

# **CAPÍTULO I**

## **INTRODUCCIÓN**

---

## 1.1 ESTRUCTURA GENERAL DE UNA COMPUTADORA

Una computadora digital es una máquina electrónica capaz de realizar cálculos con gran rapidez, obedeciendo instrucciones muy específicas y elementales que reflejan su estructura funcional y organizacional.

Se puede definir conceptualmente a una computadora como una máquina que consta de elementos de entrada, elementos de salida, un procesador central y una memoria.

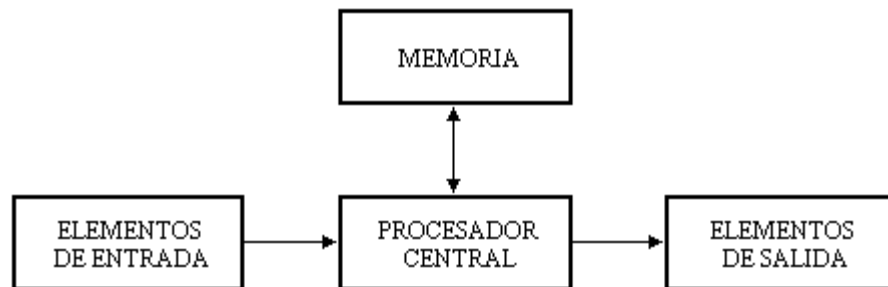


Figura 1.1. Diagrama de bloques de una computadora.

## 1.2 TIPOS DE COMPUTADORAS

Existen diversos criterios de clasificación para las arquitecturas de computadoras. Uno de ellos es la manera en como ejecutan sus cálculos, el cual origina dos categorías de computadoras:

1. Aquellas que realizan los cálculos de manera secuencial (a la cuál pertenecen la mayoría de las computadoras); y
2. Aquellas que realizan los procesos en paralelo.

### 1.2.1 COMPUTADORAS SECUENCIALES

Dentro de esta clasificación encontramos a las computadoras SISD (Single-Instruction Stream, Single-Data Stream / Flujo Único de Instrucciones, Flujo Único de Datos).

La arquitectura de von Neumann pertenece a esta clasificación que corresponde a computadoras que tienen un sólo CPU ejecutando una instrucción a la vez, además, en este tipo de computadoras

sólo se puede buscar o almacenar un elemento de datos a la vez. En las computadoras de von Neumann los programas y datos se encuentran en una memoria externa.

Para ejecutar una instrucción, la computadora tiene que efectuar las siguiente etapas:

1. Traer el código de la instrucción a ejecutar.
2. Decodificar la instrucción, es decir, saber cuáles son las micro-operaciones que tiene que realizar la computadora para ejecutar dicha instrucción.
3. Traer los operandos en caso de que los requiera la instrucción.
4. Ejecutar la instrucción y guardar el resultado.

Estas etapas se ejecutan de manera secuencial para cada instrucción, por lo tanto, una nueva instrucción no puede comenzar hasta que la anterior termine.

La figura 1.2 muestra la computadora de von Neumann.

