

# Arquitectura **de** computadoras II

**IRMA FLORES RIOS**

**Red Tercer Milenio**

# ARQUITECTURA DE COMPUTADORAS II

# ARQUITECTURA DE COMPUTUTADORAS II

IRMA FLORES RIOS

RED TERCER MILENIO



## AVISO LEGAL

---

**Derechos Reservados © 2012, por RED TERCER MILENIO S.C.**

Viveros de Asís 96, Col. Viveros de la Loma, Tlalnepantla, C.P. 54080, Estado de México.

Prohibida la reproducción parcial o total por cualquier medio, sin la autorización por escrito del titular de los derechos.

Datos para catalogación bibliográfica

Irma Flores Ríos

*Arquitectura de las computadoras II*

ISBN 978-607-733-119-3

**Primera edición: 2012**

Revisión pedagógica: Aurora Leonor Avendaño Barroeta

Revisión editorial: Mónica Gabriela Ortega Reyna

## DIRECTORIO

---

**Bárbara Jean Mair Rowberry**  
*Directora General*

**Rafael Campos Hernández**  
*Director Académico Corporativo*

**Jesús Andrés Carranza Castellanos**  
*Director Corporativo de Administración*

**Héctor Raúl Gutiérrez Zamora Ferreira**  
*Director Corporativo de Finanzas*

**Ximena Montes Edgar**  
*Directora Corporativo de Expansión y Proyectos*

## ÍNDICE

<i>Introducción</i>	4
<i>Objetivo general de aprendizaje</i>	6
<i>Mapa conceptual</i>	7
Unidad 1. Introducción al procesamiento paralelo	8
Introducción	9
Mapa conceptual	10
1.1 Paralelismo en sistemas monoprocesadores	11
1.2 Estructuras de computadores paralelas	14
1.3 Aplicaciones del procesamiento paralelo	16
1.4 Subsistemas de memoria de entrada-salida	17
Autoevaluación	20
Unidad 2. Segmentación encauzada y procesamiento vectorial	25
Mapa conceptual	26
Introducción	27
2.1 Principios de segmentación encauzada lineal	28
2.2 Clasificación de los procesadores encauzados	29
2.3 Cauces generales y tablas de reserva	31
2.4 Cauces de instrucciones y cauces aritméticos	33
2.5 Principios de diseño de los procesadores encauzados	35
2.6 Exigencias del procesamiento vectorial	36
Autoevaluación	38
Unidad 3. Computadores de segmentación encauzada y métodos de vectorización	41
Mapa conceptual	42
Introducción	43
3.1 Los computadores de segmentación encauzada	44
3.2 Procesadores vectoriales iniciales y actuales	47

3.3 Métodos de vectorización y optimización	51
Autoevaluación	53
Unidad 4. Estructuras y algoritmos para procesadores matriciales	57
Mapa conceptual	58
Introducción	59
4.1 Procesadores SIMD	60
4.2 Redes de interconexión SIMD	62
4.3 Algoritmos paralelos para procesadores matriciales	64
4.4 Procesamiento matricial asociativo	67
Autoevaluación	71
Unidad 5. Multiprocesamiento	75
Mapa conceptual	76
Introducción	77
5.1 Arquitectura y programación de los multiprocesadores	78
5.2 Multiprocesamiento: control y algoritmos	84
5.3 Ejemplo de sistemas multiprocesador	87
5.4 Computadores de flujo de datos	89
Autoevaluación	94
<i>Bibliografía</i>	97
<i>Glosario</i>	98

## INTRODUCCIÓN

Día con día las computadoras se vuelven cada vez más necesarias en nuestra vida cotidiana; de repente nos damos cuenta que sin un equipo de cómputo a nuestro lado es muy difícil realizar tareas que podemos hacer desde la comodidad de nuestro hogar y sobre todo a una velocidad impresionante. Hoy en día no tenemos la necesidad de viajar varios kilómetros de distancia para adquirir algún artículo, visitar algún museo, saludar algún amigo, estudiar y trabajar desde el mismo lugar, estas actividades son sólo algunas de las que podemos ser testigos, sin embargo existen actividades de carácter industrial, académico, científico, hidráulico, médico, etc., de las que no nos percatamos; pareciera que los procesos se realizaran automáticamente. Todo en nuestra vida ha cambiado con los recursos que nos da la tecnología.

Estas características que encontramos en nuestras veloces máquinas se deben a un diseño arquitectónico sorprendente, un hardware preciso y una programación hecha a la medida. Si bien es cierto que nuestra materia está enfocada completamente a la parte de arquitectura de los sistemas, es muy importante considerar que todo es parte de un todo interrelacionado.

El presente libro de está constituido por cinco unidades que comprenden la introducción del procesamiento paralelo, la clasificación de los procesadores encauzados, las exigencias del procesamiento vectorial, así como sus métodos de vectorización y optimización, la arquitectura y programación de los sistemas multiprocesadores y algunos ejemplos de sistemas que usan el multiprocesamiento. Sin duda alguna tenemos un gran reto que alcanzar; es importante tener presente el objetivo principal de todo sistema: hacer un diseño potente pero de costo razonable.

Se espera que este libro sirva como guía para el alumno, que genere nuevo conocimiento y se logre una experiencia del aprendizaje enriquecedora con la finalidad de lograr los objetivos propuestos por él, de manera profesional.

El objetivo general del libro es que el estudiante pueda describir, comprender y explicar el procesamiento paralelo, los principios de

segmentación encauzada y procesamiento vectorial, las estructuras de algoritmos para procesadores matriciales y la arquitectura y programación de los multiprocesadores.

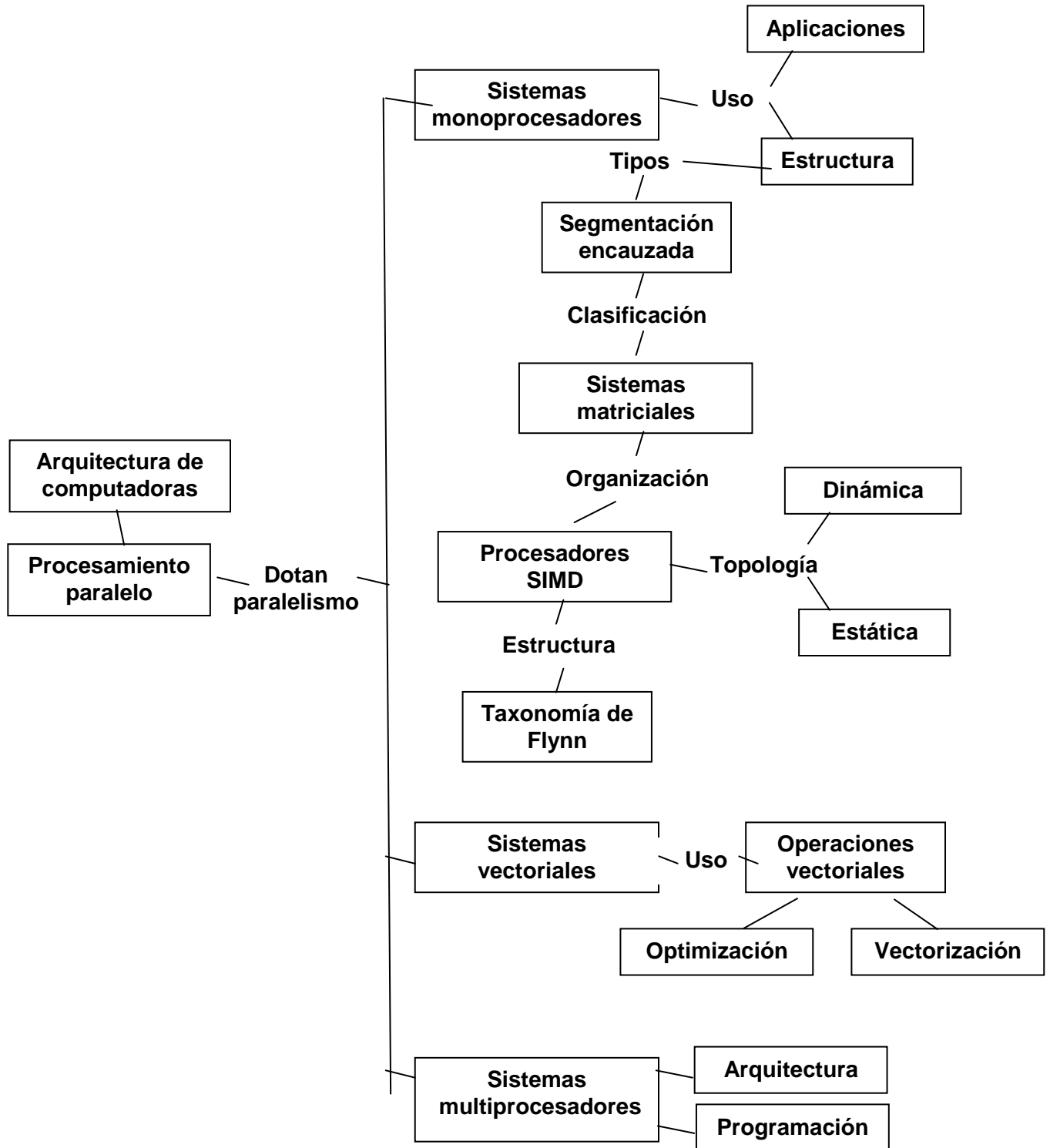
En general, el libro está constituido por el objetivo general, mapas conceptuales, general y por unidad, que permiten darle un panorama al alumno, de lo que verá durante la unidad, lo cual le facilita la comprensión del contenido. Al final de cada unidad se encuentran actividades de aprendizaje y autoevaluaciones diseñadas para que repase y se autoevalúe a partir del conocimiento adquirido; asimismo la bibliografía y el glosario, generado con términos importantes para tener una mejor comprensión del tema.



