

**UNIVERSIDAD DE GRANADA.**

**ESCUELA TECNICA SUPERIOR DE  
INGENIERIAS INFORMATICA Y DE  
TELECOMUNICACIÓN.**



**Departamento de Arquitectura y  
Tecnología de Computadores.**

**TECNOLOGÍA Y ORGANIZACIÓN DE  
COMPUTADORES.**

**TEMA 2. UNIDADES FUNCIONALES  
GUÍA DE AYUDA PARA EL APRENDIZAJE  
AUTÓNOMO**

**1º GRADO EN INGENIERÍA INFORMÁTICA.**



## TEMA 2º. UNIDADES FUNCIONALES.

### GUÍA DE AYUDA PARA EL APRENDIZAJE AUTÓNOMO.

#### 2.1.- PARTE TEÓRICA: El estudiante deberá:

2.2.1.- Descargar de la plataforma docente y leer detenidamente el material del Tema 2º ubicado en el fichero 02.-TEMA\_2\_TOC\_UNIDADES\_FUNCIONALES.PDF.

2.1.2.- Visualizar las grabaciones de clase de la asignatura:

- Tema 2. Clase de Teoría 1.

<https://drive.google.com/file/d/1YmzLb1CefVkvHhIPx7X13zBWfbKerXSc/view?usp=sharing>

- Tema 2. Clase de Teoría 2.

[https://drive.google.com/file/d/1jGrM5h1FLyT1s2LMtS2k\\_Pt5FAylEWR/view?usp=sharing](https://drive.google.com/file/d/1jGrM5h1FLyT1s2LMtS2k_Pt5FAylEWR/view?usp=sharing)

- Tema 2. Clase de Problemas.

<https://drive.google.com/file/d/1bOTYsNsNuV9KVHwkt9gwUNYmW9qnM3ST/view?usp=sharing>

2.2.3.- Visualizar las video-clases de la asignatura “Fundamentos de Informática” desarrolladas por el profesor D. Alberto Prieto Espinosa ubicadas en el enlace [http://atc.ugr.es/APrieto\\_videoclases](http://atc.ugr.es/APrieto_videoclases) correspondientes a las lecciones:

- L3.1 Elementos constitutivos de un procesador:  
<https://www.youtube.com/watch?v=xuUC3MzTayl&feature=youtu.be>
- L3.2 Funcionamiento del procesador: ejecución de instrucciones.  
<https://www.youtube.com/watch?v=GgjFgPWGIUM&feature=youtu.be>
- L3.3 Implementación de la Unidad de Control. Procesadores integrados.  
<https://www.youtube.com/watch?v=ITeFrWEquq8&feature=youtu.be>
- L3.8 Organización y jerarquía de memoria.  
<https://www.youtube.com/watch?v=m3ykDMyjhlk&feature=youtu.be>
- L3.9 Memoria interna.  
<https://www.youtube.com/watch?v=UD4oKg-QJLI&feature=youtu.be>
- L3.10 Memoria externa.  
<https://www.youtube.com/watch?v=fPeF8nGwFg0&feature=youtu.be>
- L3.11 Conectando todo: Estructura de un PC.  
<https://www.youtube.com/watch?v=n3SPpRD5NiQ&feature=youtu.be>

2.1.4.- Se recomienda el estudio en el libro [PRI06]: Prieto, A., Lloris, A., Torres, J. C.. Introducción a la Informática, 4ª Edición, McGraw-Hill, 2006. Capítulo 7, Capítulo 10.1 y 10.2 y Capítulo 12.4 y 12.6.

#### 2.2.- PARTE DE EJERCICIOS:

1. Un procesador dispone, entre otros, de los siguientes elementos: registro de dirección (AR) de 32 bits, registro de datos (DR) de 16 bits y contador de programa (PC). Indicar:
  - a. Número de bits del bus de datos
  - b. Número de bits del bus de direcciones.
  - c. Tamaño máximo posible de la memoria principal (en MB o GB)
  - d. Tamaño en bits del registro contador de programa PC

Resolución:

- a. Número de bits del bus de datos. La longitud o anchura o número de hilos del bus de datos (DB) ha de estar en consonancia con el Registro de Datos (DR). Como éste es de longitud 16 bits, la longitud o anchura del bus de datos (DB) ha de ser también de 16 bits.
- b. Número de bits del bus de direcciones. La longitud o anchura o número de hilos del bus de direcciones (AB) ha de estar en consonancia con el Registro de Direcciones (AR). Como éste es de longitud 32 bits, la longitud o anchura del bus de direcciones (AB) ha de ser también de 32 bits.
- c. Tamaño máximo posible de la memoria principal (en MB o GB). Como la anchura del bus de direcciones es de 32 bits, esto significa que puede direccionar hasta  $2^{32}$  palabras o direcciones

de memoria. Como la anchura del bus de datos es de 16 bits, esto significa que la longitud de palabra con la que puede trabajar el procesador es de 16 bits (2 Bytes) por palabra. Por tanto, este procesador podría direccionar hasta:

$2^{32}$  palabras de memoria de 16 bits por palabra =

$2^{32}$  palabras de memoria de 2 Bytes por palabra =

$2^{33}$  Bytes =  $2^3 * 2^{30}$  Bytes = 8 GBytes

- d. **Tamaño en bits del registro contador de programa PC.** El registro Contador de Programa contiene la DIRECCIÓN donde se ubica la siguiente instrucción que va a realizar el procesador. Por tanto, como contiene una dirección, debe tener la misma longitud o anchura que el Registro de Direcciones (AR) y el Bus de Direcciones (AB), es decir, 32 bits.
2. Un procesador dispone (entre otros) de los registros: **PC** (Contador de Programa), **AR** (Registro de Dirección de 12 bits), **DR** (Registro de Datos de 16 bits), **IR** (Registro de Instrucciones) y registros auxiliares para datos **R5** y **R7**. El procesador está conectado con la memoria principal. Suponiendo que el procesador está iniciando la captación de una instrucción con el contador de programa **PC = FFC** (en hexadecimal), y que el contenido inicial de la memoria principal es el de la tabla adjunta, donde tanto direcciones como datos están representados en hexadecimal, responda a las siguientes cuestiones:

Dirección (hexadecimal)	Contenido (hexadecimal)
000	7FFB
001	ABD1
002	6C25
.	.
.	.
FFB	3437
FFC	A5C1
FFD	3E26
FFE	AA32
FFF	35C6

- a) Indique el contenido de los registros **PC** e **IR** al finalizar la **fase de captación** de la instrucción.
- 1) El Contador de Programa (PC) se ha de incrementar al final de la fase de captación de la instrucción. Por tanto: **PC ← PC + 1 = FFD**
  - 2) El Registro de Instrucción (IR) debe contener el Código de Operación de la instrucción que se está ejecutando. Por tanto: **IR = A5C1**
- b) Sabiendo que el código de operación que está en **IR** corresponde a una instrucción **ST R5, 001** y que ésta instrucción consiste en almacenar el contenido del registro **R5** en la dirección de memoria **001**, y que en **R5** se tiene el dato **A000**, indique los datos que cambian en la memoria y sus correspondientes direcciones, al finalizar la fase de ejecución de la instrucción.
- Cambiará, por tanto, el contenido de la posición de memoria 001 grabándose en ella el valor que tiene el registro R5 (A000). Por tanto: **M(001) = A000**
- c) Indique el número de hilos de los buses de datos y de direcciones.
- Como las direcciones están expresados con 3 dígitos hexadecimales, esto significa que tienen una longitud de  $3 * 4 = 12$  bits. Por tanto: Número de hilos de **AddressBus = 12 hilos**
- Como los datos están expresados con 4 dígitos hexadecimales, esto significa que tienen una longitud de  $4 * 4 = 16$  bits. Por tanto: Número de hilos de **DataBus = 16 hilos**
- d) Indique el tamaño máximo (en Bytes) de la memoria principal.
- $2^{12}$  palabras de 16 bits = 4096 palabras de 2 Bytes = 8192 Bytes = 8 KBytes.**

3. Suponga que un procesador dispusiese de una instrucción cargar, **LOAD R7 DIR**, que almacena en el registro R7 del procesador el contenido de la posición de memoria indicada en la dirección DIR ( $R7 \leftarrow M(DIR)$ ). La instrucción tiene de código (en hexadecimal) **0700**. Suponiendo que esta instrucción se encuentra en la posición (en hexadecimal) **0039** de la memoria, que en DIR se encuentra la dirección (en hexadecimal) **DIR = 54C2** y que en dicha posición de memoria,  $M(54C2)$ , se encuentra el dato (en hexadecimal) **D7A2**, realice una tabla donde se indiquen las distintas microoperaciones que deben generarse durante la ejecución de la instrucción **LOAD R7 54C2**, así como los valores que tienen en cada momento los registros PC, AR, DR, IR y R7.