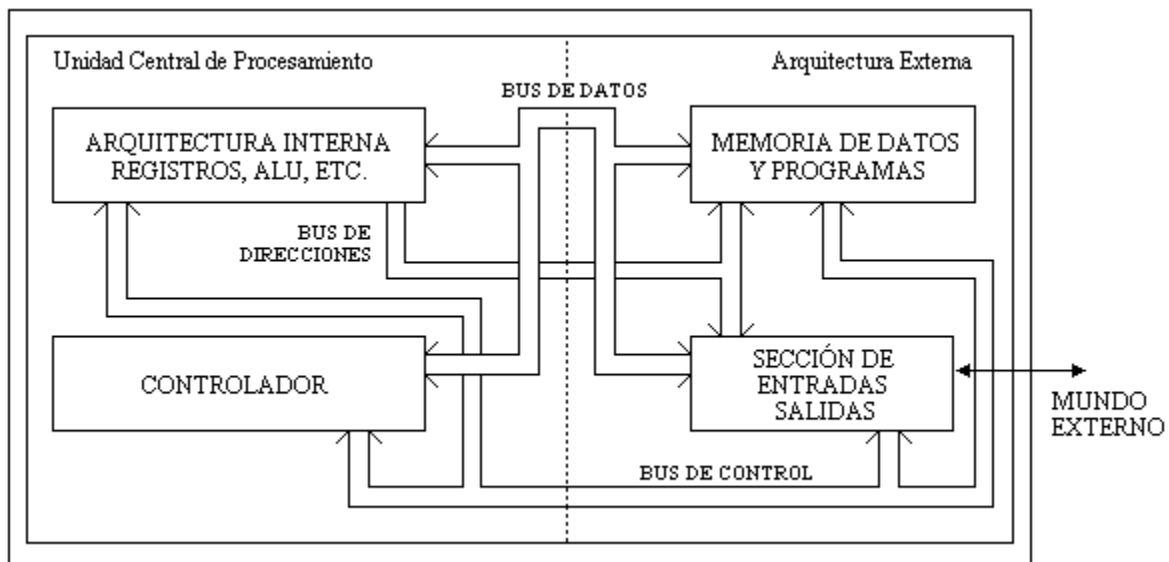


Figura 1.2. Computadora de von Neumann.

### 1.2.2 COMPUTADORAS PARALELAS

Una computadora que ejecuta procesos en paralelo cuenta con varios de sus componentes internos repetidos, de esta forma puede ejecutar varias instrucciones al mismo tiempo. Dentro de esta clasificación encontramos a las computadoras SIMD (Single-Instruction Stream, Multiple-Data Stream / Flujo Único de Instrucciones, Flujo Múltiple de Datos) y MIMD (Multiple-Instruction Stream, Multiple-Data stream / Flujo Múltiple de Instrucciones, Flujo Múltiple de Datos).

Las arquitecturas SIMD son esenciales en el mundo de las computadoras paralelas, debido a su habilidad para manejar grandes vectores y matrices de datos en tiempos muy cortos. El secreto detrás de este tipo de arquitectura es que cuentan con un varios procesadores ejecutando la misma operación sobre un conjunto de datos. Por ejemplo, cuando una simple instrucción SIMD suma 64 números, el hardware SIMD envía 64 flujos de datos a 64 ALU's (Arithmetic Logic Unit / Unidad Lógico Aritmética) para formar 64 sumas en un sólo ciclo de reloj. Incluso cuando el tamaño del vector es mayor al número de ALU's disponibles, la velocidad, comparada con una computadora secuencial, es inmensa.

Las arquitecturas MIMD, también llamadas máquinas multiprocesadores, tienen más de un procesador funcionando asíncrona e independientemente. Al mismo tiempo, los diferentes procesadores pueden estar ejecutando diferentes instrucciones sobre diferentes conjuntos de datos. Las arquitecturas MIMD se utilizan en aplicaciones de Diseño Asistido por Computadora, Graficación por Computadora, simulaciones en tiempo real, y en general, en aplicaciones que requieran gran poder de cómputo.

### 1.3 SEGMENTACIÓN ENCAUZADA

Una forma de lograr el paralelismo sin tener componentes repetidos es utilizando una metodología llamada *segmentación encauzada* (en inglés, pipeline). En la segmentación encauzada la computadora está dividida en varias etapas, de manera que cada etapa efectúa operaciones sobre instrucciones diferentes. La figura 1.3 muestra este concepto con cuatro etapas, las cuales están ligadas con registros de acoplo. La primera etapa trae la instrucción  $I_j$ ; la segunda, decodifica la instrucción  $I_{j-1}$ ; la tercera, trae los operandos de la instrucción  $I_{j-2}$ ; y por último la cuarta, ejecuta la instrucción  $I_{j-3}$ .