## OBJETIVO GENERAL

El estudiante describirá, comprenderá y explicará el procesamiento paralelo, los principios de segmentación encauzada y procesamiento vectorial, las estructuras de algoritmos para procesadores matriciales y la arquitectura y programación de los multiprocesadores.

# MAPA CONCEPTUAL



# UNIDAD 1

## INTRODUCCIÓN AL PROCESAMIENTO PARALELO

### OBJETIVO

El alumno comprenderá el funcionamiento y estructura del procesamiento paralelo y el paralelismo en sistemas monoprocesadores, asimismo describirá la estructura de los subsistemas de memoria de entrada-salida.

### TEMARIO

1.1 PARALELISMO EN SISTEMAS MONOPROCESADORES

1.2 ESTRUCTURAS DE COMPUTADORAS PARALELAS

1.3 APLICACIONES DEL PROCESAMIENTO PARALELO

1.4 SUBSISTEMAS DE MEMORIA DE ENTRADA-SALIDA

## INTRODUCCIÓN

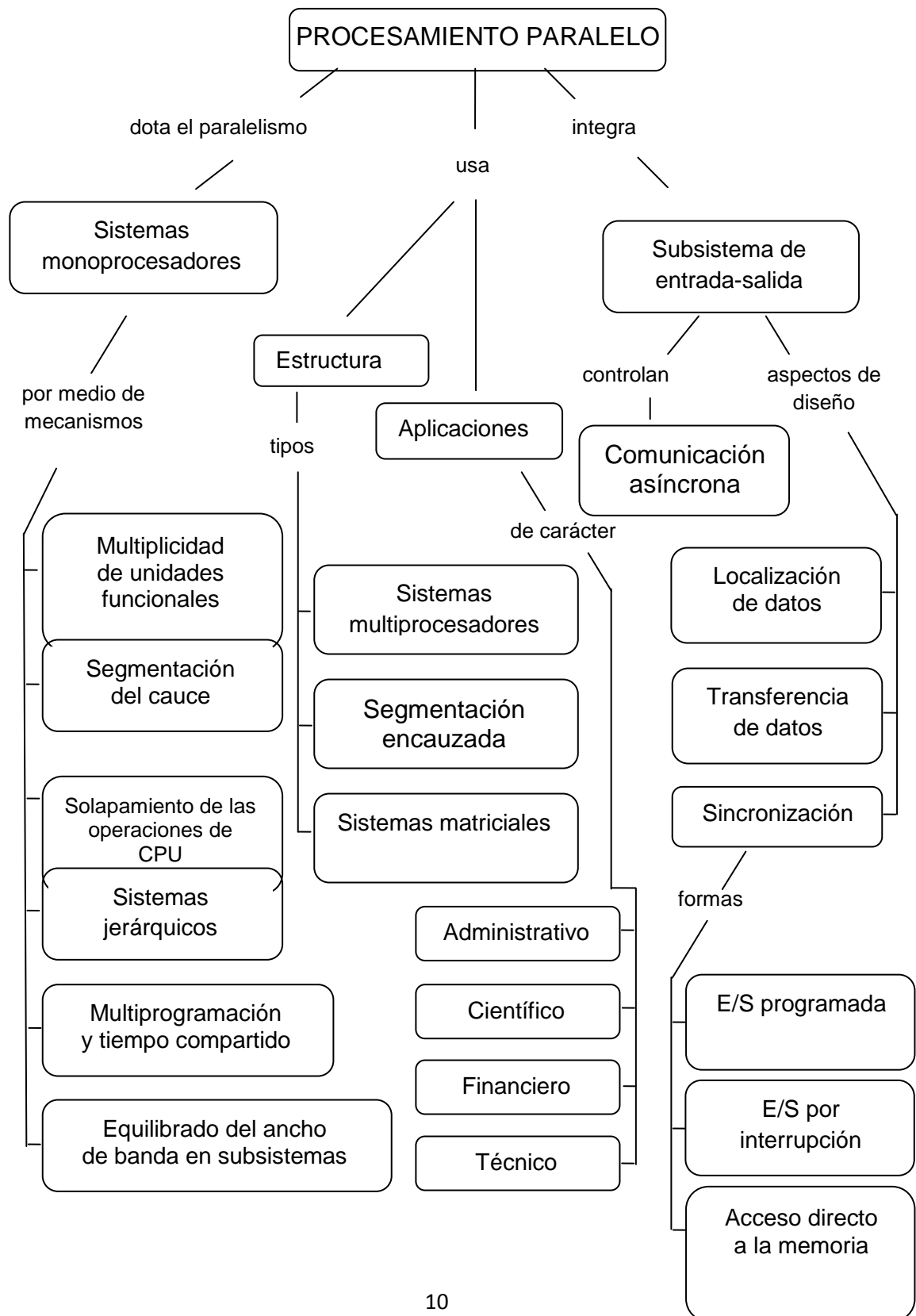
Hoy en día, en nuestra vida cotidiana, tenemos la necesidad de realizar un sinnúmero de actividades en el menor tiempo posible. En la mayoría de las ocasiones no nos percatamos cómo es que estas actividades poco a poco resultan ser más rápidas y sencillas para el ser humano.

Pues bien, lo que ocurre internamente en los procesadores de los sistemas, lo cual para el usuario final es totalmente transparente, tiene que ver en gran parte con el tema de esta unidad, el procesamiento paralelo. Esto no es otra cosa que realizar pequeñas divisiones de procesos, los cuales están compuestos por un fragmento de instrucciones, que son ejecutadas de forma simultánea en diferentes procesadores obteniendo así un tiempo de ejecución más corto. De hecho, ésta es una de las principales metas que tiene el procesamiento paralelo.

Trabajar y poder procesar grandes volúmenes de información con el fin de obtener resultados en el menor tiempo posible es otra de sus ventajas, sin embargo también se enfrenta a algunas dificultades como son la falta de coordinación al tratar de ejecutar los procesos al mismo tiempo, así como su control y monitoreo.

Sin duda alguna, este tema es muy importante por lo que se pretende que el alumno comprenda el funcionamiento, la estructura y las aplicaciones del paralelismo en los sistemas.

# MAPA CONCEPTUAL



## 1.1 PARALELISMO EN SISTEMAS MONOPROCESADORES

Un sistema monoprocesador es aquel que cuenta sólo con un procesador para ejecutar todos los procesos a realizar, bajo este concepto podríamos afirmar que en este tipo de sistemas no existe el paralelismo.

En la siguiente figura se muestra un ejemplo de la diferencia que existe entre los sistemas monoprocesadores y los multiprocesadores, estos últimos siendo los que cuentan con más de un procesador para ejecutar los diferentes procesos de forma simultánea.

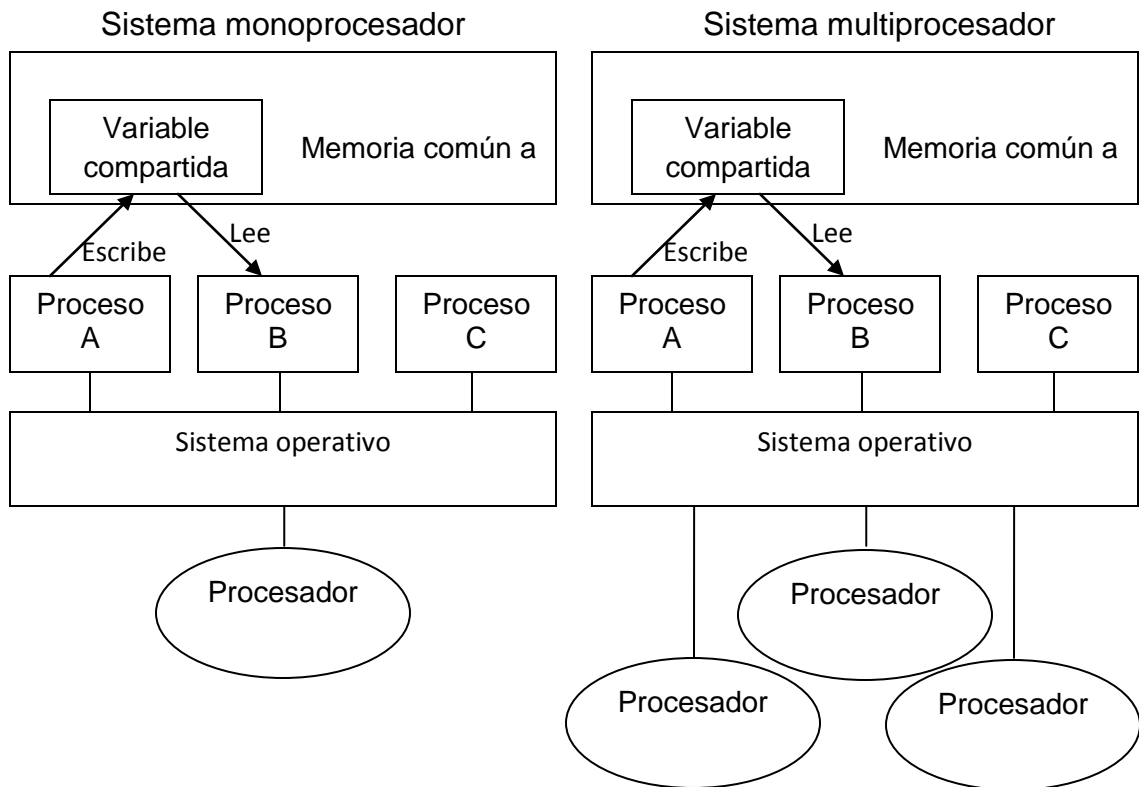


Figura 1.<sup>1</sup>

Sin embargo, haciendo uso de algunos mecanismos del procesamiento paralelo podemos observar que en los sistemas monoprocesadores se puede dar el paralelismo.

