

ARQUITECTURA Y SISTEMAS OPERATIVOS

Trabajo Práctico N.º 2: Introducción a la Arquitectura de Computadoras

1. Comprender el sistema binario y su aplicación en las computadoras.
2. Analizar el funcionamiento básico del ciclo de instrucción en una CPU.
3. Identificar el propósito y las características del código máquina y el lenguaje ensamblador.

Ejercicios

Ejercicio 1: Conversión entre sistemas numéricos

Instrucciones:

Convierte los siguientes números entre el sistema binario y decimal.

1. De **decimal a binario**:

- a) 15
- b) 43
- c) 100

2. De **binario a decimal**:

- a) 1010
- b) 11011
- c) 111101

1)

15	2	residuo = 1
7	7	
7	2	residuo = 1
3	3	
3	2	residuo = 1
1	3	
1	2	residuo = 1
0	0	

Resultado = 1111

2)

43	2	residuo = 1
21	21	
21	2	residuo = 1
10	10	
10	2	residuo = 0
5	5	
5	2	residuo = 1
2	2	
2	2	residuo = 0
1	1	
1	2	residuo = 1
0	0	

Resultado = 101011

3)	$100 \overline{) 2}$	residuo = 0	$3 \overline{) 2}$	residuo = 1
	$50 \overline{) 2}$	residuo = 0	$1 \overline{) 2}$	residuo = 1
	$25 \overline{) 2}$	residuo = 1		
	$12 \overline{) 2}$	residuo = 0		
	$6 \overline{) 2}$	residuo = 0		

Resultado = **1100100**

a) 1010

$$1 \times 2^3 = 8$$

$$0 \times 2^2 = 0$$

$$1 \times 2^1 = 2$$

$$0 \times 2^0 = 0$$

Resultado = $8 + 0 + 2 + 0 = 10_{10}$

b) 11011

$$1 \times 2^4 = 16$$

$$1 \times 2^3 = 8$$

$$0 \times 2^2 = 0$$

$$1 \times 2^1 = 2$$

$$1 \times 2^0 = 1$$

Resultado = $16 + 8 + 0 + 2 + 1 = 27_{10}$

c) 111101

$$\begin{aligned}
 1 \times 2^5 &= 32 \\
 1 \times 2^4 &= 16 \\
 1 \times 2^3 &= 8 \\
 1 \times 2^2 &= 4 \\
 0 \times 2^1 &= 0 \\
 1 \times 2^0 &= 1
 \end{aligned}$$

$$\text{Resultado} = 32 + 16 + 8 + 4 + 0 + 1 = 61_{10}$$

Ejercicio 2: Identificación del ciclo de instrucción

Instrucciones:

Relaciona cada etapa del ciclo de instrucción con su descripción.

Etapa	Descripción
a) Búsqueda	1. El procesador interpreta la instrucción para determinar su acción.
b) Decodificación	2. El procesador ejecuta la instrucción (movimientos, cálculos, etc.).
c) Ejecución	3. La CPU recupera la instrucción desde la memoria.

- a) Búsqueda : 3. La CPU recupera la instrucción desde la memoria.
- b) Decodificación : 1. El procesador interpreta la instrucción para determinar su acción.
- c) Ejecución : 2. El procesador ejecuta la instrucción (movimientos, cálculos, etc.)

Ejercicio 3: Código máquina y ensamblador

Instrucciones:

Lee el siguiente fragmento de ensamblador y responde las preguntas:

```
MOV AX, 05h    ; Cargar el valor 5 en el registro AX
ADD AX, 03h    ; Sumar 3 al valor de AX
HLT            ; Detener la ejecución
```

Preguntas:

1. ¿Qué valor final tendrá el registro AX al terminar el programa?
 - a) 3
 - b) 5
 - c) 8
 - d) 0
2. ¿Cuál es el propósito de la instrucción HLT?
 - a) Sumar valores.
 - b) Detener la ejecución del programa.
 - c) Guardar datos en memoria.
 - d) Reiniciar el procesador.

1) El valor de AX al terminar el programa es 8.

Carga el valor 5 en AX, luego con la instrucción `ADD AX 03h`, se suma 3 al a AX, lo que lo lleva a 8 y la instrucción `HLT` detiene la ejecución del programa.

2) Detener la ejecución del programa. La instrucción `HLT` indica que el programa ha finalizado su ejecución.