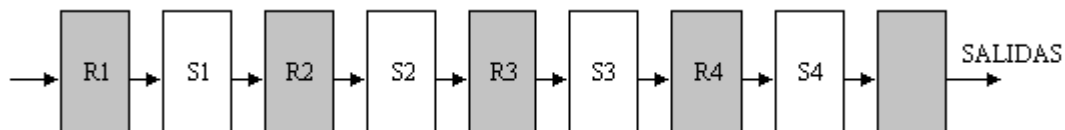


Figura 1.3. Estructura básica de un procesador segmentado.

Un procesador segmentado de  $k$  etapas es en un principio  $k$  veces más rápido que un procesador sin esta tecnología. La figura 1.4 muestra el diagrama de tiempos / tareas de un procesador segmentado de cuatro etapas. En el diagrama se observa que cada instrucción requiere 4 tiempos en ser ejecutada, sin embargo, una vez que el cauce está completo, el tiempo de salida de cada instrucción con respecto a la anterior es de periodo  $T$ .

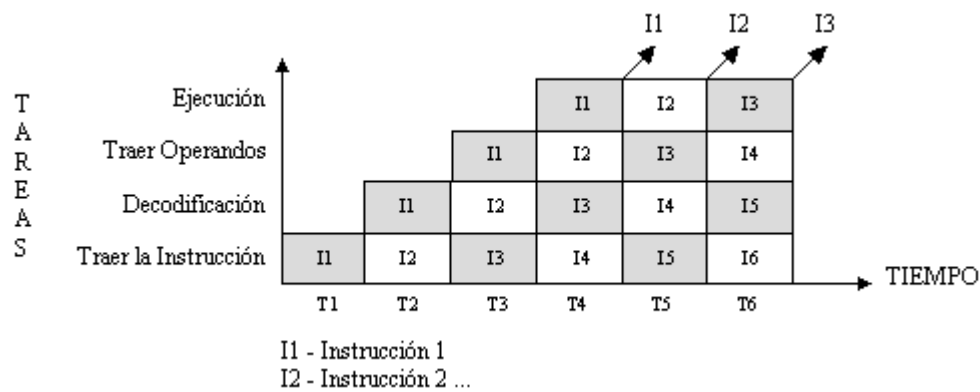


Figura 1.4. Diagrama de Tiempos / Tareas.

En este tipo de arquitecturas es común hacer lecturas y escrituras en la memoria, tanto de instrucciones de programa como de datos, de manera simultánea. Por tal motivo, es necesario que las memorias de datos y programas, con sus respectivos buses, estén separadas. La figura 1.5 muestra un tipo de esta arquitectura conocida como computadora tipo Harvard.

