 instrucciones máquina**. La unidad aritmético-lógica es capaz de realizar operaciones aritméticas y lógicas (sumar, restar, multiplicar, dividir, incrementar, disminuir en uno, etc.). El procesador emplea un ciclo para la decodificación de la instrucción. Su estructura es la siguiente:



Considere las dos siguientes instrucciones:

PUTS R1, R2	Esta instrucción inserta dos palabras en la cima de la pila, primero el contenido del registro R1 y encima el contenido del registro R2
ADDS	Esta instrucción extrae de la pila el contenido de las dos primeras palabras apiladas. Realiza la suma de estos dos valores e inserta el resultado en la cima de la pila
POP R1	Esta función extrae un elemento (palabra de la pila) y lo almacena en R1

Considerando que el registro puntero de pila es el registro R30 y que la unidad aritmética lógica dispone entre otros de una operación que permite sumar 4 al dato que entra en la ALU por el MUX A y otra que permite restar 4 al dato que entra a la ALU por el MUX A. Se pide:

- Indique, de forma razonada, el formato de las instrucciones anteriores.
- Indique, en función del formato definido en el apartado a, las operaciones elementales y las señales de control

- necesarias para la ejecución de la instrucción PUTS. Incluya también el ciclo de *fetch*.
- c) Utilizando las tres instrucciones anteriores, escriba un fragmento de programa cuyo comportamiento sea similar al que realiza la instrucción de MIPS `add R1, R2, R3`.

Solución:

- a) Como el computador dispone de 110 instrucciones, se necesitan 8 bits para codificar una instrucción. El computador es de 32 bits, por tanto, se necesitan 32 bits para codificar una dirección de memoria. Esta instrucción, por tanto, necesita tres palabras para su codificación. Su contenido es el siguiente:
El formato de la instrucción PUTS es:

CO (8 bits)	R1 (5 bits)	R2 (5 bits)	Sin usar (14 bits)
--------------	--------------	--------------	--------------------

El formato de la segunda instrucción sería:

CO (8 bits)	Sin usar (24 bits)
--------------	--------------------

El formato de la tercera instrucción es:

CO (8 bits)	R1 (5 bits)	Sin usar (19 bits)
--------------	--------------	--------------------

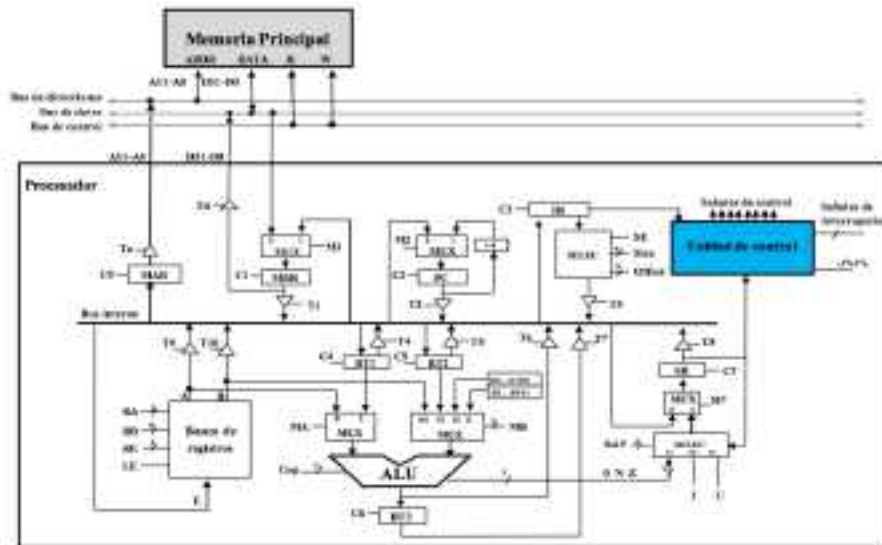
- b) Instrucción PUTS:

ciclo	Operaciones elementales	Señales de control
C0	$MAR \leftarrow PC$	T4, C1
C1	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4
C2	$RI \leftarrow MBR$	T3, C6
C3	Decodificación	
C4	$R30 \leftarrow R30 - 4$	RA = 11110 (30 en binario) MA = 0 Cod. Op = restar 4 T5, RC = 11110, SC
C5	$MAR \leftarrow R30$	RA = 11110 (30 en binario) T1 C1
C6	$MBR \leftarrow R1$	RA = 00001, T1, C3
C7	$MP \leftarrow MBR$	Td, Ta, E
C7	$R30 \leftarrow R30 - 4$	RA = 11110 (30 en binario) MA = 0 Cod. Op = restar 4 T5, RC = 11110, SC
C8	$MAR \leftarrow R30$	RA = 11110 (30 en binario) T1 C1
C9	$MBR \leftarrow R2$	RA = 00010, T1, C3
C10	$MP \leftarrow MBR$	Td, Ta, E

- c) El fragmento necesario sería:

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

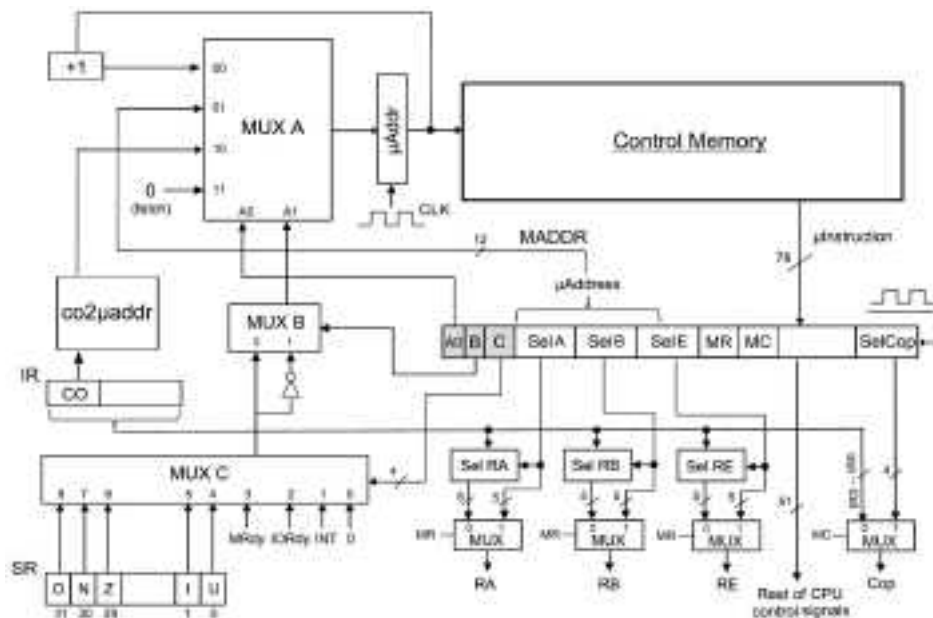
Ejercicio 11. Dado el procesador con la siguiente estructura:



Y con las siguientes características:

- Un banco de registros con 32 registros de 32 bits, capaz de procesar 127 instrucciones distintas.
- Una ALU con 16 operaciones aritmético-lógicas que incluyen la suma, resta, multiplicación, los and, or, not, xor a nivel de bit, la rotación a la derecha e izquierda de un bit, y los desplazamientos lógicos y aritméticos a la derecha, y el desplazamiento lógico a la izquierda.
- Memoria direccionable a nivel de byte y un solo ciclo de reloj para la lectura o escritura.

Este procesador dispone de una Unidad de control representada por la siguiente figura:



Responda de forma breve y justificada a las siguientes preguntas:

- Analizando el diagrama de la Unidad de Control, ¿Qué técnica de control emplea? ¿Qué tipo de secuenciamiento usa?
- Dada la instrucción de dos palabras `li w R valor` que almacena “valor” en el registro “R”, indique un formato de instrucción para la misma de forma que en la primera palabra contenga el registro “R” y en la segunda palabra esté “valor” como número binario de 32 bits.

- c) Especifique las operaciones elementales y todas las señales de control necesarias, tanto de la unidad de control como del procesador, para ejecutar la instrucción máquina *li.w R* valor anteriormente descrita. Incluya el ciclo de *fetch*.

NOTA: Asuma que R29 actúa como puntero de pila, que el puntero de pila apunta a cima de pila y que la pila crece hacia direcciones decrecientes de memoria.

Solución:

- a) Esta Unidad de Control tiene una técnica de control microprogramado y un secuenciamiento implícito.
b) Un posible formato para la instrucción descrita sería:

Primera palabra:

Código de instrucción (7 bits)	Registro (5 bits)	
--------------------------------	-------------------	--

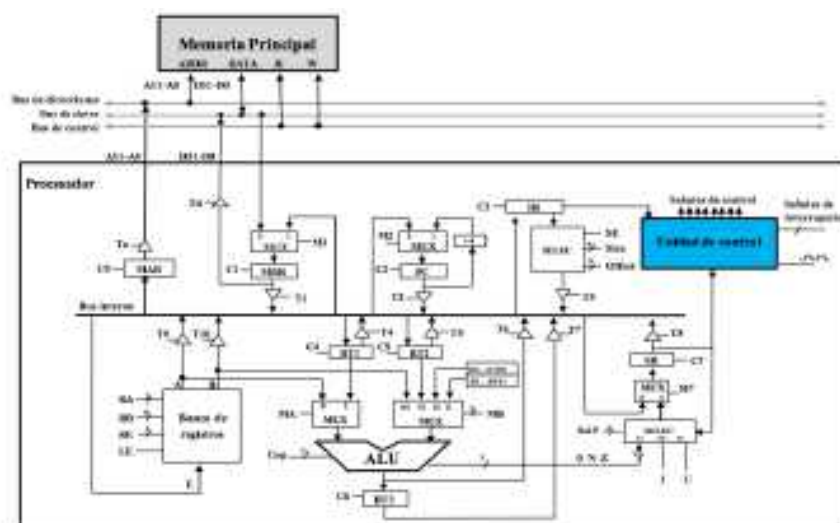
Segunda palabra:

Valor (32 bits)