Uno de los mecanismos que se mencionan anteriormente se refiere a la multiplicidad de unidades funcionales, antes de explicar a qué se refiere este

<sup>1</sup> Fuente: [http://sopa.dis.ulpgc.es/itis-sc-01-1-conceptos\\_fundamentales](http://sopa.dis.ulpgc.es/itis-sc-01-1-conceptos_fundamentales)

mecanismo recordemos que la unidad aritmético lógica (UAL) o en sus siglas en ingles, Arithmetic Logic Unit (ALU), es un circuito digital que se encarga de realizar las operaciones aritméticas y lógicas entre los datos, dentro de la CPU. Entonces podemos decir, que las funciones de la unidad aritmética lógica (UAL) se subdividen en dos o más unidades funcionales especializadas, donde cada una es encargada de llevar a cabo una parte del ciclo de la instrucción, desarrollan tareas en paralelo obteniendo un mejor tiempo de respuesta en el proceso.

En la siguiente figura se muestra un ejemplo de la cantidad y tipo de unidades funcionales que pueden llevarse a cabo por medio de cada tipo de procesador.

Procesador	Cantidad de unidades funcionales
CDC-6600	10
IBM 360/91	2
Premium Pro	6
Nx686	4
MIPS R10.000	4
Alpha 21064	4
PowerPC620	3

Figura 2.<sup>2</sup>

Otro de los mecanismos que podemos mencionar es la segmentación del cauce, este mecanismo describe las siguientes fases de ejecución que conforman la cola de una instrucción también llamada pipeline, por medio de la cual se comunican las unidades funcionales:

- ✓ Extracción de la instrucción.
- ✓ Decodificación.
- ✓ Extracción de operandos.
- ✓ Ejecución de operación.

<sup>2</sup> Fuente: <http://atc2.aut.uah.es/~acebron/cap1.pdf>

✓ Almacenamiento.

Cada una de estas fases se fraccionan en etapas y son ejecutadas y almacenadas una tras otra formando un cauce.

Un tercer mecanismo usado para dotar el paralelismo es el solapamiento de las operaciones de CPU y de entrada/salida. Las operaciones de entrada/salida se pueden ejecutar simultáneamente con los cálculos de CPU pero existe el inconveniente, dependiendo del dispositivo, que el proceso se vuelve demasiado lento. Para solucionar este problema se hace uso de los controladores de entrada/salida canales o procesadores entrada/salida separados.

Debido a que esta solución requiere más recursos, la unidad de acceso directo a la memoria (DMA) nos permite que la comunicación entre los diferentes dispositivos sea directa, sin someter a la CPU a una carga masiva de interrupciones.

El uso de sistemas jerárquicos de memoria es un mecanismo importante para el paralelismo, ya que éste nos permite dividir la memoria en niveles jerárquicos, lo cual nos ayuda a cortar la diferencia de velocidad que existe entre el procesador y la memoria. Debido a su localidad espacial y temporal se puede hacer uso de memorias más pequeñas pero más rápidas entre el procesador y la memoria principal.

El equilibrado del ancho de banda en los subsistemas es otro de los criterios que se están explicando en este tema, y empezaremos por definir qué es el ancho de banda, ya que lo que se busca al dar paralelismo con este criterio es resolver el problema de saturación del canal. El ancho de banda es la máxima cantidad de datos que pueden pasar por un camino de comunicación en un momento dado, normalmente medido en segundos. Cuanto mayor sea el ancho de banda, más datos podrán circular por ella al segundo.<sup>3</sup>

Con la finalidad de equiparar los anchos de banda de procesamiento es necesario tomar en cuenta que la CPU es la unidad más rápida, después la memoria principal y la más lenta son los dispositivos de entrada/salida, así que

---

<sup>3</sup> Fuente: <http://www.lawebdelprogramador.com/diccionario>



la solución para el equilibrado del ancho de banda entre CPU y la memoria principal ha sido la implementación de memorias caché, y para el equilibrado del ancho de banda entre memoria y dispositivos de entrada/salida se ha implementado el uso de canales a diferentes velocidades y controladores de disco inteligentes o máquinas de buses de datos, que nos permiten, filtrar los datos relevantes de las pistas del disco.

El último de los criterios que veremos en este tema es la multiprogramación y el tiempo compartido. Con la multiprogramación se busca la interacción de unos programas con otros ya que esto favorece una mejor gestión de los recursos mediante el uso de las operaciones de entrada/salida. El sistema de tiempo compartido es una extensión de la multiprogramación, debido a esto podemos decir que aplica el mismo principio. La principal ventaja del uso de tiempo compartido es que evita que un proceso sea monopolizado por el procesador. Este proceso es gestionado por el sistema operativo.

## ACTIVIDAD DE APRENDIZAJE

Elaborar un cuadro sinóptico relacionando la información del procesamiento paralelo con los sistemas monoprocesadores.

### 1.2 ESTRUCTURAS DE COMPUTADORES PARALELOS

Una computadora paralela la podemos definir como un dispositivo que cuenta con dos o más procesadores interconectados, los cuales son capaces de intercambiar información y ejecutar instrucciones de forma simultánea.

Este tipo de computadoras están divididas en tres tipos: de segmentación encauzada, matriciales y sistemas multiprocesadores, cada una de ellas explota y obtiene el paralelismo de forma diferente, es decir, los procesadores de segmentación encauzada generan computaciones solapadas para lograr paralelismo temporal; los matriciales emplean múltiples unidades aritmético-lógicas sincronizadas para lograr el paralelismo espacial, y los sistemas

multiprocesadores logran el paralelismo asíncrono ejecutando un conjunto de procesadores interactivos que disponen de recursos compartidos.

Como lo vimos anteriormente, en un procesador de segmentación encauzada la ejecución de una instrucción se ejecuta de modo solapado, donde dicha instrucción está constituida por cuatro pasos principales, los cuales son:

1. BI (búsqueda de la instrucción).
2. DI (decodificación de la instrucción).
3. BO (búsqueda del operando).
4. EJ (ejecución de la instrucción).

En el caso de las computadoras matriciales, éstas alcanzan el paralelismo de tipo especial, esto se logra gracias a la replicación de las funciones en múltiples unidades aritmético-lógicas, también llamadas elementos de proceso (EP). Un procesador matricial es un computador paralelo síncrono que permite la ejecución de instrucciones al mismo tiempo.<sup>4</sup>

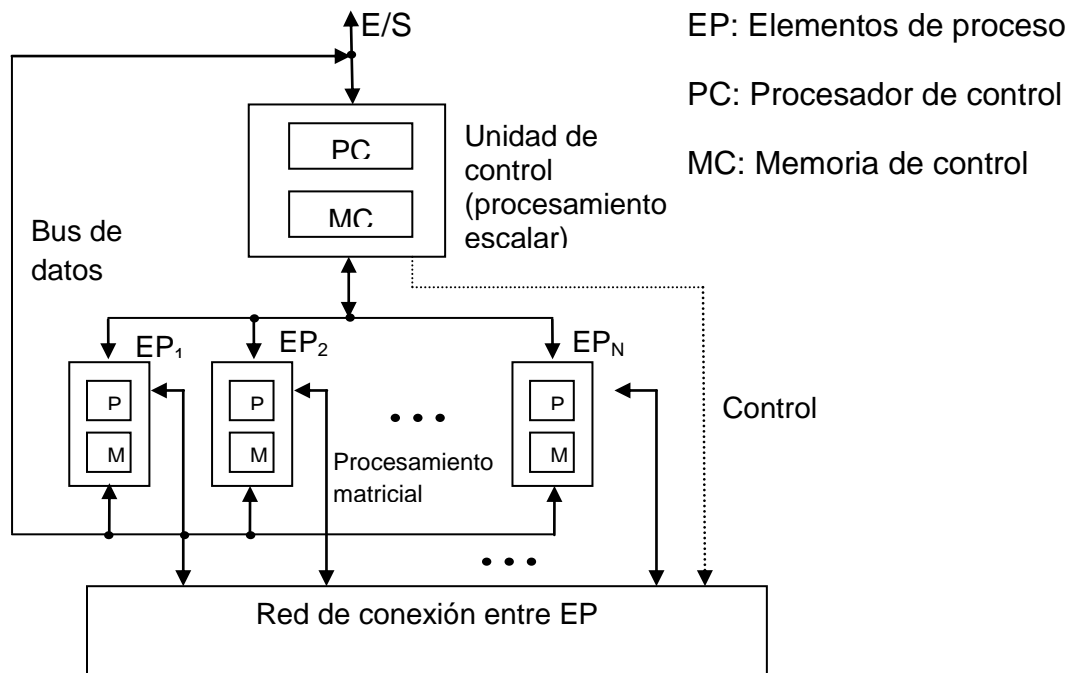


Figura 4. Estructura funcional de un procesador matricial.<sup>5</sup>

<sup>4</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

<sup>5</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

Los sistemas multiprocesadores son aquellos que están constituidos por más de dos procesadores, cada uno de éstos tienen una memoria local pero comparten accesos a una memoria en común dividida en módulos, esta memoria principal es un medio por el cual los procesadores pueden tener comunicación, así como la red de interconexión y la gestión de interrupciones.