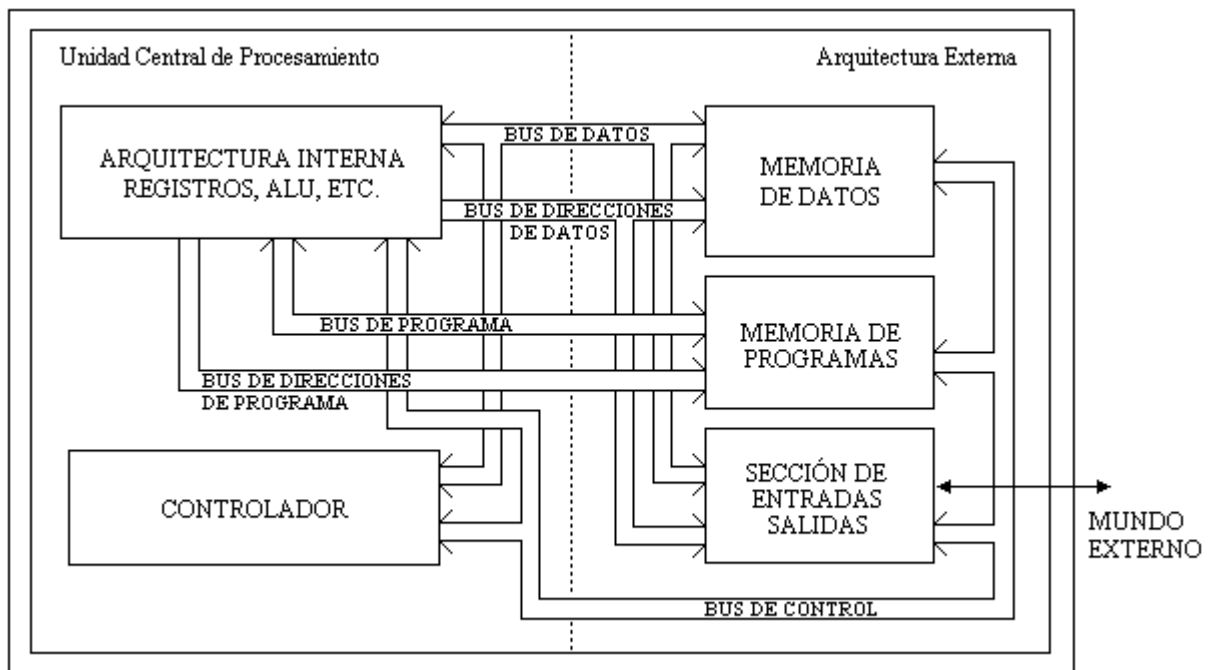


Figura 1.5. Computadora tipo Harvard.

## 1.4 EL CONTROLADOR DE LA COMPUTADORA

Un elemento fundamental es el controlador de la computadora, que como su nombre lo indica, controla y sincroniza todas las operaciones de la arquitectura interna y externa. Esto es, coordina las actividades de la computadora y determina qué operaciones se deben realizar y en qué orden. Este controlador general, denominado Dispositivo de Máquinas de Estados, se describe en el siguiente capítulo.

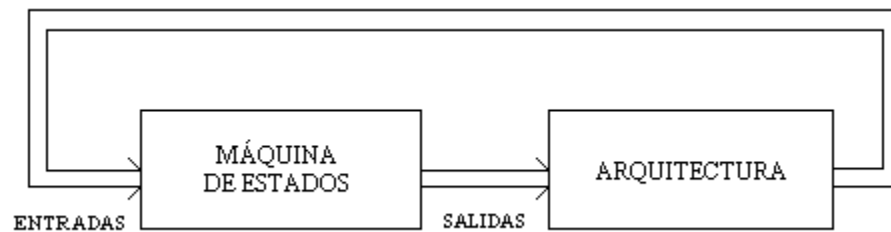


Figura 1.6. Máquina de estados controlando una arquitectura.

## 1.5 COMPUTADORAS PARALELAS

Como se mencionó anteriormente, dentro de la clasificación de computadoras paralelas se encuentran las arquitecturas SIMD y MIMD. El nombre que reciben estas arquitecturas proviene de la clasificación de computadoras que estableció Flynn, quien utilizó los siguientes dos criterios para crear su taxonomía: 1) el número de instrucciones que se están ejecutando simultáneamente en un momento dado, y 2) el número de datos sobre los que se están ejecutando esas instrucciones.

Pero, ¿qué es una computadora paralela?. Una computadora paralela es un tipo de computadora que posee dos o más procesadores independientes que cooperan y se comunican para solucionar un problema. El programador de este tipo de computadoras debe dividir el problema a resolver en varias partes, de manera que el trabajo total sea distribuido entre los distintos procesadores; además, debe calcular la forma en la que cada parte del trabajo se relaciona con el resto de las partes.

El éxito de las computadoras paralelas sobre las computadoras desarrolladas para propósitos especiales se debe a que las primeras ofrecen rendimientos más altos a precios más bajos. Además, la posibilidad de escalamiento es inherente en las computadoras paralelas, en teoría, estas pueden ser actualizadas cambiando o agregando más procesadores.

### 1.5.1 ARQUITECTURAS SIMD

Las arquitecturas SIMD (Single Instruction Multiple Data) ejecutan la misma instrucción sobre múltiples datos simultáneamente; de manera semejante a como un sargento ordena a todo el pelotón dar media vuelta, en lugar de hacerlo de soldado en soldado.