Fase	Microoperación	Contenidos de los registros				
		PC	IR	AR	DR	R7
Valores iniciales		0039	XXXX	XXXX	XXXX	XXXX
Captación de instrucción	AR ← PC	0039	XXXX	0039	XXXX	XXXX
	DR ← M(AR)	0039	XXXX	0039	0700	XXXX
	IR ← DR	0039	0700	0039	0700	XXXX
	PC ← PC+1	003A	0700	0039	0700	XXXX
Ejecución de instrucción	AR ← 54C2	003A	0700	54C2	0700	XXXX
	DR ← M(AR)	003A	0700	54C2	D7A2	XXXX
	R7 ← DR	003A	0700	54C2	D7A2	D7A2

4. Suponga que un procesador dispusiese de una instrucción de memorizar, **STORE R1 DIR** que almacena en la posición DIR de memoria el contenido del registro R1 ( $M(DIR) \leftarrow R1$ ). La instrucción tiene de código (en hexadecimal) **1100**. Suponiendo que ésta instrucción se encuentra en la posición (en hexadecimal) **A777** de la memoria, que la dirección de almacenamiento en memoria (en hexadecimal) es **DIR = 5ACD** y que R1 contiene el dato (en hexadecimal) **FFFF**, realice una tabla donde se indiquen las distintas microoperaciones que deben generarse durante la ejecución de la instrucción **STORE R1 5ACD**, los valores que tienen en cada momento los registros PC, AR, DR e IR, y los cambios producidos en la memoria.

Fase	Microoperación	Contenidos de los registros					
		PC	IR	AR	DR	R1	M(5ACD)
Valores iniciales		A777	XXXX	XXXX	XXXX	FFFF	XXXX
Captación de instrucción	AR ← PC	A777	XXXX	A777	XXXX	FFFF	XXXX
	DR ← M(AR)	A777	XXXX	A777	1100	FFFF	XXXX
	IR ← DR	A777	1100	A777	1100	FFFF	XXXX
	PC ←PC+1	A778	1100	A777	1100	FFFF	XXXX
Ejecución de instrucción	AR ← 5ACD	A778	1100	5ACD	1100	FFFF	XXXX
	DR ← R1	A778	1100	5ACD	FFFF	FFFF	XXXX
	M(AR)← DR	A778	1100	5ACD	FFFF	FFFF	FFFF

5. Suponga que un procesador dispusiese de una instrucción saltar, *JUMP DIR*, que salta a la posición de memoria indicada por la dirección *DIR*, ejecutándose entonces la instrucción almacenada en dicha posición de memoria ( $M(DIR)$ ). La instrucción tiene de código (en hexadecimal) 1200. Suponiendo que esta instrucción se encuentra en la posición (en hexadecimal) 7777 de la memoria y que en *DIR* se encuentra (en hexadecimal) el valor  $DIR = 7ACD$ , realice una tabla donde se indiquen las distintas microoperaciones que deben generarse durante la ejecución de la instrucción *JUMP 7ACD*, así como los valores que tienen en cada momento los registros *PC*, *AR*, *DR*, e *IR*.

Fase	Microoperación	Contenidos de los registros				
		PC	IR	AR	DR	
Valores iniciales		7777	XXXX	XXXX	XXXX	
Captación de instrucción	AR ← PC	7777	XXXX	7777	XXXX	
	DR ← M(AR)	7777	XXXX	7777	1200	
	IR ← DR	7777	1200	7777	1200	
	PC ← PC+1	7778	1200	7777	1200	
Ejecución de instrucción	PC ← 7ACD	7ACD	1200	7777	1200	

¿Qué ocurriría en el procesador si la dirección de salto fuese  $DIR = 7777$  que es la dirección de memoria donde se encuentra ubicada la propia instrucción que se está ejecutando?.