- c) Las operaciones elementales y las señales de control:

Operaciones elementales	Señales de control
fetch	
$MAR \leftarrow PC$	T2, C0
$MBR \leftarrow Mem[MAR]$	TA, R, BW=11, C1=1
$PC \leftarrow PC+4$, $IR \leftarrow MBR$	M2, C2, T1, C3
Salto a Código de operación	A0, B=0, C=0
li.w R valor	
$MAR \leftarrow PC$, $PC \leftarrow PC+4$	T2, C0, M2=1, C2
$MBR \leftarrow Mem[MAR]$	TA=1, R=1, BW=11, C1=1
$R \leftarrow MBR$, salto a fetch	T1=1, LE=1, MR=0, SELE=10100, A0=1, B=1, C=0

Ejercicio 12. Dado el procesador con la siguiente estructura:

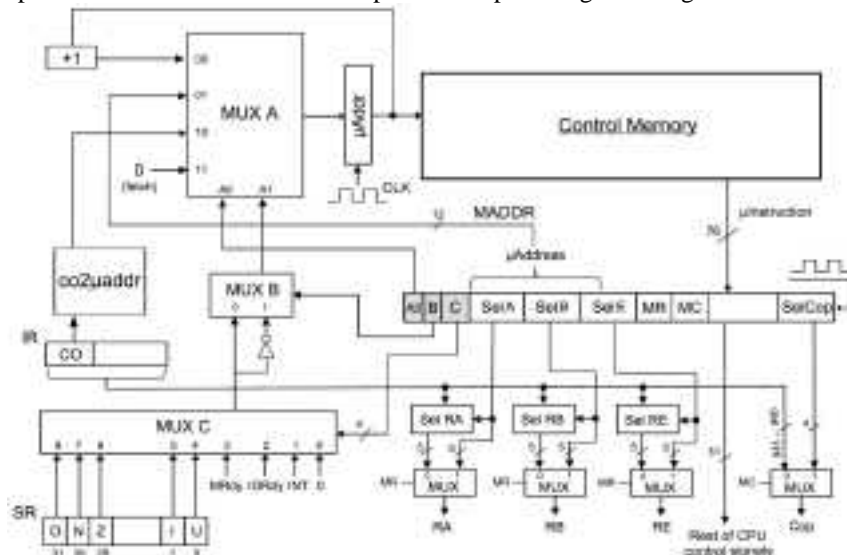


Y con las siguientes características:

- Un banco de registros con 32 registros de 32 bits, capaz de procesar 63 instrucciones distintas.

- Una ALU con 16 operaciones aritmético-lógicas que incluyen la suma, resta, multiplicación, los and, or, not, xor a nivel de bit, la rotación a la derecha e izquierda de un bit, y los desplazamientos lógicos y aritméticos a la derecha, y el desplazamiento lógico a la izquierda.
- Memoria direccionable a nivel de byte y un solo ciclo de reloj para la lectura o escritura.

Este procesador dispone de una Unidad de control representada por la siguiente figura:



Responda de forma breve y justificada a las siguientes preguntas:

- A la hora de implementar una Unidad de Control, ¿Qué técnicas de control es posible usar? ¿Qué tipo de secuenciamiento es posible usar en una Unidad de Control?
- Considere la instrucción `add.w R1 R2 valor` que almacena en el registro “R1” la suma del contenido de “R2” y “valor”. Esta instrucción suma números en complemento a dos. Para el valor inmediato “valor” se utiliza el mismo formato que el que se almacena en el registro R1 y R2. Indique un formato válido para dicha instrucción.
- Especifique las operaciones elementales y señales de control necesarias para ejecutar la instrucción máquina `add.w R1 R2 valor` anteriormente descrita. Incluya el ciclo de *fetch*.

Asuma que R29 actúa como puntero de pila, que el puntero de pila apunta a cima de pila y que la pila crece hacia direcciones decrecientes de memoria.

Solución:

- La Unidad de Control puede utilizar lógica **microprogramada** o bien lógica **cableada**. Respecto al secuenciamiento, puede ser **implícito o explícito**.
- Un posible formato para la instrucción descrita sería:

Primera palabra:

Código de instrucción (6 bits)	Registro (5 bits)	Registro (5 bits)	Relleno (16 bits)
--------------------------------	-------------------	-------------------	-------------------

Segunda palabra:

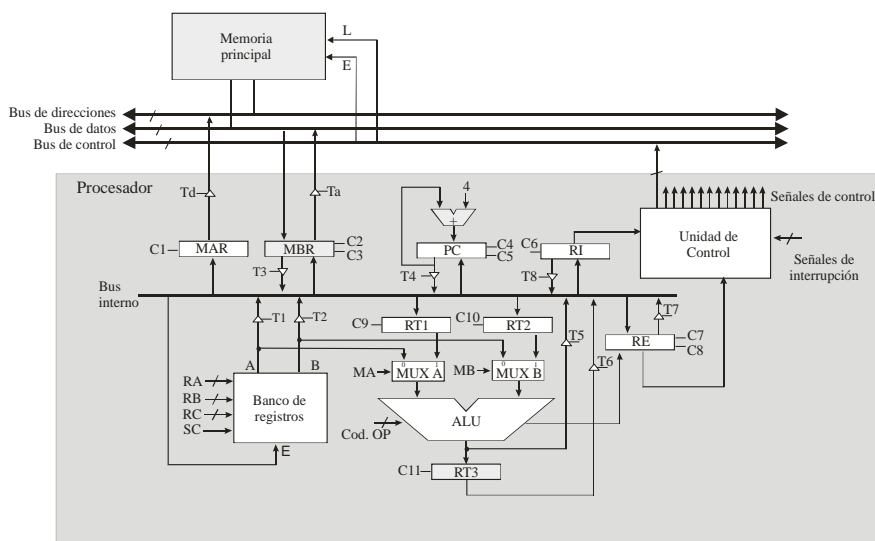
Valor (32 bits)

- Las operaciones elementales y las señales de control:

Operaciones elementales	Señales de control
fetch	
MAR <- PC	T2, C0

$MBR \leftarrow Mem[MAR]$	TA, R, BW=11, C1=1
$PC \leftarrow PC+4, IR \leftarrow MBR$	M2, C2, T1, C3
Salto a c.o.	A0, B=0, C=0
li.w R1 R2 valor	
$MAR \leftarrow PC, PC \leftarrow PC+4$	T2, C0, M2=1, C2
$MBR \leftarrow Mem[MAR]$	TA=1, R=1, BW=11, C1=1
$RT1 \leftarrow MBR$	T1, C4
$RT2 \leftarrow IR(R2)$	SE=0, Size=5, Offset=16, T3, C5
$R1 \leftarrow RT1+RT2$, salto a fetch	T1=1, LE=1, MR=0, SELE=10101, A0=1, B=1, C=0

Ejercicio 13. Considere un computador de 32 bits, que dispone de un banco de 32 registros, que se encuentra conectado a una memoria, que se direcciona por bytes y que requiere un ciclo para las operaciones de lectura y escritura. El computador dispone de un juego de instrucciones con 120 instrucciones máquina. La unidad aritmético-lógica es capaz de realizar operaciones aritméticas y lógicas (sumar, restar, multiplicar, dividir, incrementar, disminuir en uno, etc.). El procesador emplea un ciclo para la decodificación de la instrucción. El registro R0 tiene su valor cableado a 0, y se emplea el registro R31 como registro puntero de pila. Su estructura es la siguiente:



Considere la siguiente instrucción máquina `swapSP R1`. Esta instrucción intercambia el contenido almacenado en el registro R1 con el contenido almacenado en la cima de la pila.

Se pide:

- Indique, de forma razonada el formato de la instrucción anterior.
- Indique, en función del formato definido en el apartado a), las operaciones elementales y las señales de control necesarias para la ejecución de la instrucción anterior. Incluya también el ciclo de *fetch*.

Solución:

- Como el computador dispone de 120 instrucciones, se necesitan 7 bits ($\log_2 120$ por exceso) para codificar una instrucción. Como hay 32 registros, se necesitan 5 bits para identificar a cada registro ($2^5 = 32$). Esta instrucción, por tanto, necesita una palabra para su codificación. Su contenido es el siguiente:

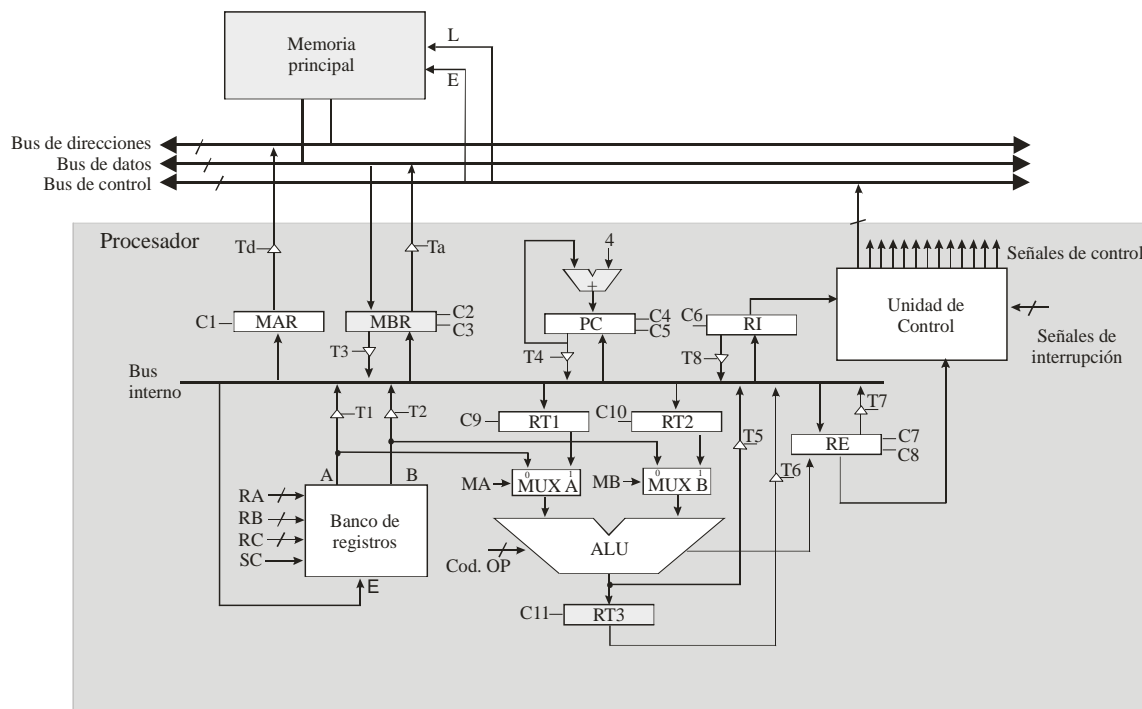
CO (7 bits)	R1 (5 bits)	
--------------	--------------	--