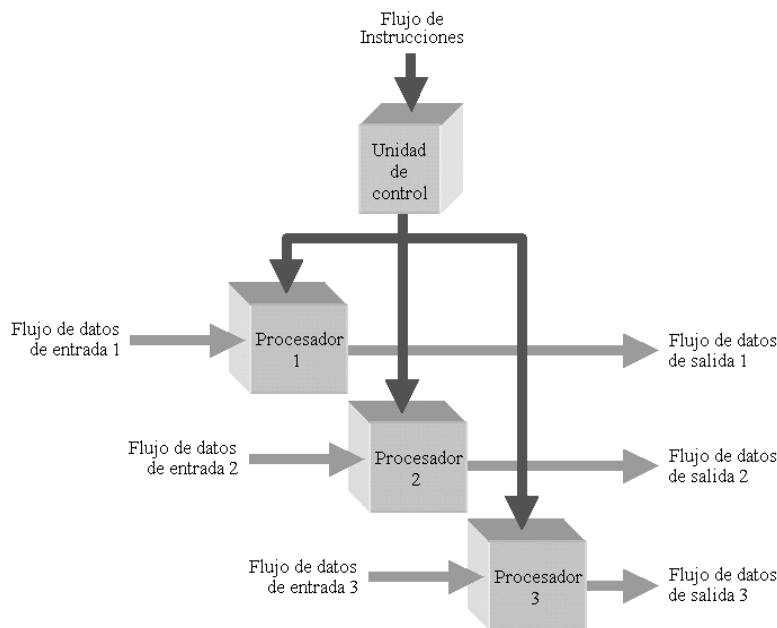


Figura 1.7. Diagrama de bloques de la arquitectura SIMD.

La característica principal de este tipo de arquitecturas es que cuentan con una sola unidad de control, quien es la responsable de interpretar y distribuir la misma instrucción a todos los procesadores. En caso de que no se desee ejecutar la misma instrucción en todos los procesadores, es posible habilitar sólo los procesadores necesarios por medio de una máscara. La figura 1.7 muestra un modelo de arquitectura SIMD de tres procesadores, cada uno operando con su propia memoria local. Todos los procesadores operan bajo el control de una sola secuencia de instrucciones emitida por una unidad de control central. Existen tres flujos de entrada de datos, uno por procesador, los cuales se operan de manera síncrona. En cada paso, todos los procesadores ejecutan la misma instrucción sobre un dato diferente.

La mayoría de las aplicaciones interesantes que utilizan este tipo de computadoras requieren que los procesadores puedan comunicarse entre sí con el fin de intercambiar datos o resultados intermedios. Esta comunicación se logra por alguno de los siguientes dos esquemas: mediante una memoria común (arquitecturas SIMD a memoria compartida) o mediante una red de interconexión (arquitecturas SIMD a memoria distribuida).

En las arquitecturas a memoria compartida todos los procesadores comparten el mismo espacio de memoria; esto significa que el conocimiento de donde se almacenan los datos no es preocupación para el usuario pues hay solamente una memoria para todos los procesadores. En las arquitecturas de memoria distribuida cada procesador tiene su propia memoria asociada. Los procesadores están conectados por medio de una red de interconexión que les permite intercambiar datos entre sus respectivas memorias cuando es necesario. En contraste con las máquinas de memoria compartida, el usuario debe estar enterado de la localización de los datos en las memorias locales y deberá mover o distribuir estos datos explícitamente cuando sea necesario.

Las principales ventajas de estas arquitecturas son las siguientes:

- Funcionan muy bien con vectores de datos.
- La eficiencia es óptima cuando se manejan arreglos de datos en ciclos for.

Entre las desventajas se encuentran las siguientes:

- Las instrucciones de salto y las condicionales no pueden ser ejecutadas en paralelo porque sólo existe una unidad de control de instrucciones.
- El rendimiento decae considerablemente en las sentencias case en un factor de  $1/p$ , donde  $p$  es el número de bloques case.
  - Durante los ciclos while, los datos en algunos procesadores pueden encontrar la condición de salida del ciclo antes de que ocurra en otros procesadores. En este caso, los procesadores que hayan completado el ciclo deben deshabilitarse hasta que el resto de los procesadores cumplan con la condición de salida.