La secuencia de microoperaciones en ese caso sería:

Fase	Microoperación	Contenidos de los registros				
		PC	IR	AR	DR	
Valores iniciales		7777	XXXX	XXXX	XXXX	
Captación de instrucción	AR ← PC	7777	XXXX	7777	XXXX	
	DR ← M(AR)	7777	XXXX	7777	1200	
	IR ← DR	7777	1200	7777	1200	
	PC ← PC+1	7778	1200	7777	1200	
Ejecución de instrucción	PC ← 7777	7777	1200	7777	1200	

El contador de programa (*PC*) quedaría con el mismo valor que tenía al principio de la ejecución de la instrucción y, por tanto, el procesador volvería a ejecutar la misma instrucción indefinidamente en el tiempo, quedando “atrapado” en dicha instrucción.

6. Suponga que un procesador dispusiese de una instrucción de sumar, *ADD R1 DIR1, DIR2* que almacena en el registro *R1* del procesador la suma de los contenidos de las posiciones de memoria *DIR1* y *DIR2* ( $R1 \leftarrow M(DIR1) + M(DIR2)$ ). La instrucción tiene de código (en hexadecimal) 7707. Suponiendo que ésta instrucción se encuentra en la posición (en hexadecimal) B700 de la memoria, que la dirección (en hexadecimal) de los datos en memoria fuesen  $DIR1 = 5A00$  y  $DIR2 = 5A01$ , que los contenidos (en hexadecimal) de dichas posiciones de memoria son  $M(5A00) = 3CD2$  y  $M(5A01) = 4F56$  realice una tabla donde se indiquen las distintas microoperaciones que deben generarse durante la ejecución de la instrucción *ADD R1 5A00 5A01* y los valores que tienen en cada momento los registros *PC*, *AR*, *DR* e *IR* (nota:  $3CD2 + 4F56 = 8C28$ ). Para realizar la instrucción, suponga que el registro *R1* va acumulando progresivamente los resultados de la suma.

Fase	Microoperación	Contenidos de los registros				
		PC	IR	AR	DR	R1
Valores iniciales		B700	XXXX	XXXX	XXXX	XXXX
Captación de instrucción	AR ← PC	B700	XXXX	B700	XXXX	XXXX
	DR ← M(AR)	B700	XXXX	B700	7707	XXXX
	IR ← DR	B700	7707	B700	7707	XXXX
	PC ← PC+1	B701	7707	B700	7707	XXXX
Ejecución de instrucción	R1 ← 0000	B701	7707	B700	7707	0000
	AR ← 5A00	B701	7707	5A00	7707	0000
	DR ← M(AR)	B701	7707	5A00	3CD2	0000
	R1 ← R1 + DR	B701	7707	5A00	3CD2	3CD2
	AR ← 5A01	B701	7707	5A01	3CD2	3CD2
	DR ← M(AR)	B701	7707	5A01	4F56	3CD2
	R1 ← R1 + DR	B701	7707	5A01	4F56	8C28

7. Considere un procesador que trabaja con datos de 2 Bytes y que funciona a una frecuencia de 1 GHz. Suponiendo que las transferencias en los buses no ralentizan el funcionamiento del procesador (es decir, en un ciclo de reloj se hace una transferencia de información), hacer una estimación del ancho de banda (MB/s) en los buses externos de dirección y de datos.

Dado que la frecuencia del reloj del procesador es  $f = 1 \text{ GHz} = 1 \cdot 10^9 \text{ Hz}$ , el periodo del reloj o tiempo de ciclo (TC) del procesador será  $T = 1/f = 10^{-9} \text{ s} = 1 \text{ ns}$ .

Como dice el enunciado que “en un ciclo de reloj se hace una transferencia de información”, eso significa que en un ciclo de reloj ( $10^{-9} \text{ s}$ ) se transfiere una unidad de información o palabra de memoria (2 Bytes) entre el procesador y memoria o viceversa

Además, se indica que el bus “no ralentiza el funcionamiento del procesador”. En la realidad se tarda un cierto tiempo (no nulo) en que viaje el dato de procesador a memoria o viceversa. Suponiendo que este tiempo (denominado tiempo de latencia del bus) es  $t = 0 \text{ s}$ , un dato de 2 Bytes tardaría en transferirse de procesador a memoria un ciclo de reloj, es decir,  $10^{-9} \text{ s}$ . Por tanto, por una simple regla de tres:

Si en  $10^{-9} \text{ s}$  se transfieren 2 Bytes,

En 1 s se transferirán X Bytes

Por tanto  $X = 2/10^{-9} \text{ Bytes/s} = 2 \cdot 10^9 \text{ Bytes/s} = 1.907,25 \text{ Mbytes/s}$

8. Suponga un procesador que trabaja con datos de 16 bits y que se necesitan 3 ciclos de reloj para transferir un dato y que la frecuencia de reloj es de 8 MHz. ¿Qué velocidad de transferencia de datos se obtiene?.