Los sistemas multiprocesadores están gestionados por el sistema operativo y pueden acceder a los dispositivos propios de entrada/salida.

## ACTIVIDAD DE APRENDIZAJE

Dar seguimiento al cuadro sinóptico relacionando la información del procesamiento paralelo, los sistemas monoprocesadores y la estructura de computadores paralelos.

### 1.3 APLICACIONES DEL PROCESAMIENTO PARALELO

En este tema explicaremos algunas de las alternativas donde podríamos aplicar prácticamente el procesamiento paralelo; así como este proceso es muy utilizado en el área de cómputo, también podemos hacer uso de él en los ámbitos administrativo, técnico, científico, financiero, etcétera.

Podríamos empezar formulando una pregunta sobre rendimiento o velocidad, por ejemplo, ¿por qué es importante conseguir mayor velocidad? Supongamos que contamos con un algoritmo que evalúa algún problema en particular, que nos interesa de la vida diaria, por ejemplo la contaminación del Valle de México, aplicando este tipo de algoritmo al problema de la contaminación, podríamos obtener datos muy rápidos, útiles y sobre todo oportunos. Otro caso podría ser un problema de salud, aplicando nuestro algoritmo podríamos tener los resultados del estado de salud de la persona inmediatamente.

En todos los ámbitos estas aplicaciones pueden ser de gran utilidad, es por eso que el procesamiento paralelo se extiende al sector privado, el comercio, la educación, las pequeñas y medianas empresas, así como las

instancias gubernamentales donde son requeridos sistemas computacionales de alto rendimiento.<sup>6</sup>

## ACTIVIDAD DE APRENDIZAJE

Dar seguimiento al cuadro sinóptico relacionando la información del procesamiento paralelo y sus aplicaciones, los sistemas monoprocesadores y la estructura de computadores paralelos.

### 1.4 SUBSISTEMAS DE MEMORIA DE ENTRADA-SALIDA

Los subsistemas de entrada/salida son aquellos que controlan las comunicaciones asíncronas respecto de la CPU. El procesador y la memoria funcionan regulados completamente por el reloj.<sup>7</sup>

En la figura 5 se observa cómo una petición de disco puede tardar hasta 20 minutos, antes de empezar a transmitir 4096 bytes a una tasa de transferencia de un byte, cada pocos ciclos de reloj.

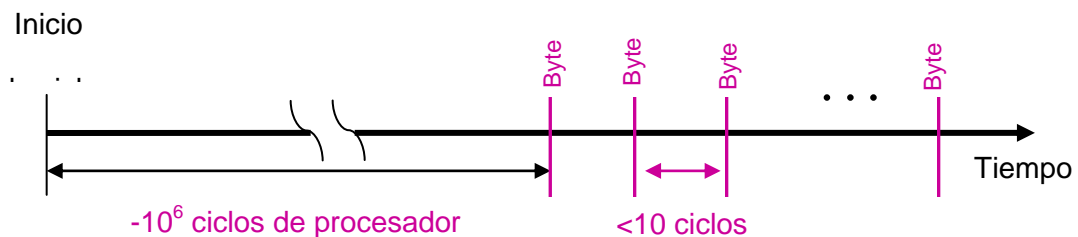


Figura 5.<sup>8</sup>

Sobre el diseño de los subsistemas de memoria de entrada/salida se consideran algunos aspectos como la localización y la transferencia de datos y la sincronización, este último aspecto usa las siguientes formas para la sincronización:

<sup>6</sup> Fuente: <http://www.di.ujiaen.es/assignaturas/ArqTel/teoria/Tema1.pdf>

<sup>7</sup> Fuente: <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>

<sup>8</sup> Fuente: <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>

- Entrada/salida programada.
- Entrada/salida por interrupción.
- Acceso directo a la memoria.

Para finalizar este tema revisaremos la estructura del bus de entrada/salida. Todas las computadoras tienen distintos grados de separación entre la transmisión de datos de entrada/salida y la memoria. En las figuras 6, 7 y 8 se muestran los diferentes grados de separación mencionados.

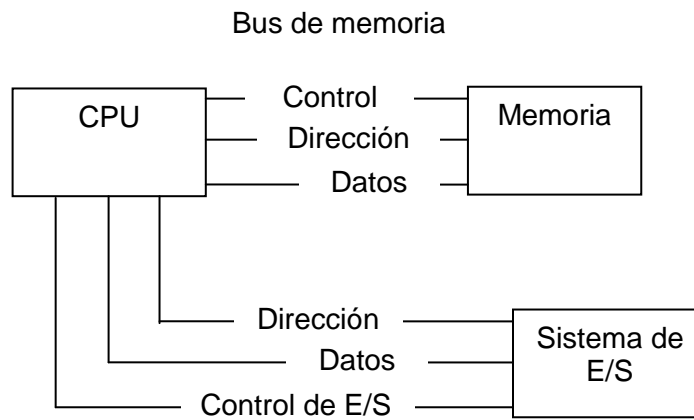


Figura 6. Entrada/salida aislada.<sup>9</sup>

La entrada/salida aislada contiene subbuses de dirección, datos y control separados.

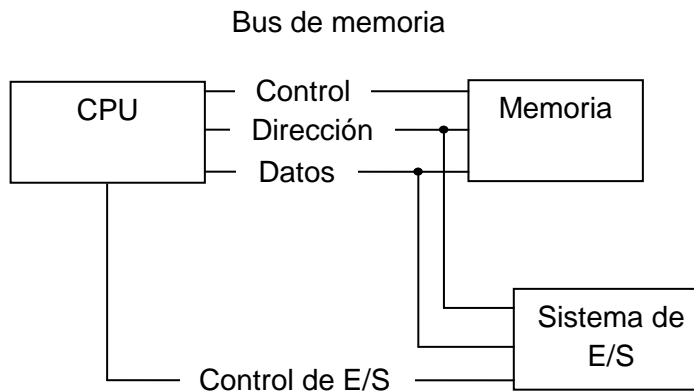


Figura 7. Entrada/salida compartida.<sup>10</sup>

<sup>9</sup> Fuente: <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>

<sup>10</sup> Fuente: <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>

La entrada/salida compartida tiene en común subbuses de datos y direcciones.

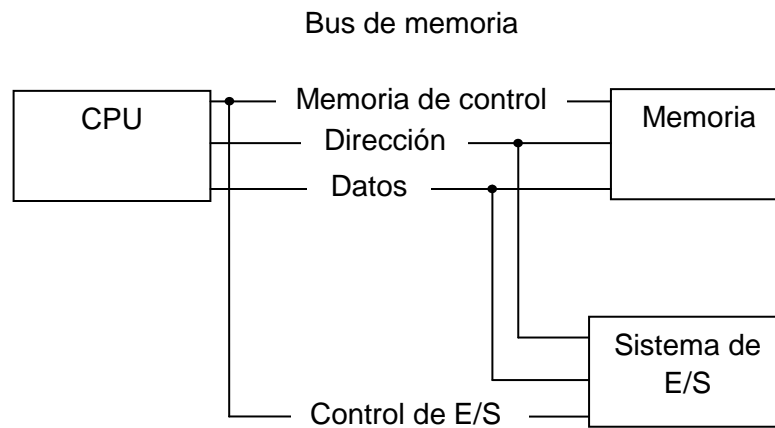


Figura 8. Entrada/salida mapeada en memoria.<sup>11</sup>

La entrada/salida mapeada en memoria utiliza un solo bus para entrada/salida y memoria. Este tipo de subsistema se usa hoy en día en la mayoría de los procesadores.

## ACTIVIDAD DE APRENDIZAJE

Desarrollar el cuadro sinóptico en un documento de word, relacionando la totalidad de los temas de la unidad 1 y enviarlo al docente de forma electrónica.

---

<sup>11</sup> Fuente: <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>

## AUTOEVALUACIÓN

1. Observa la siguiente tabla y relaciona cada uno de los mecanismos que se usan para dotar el paralelismo en los sistemas monoprocesadores, con la descripción que a tu juicio establezca la relación correcta y marca con una **X** sólo uno de los incisos.