- A continuación, se muestran los ciclos, operaciones elementales y señales de control necesarias, teniendo en cuenta que el puntero de pila es el registro R31:

Ciclo	Operaciones elementales	Señales de control
-------	-------------------------	--------------------

C0	$MAR \leftarrow PC$	T4, C1
C1	$MBR \leftarrow MP, PC \leftarrow PC + 4$	Td, L, C2, C4
C2	$RI \leftarrow MBR$	T3, C6
C3	Decodificación	
C4	$MAR \leftarrow R31$	RA=11110, T1, C1
C5	$MBR \leftarrow MP$	L, TD, TA, C2
C6	$RT2 \leftarrow MBR$	T3, C10
C7	$MBR \leftarrow R1$	RA=00001, T1, C3
C8	$MP \leftarrow MBR$ $R1 \leftarrow R0 + RT2$	TD, TA, E Cod. OP=suma, MB, RA=00000, T5, RC=00000, SC

Ejercicio 14. Considere un computador de 32 bits, con una frecuencia de reloj de 2 GHz, que dispone de un banco de 32 registros, que se encuentra conectado a una memoria, que se direcciona por bytes y que requiere un ciclo para las operaciones de lectura y escritura. El computador dispone de un juego de instrucciones con 120 instrucciones máquina. La unidad aritmético-lógica es capaz de realizar operaciones aritméticas y lógicas (sumar, restar, multiplicar, dividir, incrementar, disminuir en uno, etc.). El procesador emplea un ciclo para la decodificación de la instrucción. El registro R0 tiene su valor cableado a 0, y se emplea el registro R31 como registro puntero de pila. Su estructura es la siguiente:



Considere la siguiente instrucción máquina, cuyo código de operación es ADDSP R1, dir. Esta instrucción suma el contenido almacenado en el registro R1 con el contenido almacenado en la dirección de memoria dir y deja en resultado en la cima de la pila.

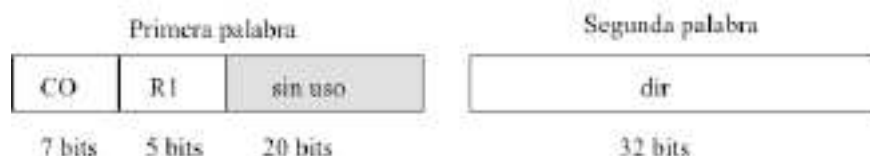
Se pide:

- Indique, de forma razonada el formato de la instrucción anterior.
- Indique, en función del formato definido en el apartado a), las operaciones elementales y las señales de control necesarias para la ejecución de la instrucción anterior. Incluya también el ciclo de *fetch*.
- Considere que el 20% de las instrucciones de este computador necesitan el mismo número de ciclos para su ejecución que la instrucción ADDSP y que el 80% restante necesita en media 7 ciclos. Calcule el valor MIPS para este computador.

Solución:

- a) Para representar la instrucción ADDSP R1, dir se necesitan los siguientes bits:
- 7 bits para el código de operación, puesto que con 7 bits se puede identificar hasta 127 instrucciones.
 - 5 bits para el registro puesto que con 5 bits se puede identificar a los 32 registros.
 - Las direcciones de memoria en ese computador ocupan 32 bits.

Por tanto, para poder representar de forma completa esta instrucción se necesitan $5 + 7 + 32$ bits, es decir, dos palabras. El formato será el siguiente:

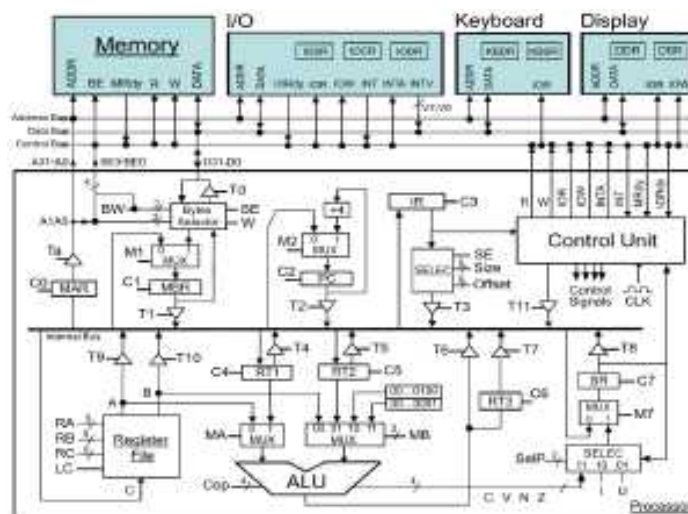


- b) Las operaciones elementales y señales serían:

PC → MAR	T4, C1 // fetch
Mem[MAR] → MBR, PC + 4 → PC	Td, L, C2, C4
MBR → RI	T3, C6
Decodificación	
PC → MAR	T4, C1 // lectura de la segunda palabra
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, CodOP=+, T5, C3
R31 → MAR	RA=R31, T1, C1
MBR → Mem[MAR]	E, TA

- c) Para la ejecución de la instrucción anterior se necesitan 12 ciclos. Por tanto, el número medio de ciclo por instrucción en este computador = $0,8 * 7 + 0,2 * 12 = 8$ ciclos en media por instrucción.
 2 GHz supone 2×10^9 ciclos por segundo, a 8 ciclos por instrucción en media se podrían ejecutar $(2 \times 10^9) / 8 = (2000 \times 10^6) / 8 = 250 \text{ MIPS}$.

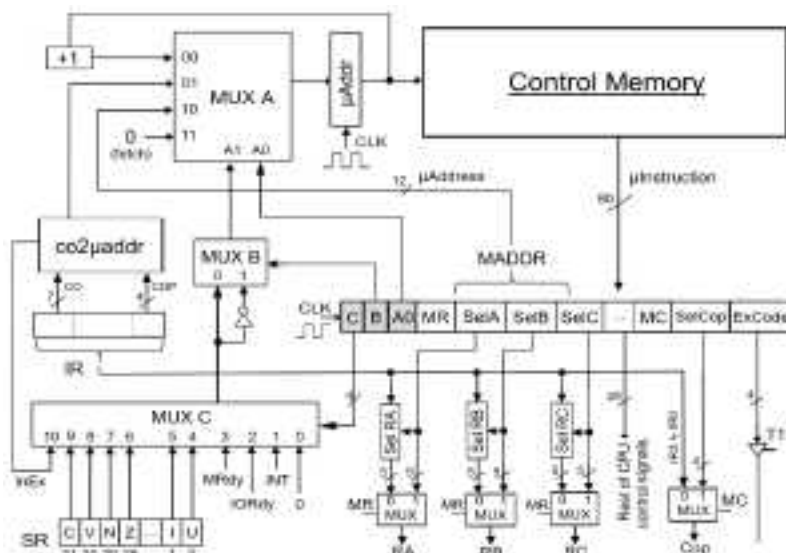
Ejercicio 15. Dado el procesador con la siguiente estructura:



Y con las siguientes características:

- Un banco de registros con 32 registros de 32 bits, capaz de procesar 127 instrucciones distintas.
- Una ALU con 16 operaciones aritmético-lógicas que incluyen la suma, resta, multiplicación, los and, or, not, xor a nivel de bit, la rotación a la derecha e izquierda de un bit, y los desplazamientos lógicos y aritméticos a la derecha, y el desplazamiento lógico a la izquierda.
- Memoria direccionable a nivel de byte y un solo ciclo de reloj para la lectura o escritura.

Este procesador dispone de una Unidad de control representada por la siguiente figura:



Responda de forma breve y justificada a las siguientes preguntas:

- Analizando el diagrama de la Unidad de Control, ¿Qué técnica de control emplea? ¿Qué tipo de secuenciamiento usa?
- Considere la instrucción set.v R1, R2, dirección. Esta instrucción es una instrucción vectorial que almacena el valor contenido en el registro R1 en las N direcciones de memoria consecutivas a partir de dirección. El valor N se encuentra almacenado en el registro R2. Es decir, almacena el valor que hay en R1 en las posiciones de memoria: dirección, dirección + 4, dirección + 8 dirección + Nx4. Especifique adecuadamente el formato de esta instrucción.
- Especifique las operaciones elementales y todas las señales de control necesarias, tanto de la unidad de control como del procesador, para ejecutar la instrucción máquina anteriormente descrita. Incluya el ciclo de fetch.

Asuma que R29 actúa como puntero de pila, que el puntero de pila apunta a cima de pila y que la pila crece hacia direcciones decrecientes de memoria.