Dado que la frecuencia del reloj del procesador es  $f = 8 \text{ MHz} = 8 \cdot 10^6 \text{ Hz}$ , el periodo del reloj o tiempo de ciclo (TC) del procesador será  $T = 1/f = 1/(8 \cdot 10^6) \text{ s} = 0,125 \cdot 10^{-6} \text{ s} = 0,125 \mu\text{s}$ .

Como dice el enunciado que “se necesitan 3 ciclos de reloj para transferir un dato (de 16 bits = 2 Bytes)”, eso significa que se necesitan TRES ciclo de reloj ( $3 \cdot 0,125 \cdot 10^{-6} \text{ s} = 0,375 \cdot 10^{-6} \text{ s}$ ) para transferir un dato de 2 Bytes de procesador a memoria o viceversa.

Por tanto, por una simple regla de tres:

Si en  $0,375 \cdot 10^{-6} \text{ s}$  se transfieren 2 Bytes,

En 1 s se transferirán X Bytes

Por tanto  $X = 2/(0,375 \cdot 10^{-6}) \text{ Bytes/s} = 5,333 \cdot 10^6 \text{ Bytes/s} = 5,086 \text{ Mbytes/s}$



9. Un procesador que trabaja con una frecuencia de reloj de **450 MHz**, se diseña de la siguiente forma:
- En la **fase de captación** todas sus instrucciones consumen **2 ciclos de reloj**.
  - En la **fase de ejecución** todas las instrucciones, en general, consumen **2 ciclos de reloj**, salvo las instrucciones que son de almacenamiento en memoria que consumen **4 ciclos de reloj**.

Se ejecuta en el procesador un programa que tiene **200 instrucciones** (de las que **150** son instrucciones de carácter general y **50** de ellas son instrucciones de almacenamiento en memoria). Indique:

- a) Número de ciclos totales de reloj que consume la realización de ese programa.
- b) Tiempo que se tarda en realizar este programa en el procesador.
- c) Velocidad de procesamiento en **MIPS** (Millones de Instrucciones Por Segundo) que se mide de acuerdo a la realización de este programa.

- a) Número de ciclos totales de reloj que consume la realización de ese programa.  
Como en el programa hay 200 instrucciones de las que 150 son de carácter general (2 ciclos de captación + 2 ciclos de ejecución = 4 ciclos en total) y 50 son de almacenamiento en memoria (2 ciclos de captación + 4 ciclos de ejecución = 6 ciclos en total) entonces, el número total de ciclos que se realizan en el programa serían:

$$(150 \text{ instrucciones a 4 ciclos por instrucción}) + (50 \text{ instrucciones a 6 ciclos por instrucción}) = 600 \text{ ciclos} + 300 \text{ ciclos} = 900 \text{ ciclos de reloj} = N.C.$$

- b) Tiempo que se tarda en realizar este programa en el procesador.  
Como la frecuencia de reloj es de  $f = 450 \cdot 10^6 \text{ Hz}$ , entonces, el periodo de reloj  $T = T.C. = 2,2 \cdot 10^{-9} \text{ s}$ , por tanto, el tiempo de ejecución será  $T.E. = N.C. \cdot T.C. = 900 \cdot 2,2 \cdot 10^{-9} \text{ s} = 2 \cdot 10^{-6} \text{ s} = 2 \mu\text{s}$
- c) Velocidad de procesamiento en MIPS (Millones de Instrucciones Por Segundo) que se mide de acuerdo a la realización de este programa.