Mecanismo	Descripción
_____	a) Las funciones de la UAL se subdividen y cada una de éstas es encargada de llevar a cabo una etapa del ciclo de la instrucción.
1. Uso de sistemas jerárquicos de memoria.	b) La fragmentación de la memoria permite cortar la diferencia de velocidad que existe entre el procesador y la memoria.
2. Solapamiento de operación de CPU y E/S.	c) Describe cinco fases de ejecución que conforman la cola de una instrucción también llamada pipeline.
3. Multiplicidad de unidades funcionales.	d) Dos operaciones se pueden ejecutar simultáneamente teniendo como inconveniente la lentitud del proceso.
_____	e) Busca la interacción entre programas, favorece la gestión de recursos mediante el uso de las operaciones de entrada/salida.

- a) 1b; 2d; 3a.
- b) 1a; 2e; 3c.
- c) 1c; 2d; 3b.
- d) 1d; 2c; 3d.

2. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta que a tu juicio sea la correcta.

\_\_\_\_\_ es el mecanismo conformado por fases de ejecución que conforman, a su vez, la cola de una instrucción también llamada pipeline.

- a) Equilibrado de los ancho de banda de los subsistemas.
- b) Multiprogramación.
- c) Segmentación en cauce.
- d) Tiempo compartido.

3. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta que a tu juicio sea la correcta.

La segmentación del cauce maneja cinco fases de ejecución que conforman la cola de una instrucción, ¿cuál de los incisos nombra tres de estas fases?

- a) Almacenamiento, decodificación, multiprogramación.
- b) Ejecución de operación, almacenamiento, tiempo compartido.
- c) Extracción de la instrucción, multiprogramación, decodificación.
- d) Almacenamiento, decodificación, extracción de operandos.

4. Observa la siguiente tabla y relaciona cada uno de los tipos de computadores paralelos con la forma que le corresponda para lograr el paralelismo, marca con una **X** sólo uno de los incisos con la respuesta, que a tu juicio, establezca la relación correcta.

<b>Tipo de computador</b>	<b>Tipo de paralelismo</b>
1. Segmentación encauzada.	a) Especial.
2. Sistemas matriciales.	b) Temporal.
3. Sistemas multiprocesadores.	c) Asíncrono.

- a) 1b; 2a; 3c.
- b) 1a; 2b; 3c.



c) 1c; 2b; 3a.

d) 1b; 2c; 3a.

5. Lee la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta que a tu juicio sea la correcta.

Los \_\_\_\_\_ están clasificados en tres tipos, dos de estas clasificaciones son sistemas matriciales y sistemas multiprocesadores.

a) Sistemas monoprocesadores.

b) Computadores paralelos.

c) Computadores segmentados.

d) Procesadores vectoriales.

6. Completa el siguiente concepto escribiendo la palabra que falte en el espacio correspondiente.

Las computadoras matriciales alcanzan el paralelismo de tipo especial, gracias a \_\_\_\_\_ en múltiples unidades aritmético-lógicas también llamadas elementos de proceso (EP).

7. Marca con una **X** sólo uno de los incisos con la respuesta, que a tu juicio sea la correcta.

a) Es aquel que ejecuta una instrucción de modo solapado donde la instrucción indica los pasos búsqueda de la instrucción, decodificación de la instrucción, búsqueda del operando y ejecución de la instrucción. Sistema matricial.

b) Sistema monoproceso.

c) Segmentación encauzada.

d) Sistema multiproceso.

8. Haciendo referencia a la estructura del bus de memoria de entrada/salida, las computadoras tienen distinto grado de separación entre la transmisión de datos de entrada/salida y la memoria. Observa la siguiente tabla y relaciona la estructura del bus de memoria con el grado de separación que corresponde con cada una; marca con una **X** sólo uno de los incisos con la respuesta que a tu juicio establezca la relación correcta.

Estructura de bus de memoria	Grado de separación
1. Tiene en común buses de datos y direcciones.	a) Compartida.
2. Utiliza sólo un bus para entrada/salida y memoria.	b) Mapeada en memoria.
3. Contiene buses de dirección, datos y control separados.	c) Aislada.

- a) 1a; 2c; 3b.
- b) 1b; 2c; 3a.
- c) 1a; 2b; 3c.
- d) 1c; 2a; 3b.

9. Completa el siguiente concepto escribiendo la palabra que falte en el espacio correspondiente.

El diseño de los \_\_\_\_\_ considera algunos aspectos como la localización y transferencia de datos, y la sincronización.

10. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta que a tu juicio sea la correcta.

La sincronización, es parte del diseño de los subsistemas de memoria de entrada/salida, ¿cuáles son las formas usadas por este aspecto?

- a) Entrada/salida programada, acceso directo a la memoria, entrada/salida por interrupción.
- b) Entrada/salida por instrucción, entrada/salida por interrupción, entrada/salida programada.
- c) Entrada/salida por extracción, entrada/salida por interrupción, entrada/salida por instrucción.
- d) Acceso directo a la memoria, entrada/salida por interrupción, entrada/salida por extracción.

## RESPUESTAS

1. *a.*
2. *c.*
3. *d.*
4. *a.*
5. *b.*
6. Replicación de las funciones.
7. *c.*
8. *c.*
9. Subsistemas de memoria de entrada/salida.
10. *a.*

## UNIDAD 2

# SEGMENTACIÓN ENCAUZADA Y PROCESAMIENTO VECTORIAL

### OBJETIVO

El alumno estudiará y comprenderá los principios de la segmentación encauzada lineal, la clasificación de los procesadores encauzados y los principales términos del procesamiento vectorial.

### TEMARIO

2.1 PRINCIPIOS DE SEGMENTACIÓN ENCAUZADA LINEAL

2.2 CLASIFICACIÓN DE LOS PROCESADORES ENCAUZADOS

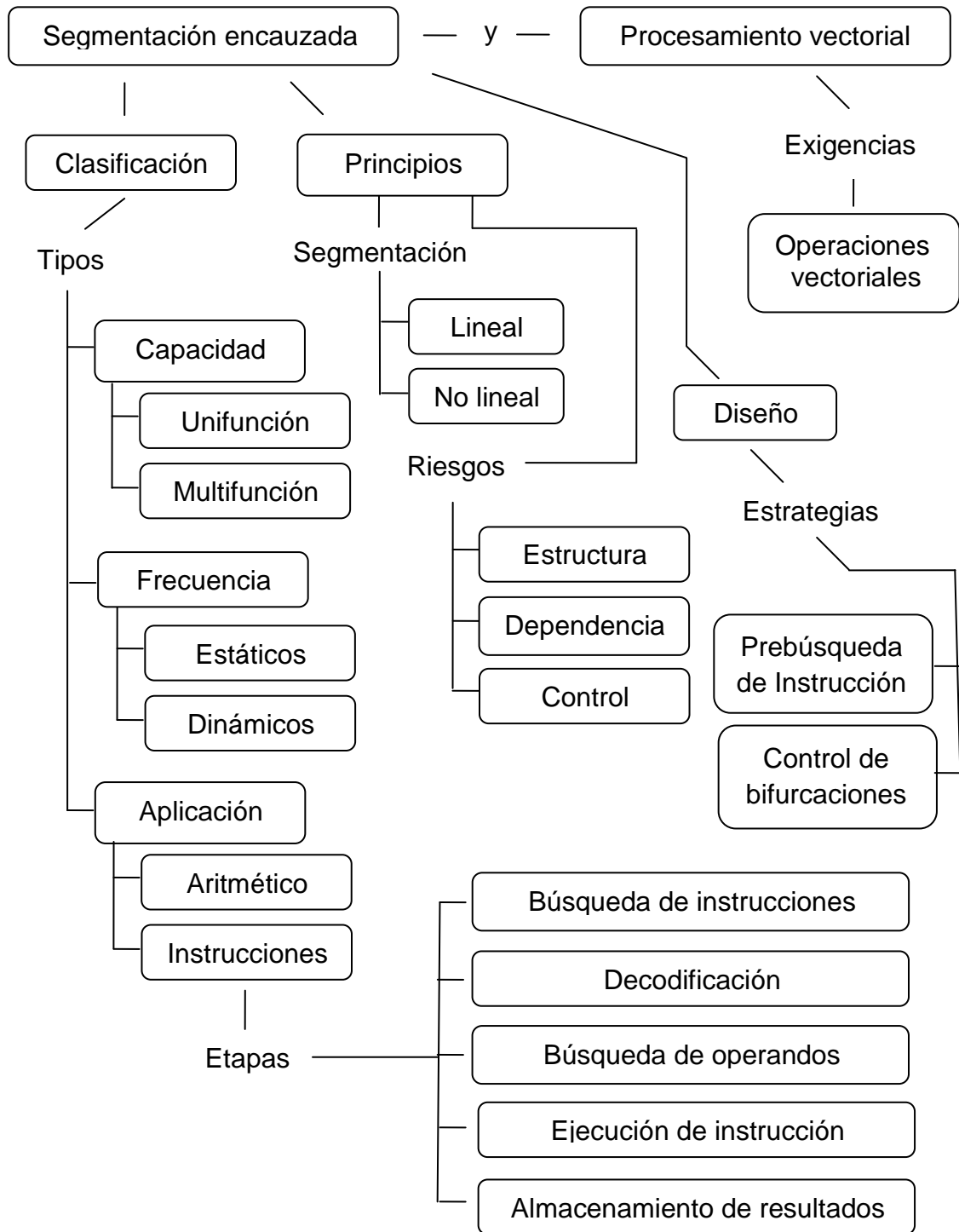
2.3 CAUCES GENERALES Y TABLAS DE RESERVA

2.4 CAUCES DE INSTRUCCIONES Y CAUCES ARITMÉTICOS

2.5 PRINCIPIOS DE DISEÑO DE LOS PROCESADORES ENCAUZADOS

2.6 EXIGENCIAS DEL PROCESAMIENTO VECTORIAL

# MAPA CONCEPTUAL



## INTRODUCCIÓN

Es importante mencionar que en esta unidad, así como en las unidades anteriores y asignaturas anteriores, se lleva una secuencia de aprendizaje, por lo que te recomiendo que tengas presente lo visto en la unidad 1, así como conceptos y principios fundamentales de la materia Arquitectura de computadoras I.