Solución:

- Esta Unidad de Control tiene una técnica de **control microprogramado** y un **secuenciamiento implícito**.
- Un posible formato para la instrucción descrita sería:

Primera palabra:

Código de instrucción (7 bits)	Registro (5 bits)	Registro (5 bits)	(15 bits)
--------------------------------	-------------------	-------------------	-----------

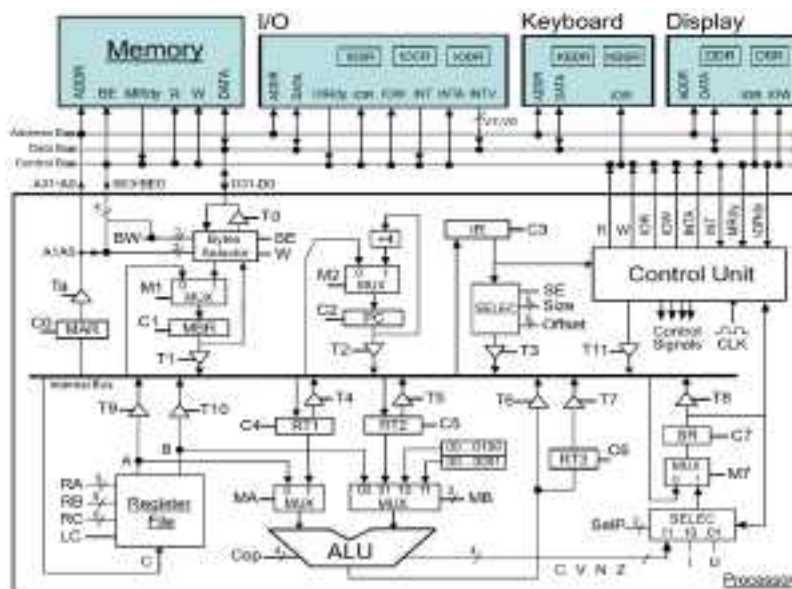
Segunda palabra:

Valor (32 bits)

c) Las operaciones elementales y las señales de control:

Ciclo	Operaciones elementales	Señales de control
Fetch		
C1	$MAR \leftarrow PC$	T2, C0
C2	$MBR \leftarrow Mem[MAR]$	TA, R, BW=11, C1, M1
C3	$PC \leftarrow PC+4, IR \leftarrow MBR$	M2, C2, T1, C3
C4	Salto a c.o.	A0, B=0, C=0
Ejecución de la instrucción		
C5	$MAR \leftarrow PC$	T2, C0
C6	$MBR \leftarrow MP[MAR], PC \leftarrow PC + 4$	TA, R, BW=11, C1, M2, C2, M1
C7	$RT1 \leftarrow R2$	C4, T9, SelA=R2_offset, MR=0
C8	$RT3 \leftarrow RT1 * 1$	MA, MB=11, SelCOP=*, SelP=11, M7, C7
C9	If (SR.Z==1) goto fetch	C=6, B=0, A0=0, MADDDR=0
C10	$RT2 \leftarrow MAR \leftarrow MBR$	T1, C0, C5
C11	$MBR \leftarrow R1$	T9, SelA=R1_offset, MR=0, M1=0, C1
C12	$MP[MAR] \leftarrow MBR$	TA, TD, BW=11, W
C13	$RT1 \leftarrow RT2$	T5, C4
C14	$RT2 \leftarrow MAR \leftarrow RT1 + 4$	MA, MB=10, SelCOP=+, T6, C0, C5
C15	$RT1 \leftarrow RT3$	T7, C4
C16	$RT3 \leftarrow RT1 - 1$	MA, MB=11, SelCOP=--, C6, SelP=11, M7, C7
C17	If (NOT SR.Z==0) goto C12	C=6, B=0, A0=0, MADDDR=C12
C18	Salto a fetch	A0, C=0, B

Ejercicio 16. Dado el procesador con la siguiente estructura:

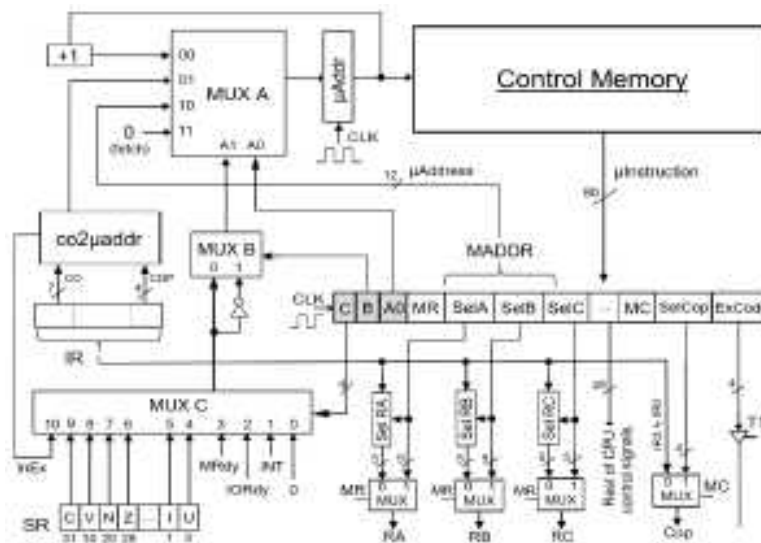


Y con las siguientes características:

- Un banco de registros con 32 registros de 32 bits, capaz de procesar 127 instrucciones distintas.
- Una ALU con 16 operaciones aritmético-lógicas que incluyen la suma, resta, multiplicación, los and, or, not, xor a nivel de bit, la rotación a la derecha e izquierda de un bit, y los desplazamientos lógicos y aritméticos a la derecha, y el desplazamiento lógico a la izquierda.
- Memoria direccionable a nivel de byte y un solo ciclo de reloj para la lectura o escritura.