### 1.5.2 ARQUITECTURAS MIMD

Esta es quizás la mejor estrategia de diseño orientada a obtener el más alto rendimiento y la mejor relación costo/rendimiento. La idea detrás de las arquitecturas MIMD (Multiple Instruction Multiple Data) es bastante simple: aumentar la capacidad de trabajo conectando varios procesadores entre sí para obtener un rendimiento global lo más cercano a la suma de rendimientos de cada procesador por separado. La filosofía de trabajo plantea la división de un problema (programa) en varias tareas (procesos) independientes, de manera que a cada procesador se le asigna la resolución de uno de estos procesos.

Generalmente, las arquitecturas de este tipo consisten de varios procesadores autónomos que pueden ejecutar flujos independientes de instrucciones usando datos locales. La característica principal de las arquitecturas MIMD es que son computadoras asíncronas con control descentralizado de hardware, es decir, cada procesador tiene su propia unidad de control ejecutando un programa diferente.



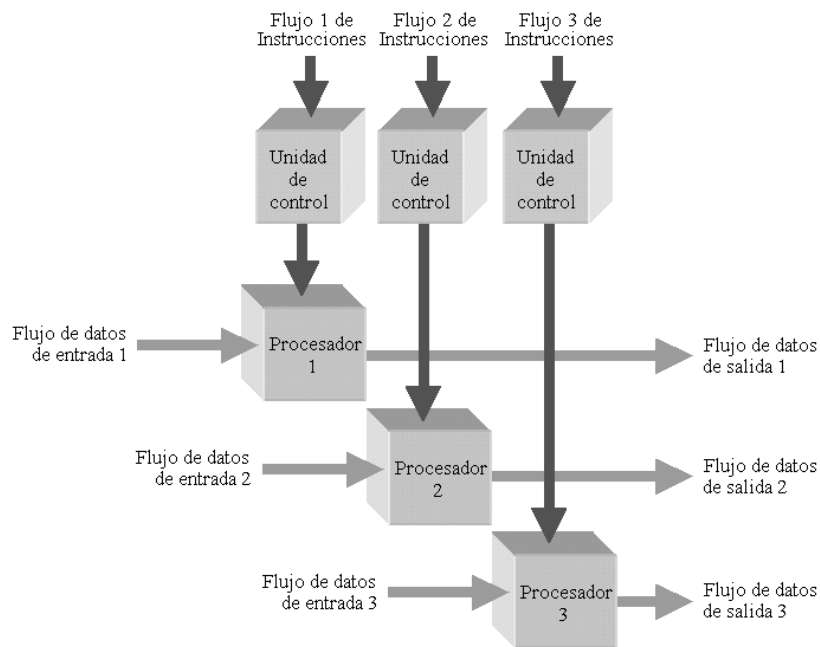


Figura 1.8. Diagrama de bloques de la arquitectura MIMD.

La comunicación entre procesadores se realiza de manera semejante a la arquitecturas SIMD, es decir, mediante una memoria común (arquitecturas MIMD a memoria compartida) o mediante una red de interconexión (arquitecturas MIMD a memoria distribuida).

En las arquitecturas de memoria compartida todos los procesadores se comunican por medio de una memoria común. Por lo tanto, para transmitir un dato desde el procesador  $P_i$  al procesador  $P_j$  son necesarios dos pasos: 1) el procesador  $P_i$  escribe el dato en una dirección de memoria conocida por el procesador  $P_j$ ; y 2) el procesador  $P_j$  lee esa localidad de memoria.