Recordemos que en la unidad 1 revisamos la introducción al procesamiento paralelo, así como la estructura de los sistemas paralelos, y justamente uno de los tipos que compone dicha estructura es el tema que tocaremos en esta unidad, segmentación encauzada.

En esta unidad revisaremos los principios básicos de la segmentación encauzada lineal, así como su clasificación, principios de diseño y fundamentos del procesamiento vectorial.

## 2.1 PRINCIPIOS DE SEGMENTACIÓN ENCAUZADA LINEAL

En la segmentación encauzada lineal todas las etapas de las unidades funcionales son ejecutadas de forma secuencial y se caracterizan por tener un orden de precedencia en la ejecución de las subtarefas del cauce, esto quiere decir, que una tarea no puede empezar su ejecución hasta que una previa haya finalizado. En este tipo de segmentación no existe la posibilidad de tener lazos hacia atrás.

Sin embargo, en la segmentación encauzada no lineal existe cierta flexibilidad en el orden y la forma de ejecución de estas subtarefas. Hay posibilidad de tener lazos hacia atrás y se puede pasar más de una vez por cada etapa en cada iniciación de datos.

En las figuras 1 y 2 podemos observar la diferencia que existe entre la segmentación encauzada lineal y la no lineal.

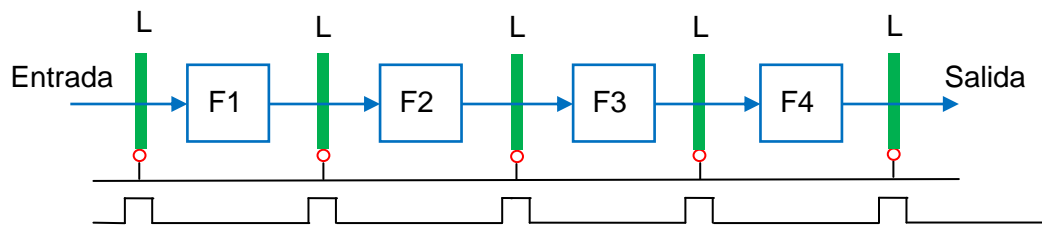


Figura 1. Segmentación encauzada lineal.<sup>12</sup>

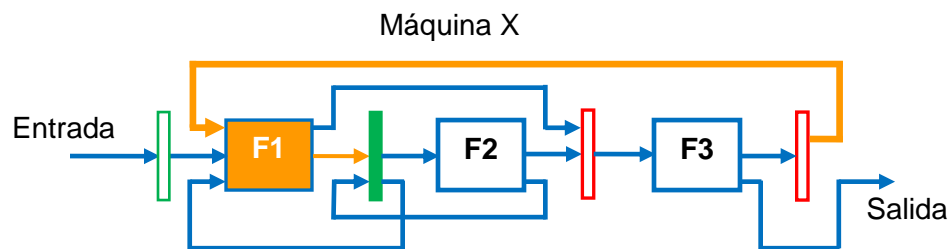


Figura 2. Segmentación encauzada no lineal.<sup>13</sup>

En la figura 3 observamos un ejemplo de grafos de precedencia, donde el primer grafo muestra la segmentación lineal, también podemos observar

<sup>12</sup> Fuente: <http://www.nebrija.es/~jmaestro/AT5118/Segmentacion.pdf>

<sup>13</sup> Fuente: <http://www.nebrija.es/~jmaestro/AT5118/Segmentacion.pdf>

cómo las tareas pasan por todas y cada una de las etapas, en orden y una sola vez, y existe sólo un punto de entrada y uno de salida.

En la segmentación no lineal, la tarea puede no pasar por todas las etapas y tener múltiples puntos de entrada y salida.

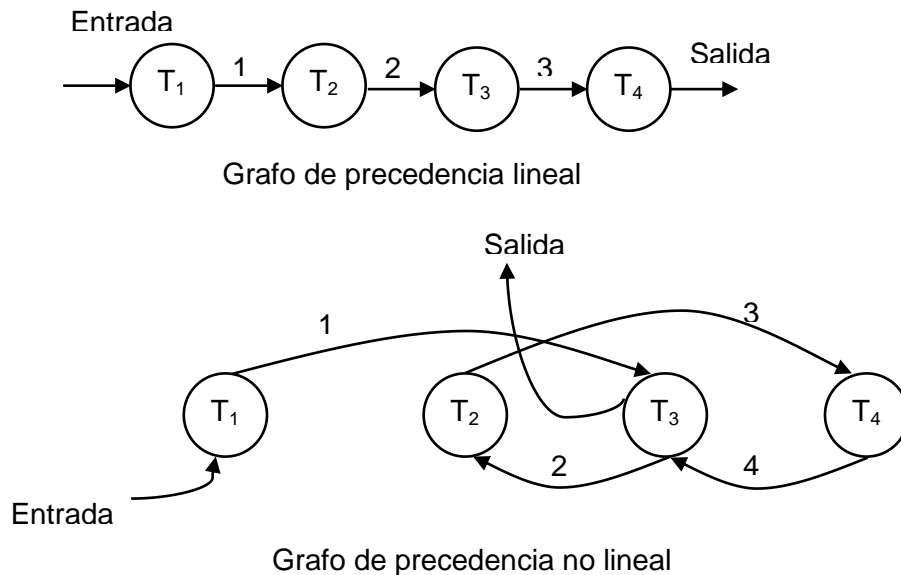


Figura 3. Grafos de precedencia.<sup>14</sup>

## ACTIVIDAD DE APRENDIZAJE

Elabora un resumen sobre la segmentación encauzada lineal y explica con tus propias palabras la diferencia que existe entre la segmentación encauzada lineal y no lineal.

### 2.2 CLASIFICACIÓN DE LOS PROCESADORES ENCAUZADOS

¿Cómo se clasifican los procesadores encauzados? Existen varios criterios para establecer la clasificación de los procesadores encauzados, veamos a continuación.

1. Por su frecuencia de cambio de función:

<sup>14</sup> Fuente: <http://www.snip.gob.ni/Xdc/amc3/Folleto.pdf>



- Estáticos. Son aquellos que asumen una configuración funcional única, la cual no debe modificarse frecuentemente. Pueden ser de tipo unifunción o multifunción.
- Dinámicos. Son aquellos que asumen varias configuraciones funcionales simultáneamente, por esta razón su rendimiento puede ser muy bajo. En una configuración dinámica es necesario el uso de mecanismos de control más elaborados que los que se usan en la configuración estática.

2. Por su capacidad:

- Unifunción. Es aquel que sólo puede ejecutar una única función sobre los datos, por ejemplo el sumador de coma flotante, una unidad encauzada con una función fija y dedicada.
- Multifunción. Es aquel, que dependiendo de la señal de control, puede ejecutar varias funciones sobre los datos en diferentes momentos o a la vez.

3. Por su posibilidad o no de retroalimentación:

- Lineales. No permiten retroceder etapas hacia atrás, la ejecución de las tareas es secuencial.
- No lineales. Son aquellos que tienen un grado de flexibilidad, ya que permiten tener varios puntos de entrada y salida. Permiten hacer ejecuciones de tareas en más de una etapa.

4. Por su aplicación:

- Operadores aritméticos segmentados. Realizan varias operaciones encauzadas de ALU en diferentes formatos de datos.
- Cauces del flujo de instrucción. Regularmente son cauces lineales, algunas de sus fases pueden sub-segmentarse y permitir solapar la ejecución de la instrucción con la decodificación y búsqueda de los operandos de las instrucciones subsiguientes.

## ACTIVIDAD DE APRENDIZAJE

Elabora un cuadro sinóptico con la clasificación de los procesadores encauzados y expón el tema a tus compañeros.

### 2.3 CAUCES GENERALES Y TABLAS DE RESERVA

Los conflictos estructurales en los cauces generales consisten en la posibilidad de colisiones por el intento de uso de una etapa por más de una tarea.<sup>15</sup> El control de las colisiones se hace más complejo en las tareas no lineales.

Estos conflictos disminuyen cuando se realiza una buena planeación de los recursos y esto hace que no haya colisiones, sin embargo, lo que es muy difícil evitar es la presencia de las latencias.

En los cauces generales se requiere de una tabla de reserva para cada función. La tabla 1 es un ejemplo de una tabla de reserva del cauce, ésta contiene las etapas del cauce contra el periodo de tiempo que requiere la función para terminar.

$S_j \backslash T_i$	0	1	2	3	4	5	6	7
1	x							x
2		x	x				x	
3				x				
4					x			
5						x		

Tabla 1. Ejemplo de una tabla de reserva del cauce.<sup>16</sup>

Observemos la tabla, una x en la intercepción  $S_j$  con  $T_i$ , nos indica si la etapa  $S_j$  es ocupada por la función F en el ciclo  $T_i$ .