Como en el programa hay 200 instrucciones que se realizan en un tiempo de  $2 \cdot 10^{-6} \text{ s}$ , por una simple regla de tres se tiene que:

Si en  $2 \cdot 10^{-6} \text{ s}$  se realizan 200 Instrucciones,

En 1 s se realizarán X instrucciones

$$\text{Por tanto } X = 200 / (2 \cdot 10^{-6}) \text{ Instrucciones/s} = 100.000.000 \text{ instrucciones/s} = 100 \text{ MIPS}$$

10. Se tienen tres procesadores (A, B, C) que trabajan con las frecuencias de reloj que se indican en la tabla. Se realiza en cada uno de ellos un programa benchmark de prueba de 5 millones de instrucciones. En la tabla se indica el número de ciclos de reloj que requiere la realización de cada instrucción (fase de captación + fase de ejecución).

- a. Indique el **tiempo de realización** del programa **benchmark**.
- b. Indique las prestaciones en **MIPS** (Millones de Instrucciones Por Segundo) de cada procesador.

	Frecuencia reloj (GHz)	Tiempo Ciclo (s)	Número de ciclos de reloj por instrucción.	Tiempo ejecución Benchmark (segundos)	MIPS
A	1,5		4		
B	2,0		8		
C	3,0		10		

- a. Indique el **tiempo de realización** del programa **benchmark**.  
Hay que calcular los periodos de reloj (T) o tiempo de ciclo (T.C.) de cada procesador  
 $f_A = 1,5 \cdot 10^9 \text{ Hz}$ , entonces, el periodo de reloj  $T_A = T.C. = 6,66 \cdot 10^{-10} \text{ s}$   
 $f_B = 2,0 \cdot 10^9 \text{ Hz}$ , entonces, el periodo de reloj  $T_B = T.C. = 5,00 \cdot 10^{-10} \text{ s}$   
 $f_C = 3,0 \cdot 10^9 \text{ Hz}$ , entonces, el periodo de reloj  $T_C = T.C. = 3,33 \cdot 10^{-10} \text{ s}$



Como en cada programa se realizan 5 millones de instrucciones, el Tiempo de Ejecución de cada programa se calculará multiplicando el número de instrucciones del mismo (5.000.000) por el Número de Ciclos (NC) de cada instrucción por el Tiempo de Ciclo TC de cada procesador.

$$TE_A = 5.000.000 \text{ Instrucciones} * 4 \text{ Ciclos/instrucción} * 6,66 * 10^{-10} \text{ s} = 0,0132 \text{ s}$$

$$TE_B = 5.000.000 \text{ Instrucciones} * 8 \text{ Ciclos/instrucción} * 5,00 * 10^{-10} \text{ s} = 0,0200 \text{ s}$$

$$TE_C = 5.000.000 \text{ Instrucciones} * 10 \text{ Ciclos/instrucción} * 3,33 * 10^{-10} \text{ s} = 0,0165 \text{ s}$$

- b. Indique las prestaciones en **MIPS** (Millones de Instrucciones Por Segundo) de cada procesador. Como en el programa hay 5.000.000 de instrucciones que se realizan en un los tiempos indicados anteriormente para cada procesador, por una simple regla de tres se tiene que, para cada procesador:

b.1) Procesador A:

Si en 0,0132 s se realizan 5.000.000 Instrucciones,

En 1 s se realizarán  $X_A$  instrucciones

$$\text{Por tanto } X_A = 5.000.000 / 0,0132 \text{ Instrucciones/s} = 378.787.878 \text{ instrucciones/s} = 378,78 \text{ MIPS}$$

b.2) Procesador B:

Si en 0,0200 s se realizan 5.000.000 Instrucciones,

En 1 s se realizarán  $X_B$  instrucciones

$$\text{Por tanto } X_B = 5.000.000 / 0,0200 \text{ Instrucciones/s} = 250.000.000 \text{ instrucciones/s} = 250,00 \text{ MIPS}$$

b.3) Procesador C:

Si en 0,0165 s se realizan 5.000.000 Instrucciones,

En 1 s se realizarán  $X_C$  instrucciones

$$\text{Por tanto } X_C = 5.000.000 / 0,0165 \text{ Instrucciones/s} = 303.030.303 \text{ instrucciones/s} = 303,03 \text{ MIPS}$$

Resumen:

	Frecuencia reloj (GHz)	Tiempo Ciclo (s)	Número de ciclos de reloj por instrucción.	Tiempo ejecución Benchmark (segundos)	MIPS
A	1,5	$6,6 * 10^{-10}$	4	<b>0,0132</b>	<b>378,78</b>
B	2,0	$5,0 * 10^{-10}$	8	<b>0,0200</b>	<b>250,00</b>
C	3,0	$3,3 * 10^{-10}$	10	<b>0,0165</b>	<b>303,03</b>

11. En el Computador Sencillo 1 (CS1) los datos son de 8 bits, las direcciones son de 6 bits. Tiene un Repertorio de Instrucciones de 4 instrucciones (STOP, ADD, SUB, STA). La fase de captación siempre consume 2 ciclos de reloj. La fase de ejecución de cada instrucción consume STOP (1 ciclo), ADD y SUB (3 ciclos) y STA (2 ciclos).

Si la frecuencia de reloj del CS1 es de 50 MHz y se ejecuta un programa benchmark que contiene 24 instrucciones ADD, 26 instrucciones SUB, 10 instrucciones STA y una instrucción STOP. Indicar:

- Número de ciclos de reloj totales consumidos por el programa.
- Tiempo de realización del programa de bechmark.
- Prestaciones del CS1 en MIPS (Millions Instuctions Per Second).
- Tamaño de memoria máximo direccionable en Bytes.

- Número de ciclos de reloj totales consumidos por el programa.

En total en el programa hay 61 instrucciones, repartidas de la siguiente manera:

$$24 \text{ instrucciones ADD} = 24 * (2 + 3) \text{ ciclos} = 120 \text{ ciclos}$$

$$26 \text{ instrucciones SUB} = 26 * (2 + 3) \text{ ciclos} = 130 \text{ ciclos}$$

$$10 \text{ instrucciones STA} = 10 * (2 + 2) \text{ ciclos} = 40 \text{ ciclos}$$

$$1 \text{ instrucción STOP} = 1 * (2 + 1) \text{ ciclos} = 3 \text{ ciclos}$$

$$\text{Total: } 120 + 130 + 40 + 3 = 293 \text{ ciclos de reloj o ciclos máquina.}$$

b) Tiempo de realización del programa de benchmark.

Una vez calculados el número de ciclos máquina que ocupa el programa, basta con multiplicarlos por el periodo de reloj (T) o el Tiempo de Ciclo (TC) y se obtiene el Tiempo de Ejecución (TE) o la duración del programa. Para ello, hay que calcular primeramente el periodo de reloj partiendo de la frecuencia del procesador:

$$f = 50 \text{ MHz} = 50 \cdot 10^6 \text{ Hz} ; T = 1/f = 1/(50 \cdot 10^6) \text{ s} = 2 \cdot 10^{-8} \text{ s}$$

$$TE = 293 \text{ ciclos} \cdot 2 \cdot 10^{-8} \text{ s/ciclo} = 586 \cdot 10^{-8} \text{ s} = 5,86 \cdot 10^{-6} \text{ s}$$

c) Prestaciones del CS1 en MIPS (Millions Instructions Per Second).

Como en el Tiempo de Ejecución se realizan las 61 instrucciones del programa, mediante una simple regla de tres, se tiene:

Si en  $5,86 \cdot 10^{-6} \text{ s}$  se realizan 61 Instrucciones,

En 1 s se realizarán X instrucciones

$$\text{Por tanto } X = 61 / (5,86 \cdot 10^{-6}) \text{ Instrucciones/s} = 10.409.556 \text{ instrucciones/s} = 10,409556 \text{ MIPS}$$

d) Tamaño de memoria máximo direccionable en Bytes.

Como se indica en el enunciado "los datos son de 8 bits, las direcciones son de 6 bits", su Bus de Datos (DB) será de 8 bits y su bus de Direcciones (AB) será de 6 bits. Por tanto, se podrán direccionar hasta:

$$2^6 \text{ posiciones de memoria de 8 bits} =$$

$$64 \text{ posiciones de memoria de 1 Byte} =$$

$$64 \text{ Bytes}$$

**Realice los ejercicios siguientes:**

- Ejercicios de la relación de problemas del Tema 2º.
- Ejercicios 1.4 a 1.7 y 7.1 a 7.7 del libro [PRI06]: Prieto, A., Lloris, A., Torres, J. C.. Introducción a la Informática, 4ª Edición, McGraw-Hill, 2006.