En los sistemas de memoria compartida, permitir múltiples lecturas simultáneas sobre la misma posición de memoria no debe ocasionar ningún problema. Conceptualmente, si cada procesador requiere leer desde una misma posición de memoria copia el contenido de esa posición y lo almacena en su memoria local (memoria caché). Sin embargo, si varios procesadores requieren escribir simultáneamente diferentes datos sobre la misma posición de memoria, entonces, es necesario establecer mecanismos de sincronización entre los procesadores para el acceso a zonas de memoria compartida.

La otra forma de comunicación entre procesadores es por medio de una red de interconexión. En este modelo, la memoria es dividida entre el conjunto de procesadores para su acceso local, además, cada procesador cuenta con su propia memoria caché. La red de interconexión puede ser de dos tipos: 1) directa, en las que existen enlaces físicos que conectan directamente pares de procesadores, permitiendo enviar o recibir datos en cualquier instante de tiempo; y 2) de múltiples etapas, que tienen una baja cantidad de enlaces entre procesadores, de manera que cuando es necesario

comunicar un mensaje entre dos procesadores que no tienen conexión directa, debe encaminarse o enrutarse dicho mensaje por procesadores intermedios entre éstos dos.

En resumen, los MIMD han logrado una posición consolidada en el mercado y han demostrado que para una carga alta de trabajo en tiempo compartido son más eficientes que los SISD. Un programa MIMD no emplea menos tiempo de procesador, pero puede efectuar mayor número de tareas independientes por unidad de tiempo gracias a que distintos programas no comparten el mismo procesador, sino que se ejecutan en procesadores separados y totalmente dedicados.

Por el momento, el único inconveniente es que no existen muchas aplicaciones que corran satisfactoriamente aprovechando las características del paralelismo. Esto no tiene nada que ver con características del hardware, sino más bien con el escaso desarrollo de aplicaciones verdaderamente paralelas aplicables a estas arquitecturas. Finalmente, se muestra un cuadro comparativo entre las arquitecturas SIMD y MIMD en la tabla 1.1.

| <i>Arquitecturas SIMD</i>   | <i>Arquitecturas MIMD</i>  |
|---|--|
| Requiere menos hardware: una unidad de control.   | Requiere más hardware, cada procesador tiene su propia unidad de control.  |
| Necesita menos memoria: una sola copia del programa.  | Necesita más memoria para cada uno de los programas.   |
| Adecuada para aplicaciones que requieran ejecutar las mismas instrucciones sobre un gran número de datos. | Puede ejecutar tareas distintas al mismo tiempo o emular un procesador SIMD mediante mecanismos de sincronización. |
| Necesitan menor tiempo para comunicarse con los procesadores vecinos dado que poseen un reloj global.     | Para comunicarse es necesario usar mecanismos de sincronización.   |
| Son más costosas porque requieren diseñar un microchip de arquitectura especial.                          | Son más baratas porque se pueden construir usando computadoras convencionales de propósito general.                |

Tabla 1.1. Cuadro comparativo entre las arquitecturas SIMD y MIMD.

## PROBLEMAS

1. Indique cuáles fueron las principales aportaciones de las siguientes personas en el área de la computación.
  - Charles Babbage

- Herman Hollerith
- Alan Mathison Turing
- Norbert Wiener
- Claude Elwood Shannon
- John von Neumann

2. ¿Cuál se considera la primera computadora?
3. ¿Cuál fue la razón fundamental por la que se comenzaron a utilizar los números binarios para representar la información en las computadoras?
4. ¿Qué otra aportación hizo a la sociedad Henry Ford además de haber introducido el concepto de línea de ensamble en la fabricación de automóviles?
5. ¿Qué compañía introdujo el concepto de línea de ensamble en la computación y en qué fechas?
6. Mencione brevemente las características de las arquitecturas SIMD e investigue en qué aplicaciones son utilizadas este tipo de computadoras.
7. Mencione brevemente las características de las arquitecturas MIMD e investigue en qué aplicaciones son utilizadas este tipo de computadoras.
8. Investigue las características de las arquitecturas SISD y MISD. ¿En qué aplicaciones se utilizan o se podrían utilizar este tipo de computadoras?