Acerca de las tablas de reserva iniciaremos dando una breve descripción. Podemos decir que son un instrumento o herramienta que nos sirve para identificar y solucionar los riesgos estructurales (ver tabla 2) de una

<sup>15</sup> <http://www.snip.gob.ni/Xdc/amc3/Folleto.pdf>

<sup>16</sup> Fuente: <http://www.snip.gob.ni/Xdc/amc3/Folleto.pdf>

forma detallada, ya que en las columnas de la matriz se mapean las etapas del cauce, y en las filas las instrucciones del segmento.

Tipo de recurso	Riesgos	Solución
Sistema de memoria	Acceso a memoria BUS y MEM.	Usar dos memorias caches.
Recursos de cálculo	Conflicto uso ALU: BUS para incrementar PC, ALU para operar A y B y ALU para comparar.	Usar un sumador en BUS. Usar un comparador en ALU.
Latches	Sobreescritura de latches Colisión acceso a escritura simultánea.	Duplicar latches. El PC es caso especial: se multiplexa.
Instrucción en etapa	Perder la información de qué instrucción se está ejecutando en cada etapa.	Crear réplicas del IR en cada etapa excepto ESC para propagar la instrucción junto con sus datos.

Tabla 2. Resumen de riesgos estructurales con sus soluciones.<sup>17</sup>

Recordemos que en la clasificación de los procesadores segmentados revisamos los cauces unifunción; es importante decir que en este tipo de cauces los riesgos estructurales se resuelven duplicando los recursos de hardware que sean necesarios, sin embargo en los cauces multifunción, los riesgos que nos encontramos frecuentemente son los relacionados con la memoria, de hecho, como lo podemos observar en la tabla 2, este tipo de riesgo lo encontramos en primer lugar de la lista.

Los riesgos estructurales son sólo una parte de la clasificación que existe en la segmentación encauzada, también existen los riesgos por dependencia de datos y por dependencia de control.

Los riesgos estructurales son conflictos o colisiones que se generan al querer ejecutar más de dos etapas en un mismo recurso simultáneamente.

<sup>17</sup> Fuente: <http://www.snip.gob.ni/Xdc/amc3/Folleto.pdf>

Los riesgos por dependencia de datos surgen cuando varias tareas acceden al mismo tipo de datos.

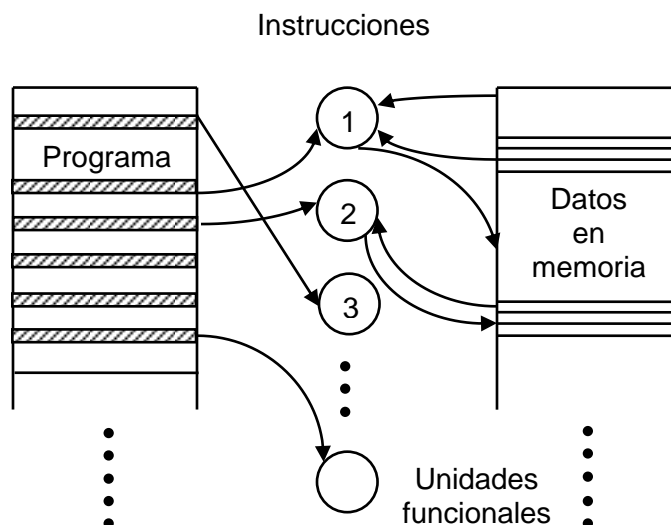
Por último, los riesgos de control surgen de la segmentación de los saltos que generan las instrucciones no leídas, es decir, existen instrucciones que primero son calculadas por la propia instrucción de control de flujo, y ya que se conoce la dirección ésta puede ser leída.

## ACTIVIDAD DE APRENDIZAJE

Elabora un mapa conceptual sintetizando en él la información sobre la tabla de reserva y los tipos de riesgos.

### 2.4 CAUCES DE INSTRUCCIONES Y CAUCES ARITMÉTICOS

Como revisamos anteriormente, los cauces de instrucción suelen ser cauces lineales. La figura 4 muestra un diagrama de cauce por instrucción, en este ejemplo podemos observar cómo las fases de un programa pueden subsegmentarse y trabajar de manera simultánea; recordemos que esto nos permite incrementar la ejecución de un procesador ya que se ejecutan varias instrucciones a la vez.



• Figura 4. Cauce por instrucción.<sup>18</sup>

<sup>18</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

También podemos observar cómo se realizan las búsquedas de instrucciones en memoria, de esa forma son seleccionadas y extraídas para ser ejecutadas, se logra un procesamiento traslapado de varias instrucciones diferentes.

El cauce por instrucciones, en este enfoque segmentado se divide en cinco etapas.

- Búsqueda de instrucción. Se realiza una búsqueda en la memoria y la instrucción que será ejecutada es extraída.
- Decodificación de la instrucción. Se verifica el tipo de instrucción que será ejecutada.
- Búsqueda de operandos. Se realiza una búsqueda, regularmente en registros y se toman los operandos requeridos.
- Ejecución de la instrucción. Se realiza la ejecución del código de operación de la instrucción que será ejecutada.
- Almacenamiento de resultados. Se refiere a la actualización de registros.

Acerca del tipo de segmentación de cauce aritmético, es importante mencionar que es de gran utilidad para ejecutar instrucciones vectoriales, así como algunas otras operaciones de la ALU. Las operaciones a las que se hace referencia pueden ser de tipo lineal, no lineal o cíclicas, por ejemplo una suma, una división o un bucle, como corresponde.

Hoy en día, el encauzamiento aritmético se emplea en un sinnúmero de procesadores que incluyen varias ALU segmentadas, con el fin de ejecutar operaciones aritméticas con números representados en punto flotante.

## ACTIVIDAD DE APRENDIZAJE

En base al diagrama utilizado en la figura 5 explica con tus propias palabras el cauce de instrucciones.

## 2.5 PRINCIPIOS DE DISEÑO DE LOS PROCESADORES ENCAUZADOS

Es elemental realizar un buen diseño de los procesadores, ya que continuamente nos encontramos con problemas graves acerca de su construcción y diseño; existen estrategias de prebúsqueda de instrucciones y control de bifurcaciones con la que se puede mejorar el diseño de dichos procesadores.

Iniciemos con la prebúsqueda de ejecución de instrucciones para su proceso por segmentación encauzada. La clasificación se puede dar en cuatro tipos: de carga aritmética, de almacenamiento, de bifurcación y de bifurcación condicional. El tipo de carga aritmética se refiere principalmente a la manipulación de datos que requieren búsquedas de operandos. El tipo de almacenamiento requiere acceso a la memoria para almacenar los datos. En la bifurcación condicional existen dos caminos, uno calcula una nueva dirección a la que se va a bifurcar, y el segundo prosigue con la instrucción en secuencia del programa.

En las figuras 5 y 6 podemos observar cómo el cauce de una instrucción realiza la ejecución del flujo de instrucciones en modo solapado, en la primera figura sin tener una sola interrupción llamada instrucción de bifurcación, y en la segunda el cauce completa la ejecución con una latencia fija por uno o dos periodos de reloj.