Este procesador dispone de una Unidad de control representada por la siguiente figura:





Responda de forma breve y justificada a las siguientes preguntas:

- ¿Qué ventaja tiene la técnica de control microprogramado frente a la técnica cableada?
¿Qué ventaja supone el secuenciamiento implícito frente al explícito?
- Considere la instrucción **reset_v R1, dirección**. Esta instrucción es una instrucción vectorial que almacena el valor cero en las N direcciones de memoria consecutivas a partir de *dirección*. El valor N se encuentra en el registro R1. Es decir, almacena el valor cero en las posiciones de memoria:
 $\text{dirección}, \text{dirección} + 4, \text{dirección} + 8 \dots \text{dirección} + (N-1) \times 4$.
Especifique adecuadamente el formato de esta instrucción.
- Especifique las operaciones elementales y todas las señales de control necesarias, tanto de la unidad de control como del procesador, para ejecutar la instrucción máquina anteriormente descrita. Incluya el ciclo de *fetch*.

Asuma que R29 actúa como puntero de pila, que el puntero de pila apunta a cima de pila y que la pila crece hacia direcciones decrecientes de memoria. Asuma, de igual forma, que el registro R0 siempre vale cero, y que el registro de estado no precisa ser guardado, la instrucción puede modificarlo en la forma que estime oportuno.

Solución:

- El control microprogramado permite poder reprogramar el microcódigo (si se usa una ROM re-escribible). El secuenciamiento implícito permite disminuir el número de bits en la memoria de control puesto que la mayoría de las veces la siguiente microinstrucción a ejecutar es la siguiente en la memoria de control, y en caso de salto se puede solapar la microdirección con otros campos de control.
- Un posible formato para la instrucción descrita sería:

Primera palabra:

Código de instrucción (7 bits)	Registro (5 bits)	(20 bits)
--------------------------------	-------------------	-----------

Segunda palabra:

Dirección (32 bits)

- Las operaciones elementales y las señales de control:

Ciclo	Operaciones elementales	Señales de control
Fetch		
C1	$MAR \leftarrow PC$	T2, C0
C2	$MBR \leftarrow \text{Mem}[MAR]$	TA, R, BW=11, C1, M1
C3	$PC \leftarrow PC+4, IR \leftarrow MBR$	M2, C2, T1, C3
C4	Salto a c.o.	A0, B=0, C=0

Ejecución de la instrucción		
C5	MAR \leftarrow PC	T2, C0
C6	MBR \leftarrow Mem[MAR], PC \leftarrow PC + 4	TA, R, BW=11, M1, C1, M2, C2
C7	RT2 \leftarrow MBR	T1, C5
C8	RT1 \leftarrow R1 * 4	SelA=Offset_R1, MB=10, SelCop=*, MC, T6, C4
C9	MBR \leftarrow R0	SelA=0, MR, T9, M1=0, C1
C10	RT1 \leftarrow RT1 - 4, SR \leftarrow 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 \leftarrow RT2 + RT1	MA, MB=01, SelCop=+, MC, T6, C0
C13	Mem[MAR] \leftarrow MBR	TA, TD, W, BW=11,
C14	Salto C10	C=0, B=1, A0=0, MADDR=C10

Ejercicio 17. Se dispone de un computador con una frecuencia de reloj de 2 GHz. El número medio de ciclos por instrucción máquina es de 20. A este computador se le conecta un dispositivo que genera 100.000 interrupciones por segundo. La rutina de tratamiento de la interrupción ejecuta 500 instrucciones. Se pide:

- Calcule el porcentaje de tiempo que este computador dedica al tratamiento de este dispositivo.
- ¿Podría este computador atender a todas las interrupciones generadas por este dispositivo, si la rutina de tratamiento de la interrupción tuviera 2000 instrucciones máquina? Razone su respuesta.

Solución:

- El computador dispone de una frecuencia de reloj de 2 GHz = 2×10^9 ciclos/segundo.
La rutina de tratamiento ejecuta 500 instrucciones máquina, lo que suponen: $500 \times 20 = 10000 = 10^4$ ciclos de reloj consumidos por cada interrupción generada. Como el dispositivo genera 100000 interrupciones por segundo, el número de ciclos por segundo dedicados a tratar a este dispositivo es $10^5 \times 10^4 = 10^9$ ciclos por segundo.

Como la frecuencia de reloj es 2×10^9 ciclos/segundo, el porcentaje de tiempo dedicado a tratar la rutina sería:

$$10^9 / (2 \times 10^9) \times 100 = 50\%.$$

- Si la rutina tuviera 2000 instrucciones máquina, requeriría $2000 \times 20 = 4 \times 10^4$ ciclos de reloj. Si se generan 100000 interrupciones por segundo, serían necesarios $10^5 \times 4 \times 10^4 = 4 \times 10^9$ ciclos. Como el computador solo dispone de 2×10^9 ciclos/segundo, no tiene capacidad, por tanto, para poder atender todas las interrupciones que genera este dispositivo.