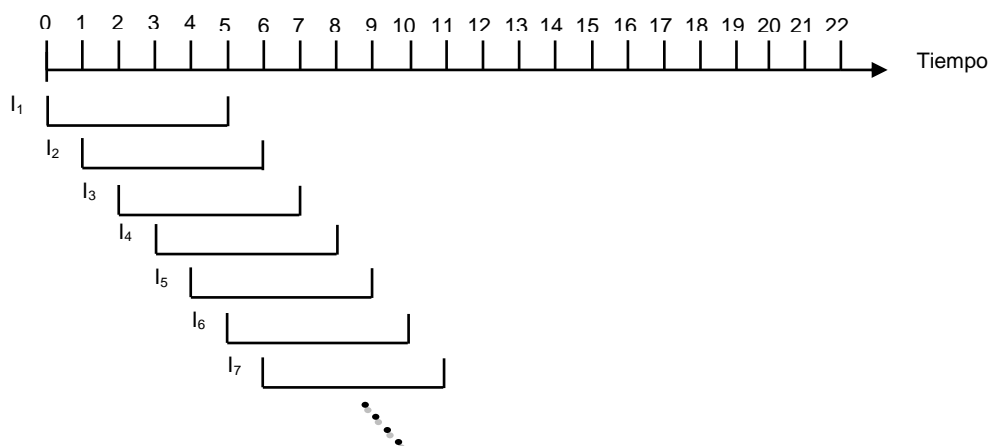


Figura 5. Ejecución solapada de instrucciones sin bifurcación.<sup>19</sup>

<sup>19</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

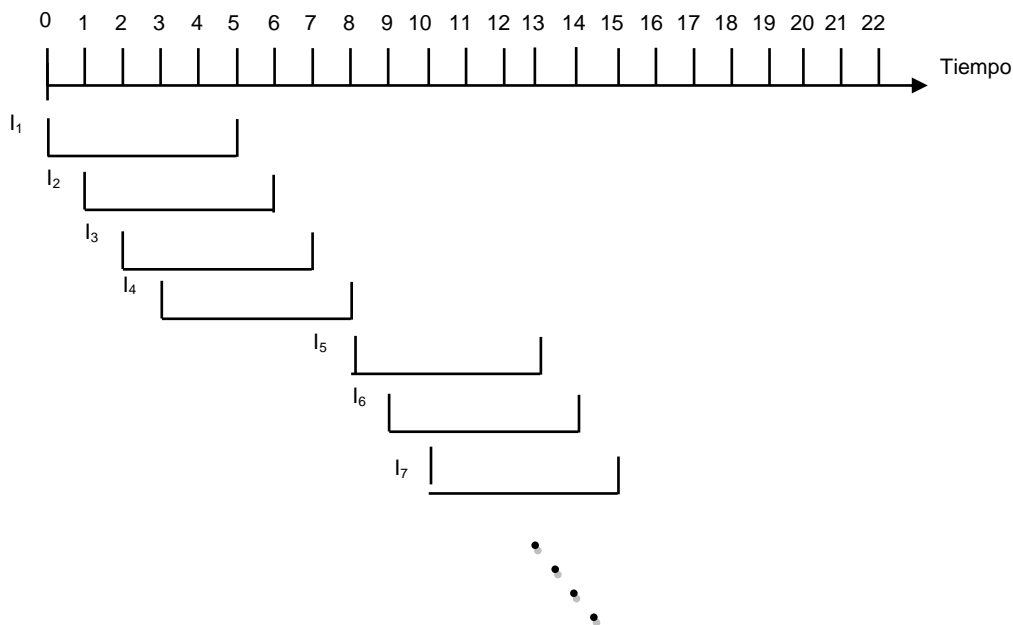


Figura 6. Ejecución solapada de instrucciones con bifurcación.<sup>20</sup>

## ACTIVIDAD DE APRENDIZAJE

Dar seguimiento al mapa conceptual sintetizando la información ya adquirida, incluyendo clasificación y diseño de los procesadores encauzados.

### 2.6 EXIGENCIAS DEL PROCESAMIENTO VECTORIAL

Los procesadores vectoriales son arquitecturas que tienen cauces para procesamiento de vectores. Están diseñados para optimizar las operaciones con estructuras vectoriales y estas operaciones radican en funciones aritméticas y lógicas que son aplicadas a los vectores.

Un vector es una secuencia de datos del mismo tipo, los cuales son almacenados en memoria.

Las exigencias del procesamiento vectorial son propias para la ejecución segmentada, básicamente porque evitamos el riesgo de control del salto del bucle.

El procesamiento vectorial consiste en los diferentes tipos de operaciones vectoriales: de complemento, sumatoria de elementos, multiplicación de vectores o matrices, escalamiento del vector, desplazamiento

<sup>20</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

y producto de punto de vectores, operaciones de comprensión de vectores, operación de expansión de vectores y operación de enmascaramiento de vectores.

Las operaciones de enmascaramiento de vectores son de gran utilidad en el tratamiento de matrices dispersas, ya que son matrices con la mayoría de los elementos nulos. En las operaciones vectoriales cada resultado es independiente de los anteriores.

Los procesadores vectoriales ofrecen grandes ventajas, entre ellas proporcionan un buen aprovechamiento del paralelismo de datos con un control relativamente sencillo. Otra ventaja es que una sola instrucción vectorial sustituye a muchas instrucciones escalares, reduciendo la necesidad de ancho de banda de instrucciones y, finalmente, optimiza el uso de la memoria con accesos predecibles que se pueden solapar. Por estas ventajas las operaciones vectoriales pueden ejecutarse de forma mucho más rápida.

En la figura 7 se observa el esquema de la realización de operaciones vectoriales con registro vectorial y registro escalar.

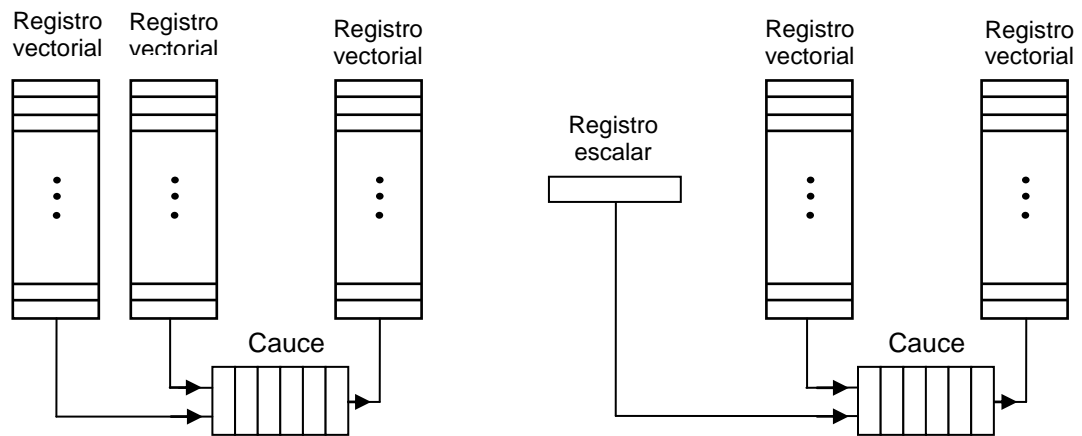


Figura 7. Operaciones con registro vectorial y escalar.<sup>21</sup>

## ACTIVIDAD DE APRENDIZAJE

Elaborar un mapa conceptual con la información vista en la unidad 2, segmentación encauzada y procesamiento vectorial.

<sup>21</sup> Fuente: <http://atc2.aut.uah.es/~acebron/cap3vect.pdf>



## AUTOEVALUACIÓN

11. Observa la siguiente tabla y relaciona cada una de las características de la segmentación encauzada lineal y no lineal; marca con una **X** sólo uno de los incisos con la respuesta que establezca la relación correcta.

Tipo de segmentación	Características
	a) Posibilidad de tener saltos hacia atrás.
4. Lineal.	b) Existe orden de precedencia en las tareas.
5. No lineal.	c) Sólo tiene un punto de entrada y uno de salida.
	d) Tiene varios puntos de entrada y de salida.

a) 1a; 1c; 2d; 2b.

b) 1c; 1b; 2a; 2d.

c) 1b; 1a; 2d; 2c.

d) 1d; 1a; 2b; 2c.

12. Con base en los conocimientos adquiridos en esta unidad, explica con tus propias palabras ¿cuál es la característica principal de la segmentación encauzada lineal?

\_\_\_\_\_

13. Completa el siguiente concepto escribiendo la palabra que falte en el espacio correspondiente.

\_\_\_\_\_ es una secuencia de datos del mismo tipo, los cuales son almacenados en memoria.

14. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

\_\_\_\_\_ es una herramienta que nos sirve para identificar y solucionar riesgos estructurales.

a) Segmentación de instrucción.

b) Cauces unifunción.

c) Segmentación aritmética.

d) Tabla de reserva.

15. Observa la siguiente tabla y relaciona cada uno de los tipos de tipo de riesgos con la descripción que corresponda, marca con una **X** sólo uno de los incisos con la respuesta, que establezca la relación correcta.

<b>Tipo de riesgo</b>	<b>Descripción</b>
1. Riesgos estructurales.	a) Son conflictos que se generan al querer ejecutar varias etapas al mismo tiempo.
2. Riesgos por dependencia de datos.	b) Surgen de la segmentación de los saltos que generan las instrucciones no leídas.
3. Riesgos de control.	c) Surgen cuando varias tareas acceden al mismo tipo de datos.

a) 1a; 2b; 3c.

b) 1a; 2c; 3b.

c) 1b; 2a; 3c.

d) 1c; 2b; 3a.

16. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

Existen varios criterios de clasificación de los procesadores encauzados, ¿Qué propiedades tienen, según su capacidad?

a) Unifunción y multifunción.

b) Estáticos y dinámicos.

c) Lineales y no lineales.

d) Aritmético y de instrucción.

17. Observa la siguiente tabla y relaciona las etapas de los cauces de instrucciones con la descripción que corresponda, marca con una **X** sólo uno de los incisos con la respuesta, que establezca la relación correcta.

<b>Etapas</b>	<b>Descripción</b>
	a) Realiza una búsqueda en registros y se toman los operandos requeridos.

1. Decodificación de la instrucción.	ca) Realiza una actualización a los registros.
2. Búsqueda de operandos.	cb) Realiza una búsqueda en memoria y la instrucción a ejecutar es extraída.
3. Ejecución de la instrucción.	cc) Verifica el tipo de instrucción que será ejecutada.
	cd) Realiza la ejecución del código de operación de la instrucción que será ejecutada.

a) 1e; 2b; 3d.

b) 1c; 2b; 3b.

c) 1d; 2a; 3e.

d) 1b; 2c; 3a.

18. Con base en lo estudiado en esta unidad, describe con tus propias palabras que es una colisión.

## RESPUESTAS

1. b.

2. Las unidades funcionales tienen orden precedente en la ejecución de las tareas.

3. Vector

4. d.

5. b.

6. a.

7. c.

8. Es el uso simultáneo de varios segmentos por parte de ejecuciones distintas de la misma o varias funciones..

## UNIDAD 3

### COMPUTADORES DE SEGMENTACIÓN ENCAUZADA Y MÉTODOS DE VECTORIZACIÓN

#### OBJETIVO

El alumno comprenderá y describirá las bases de la segmentación encauzada y los métodos de vectorización y optimización.

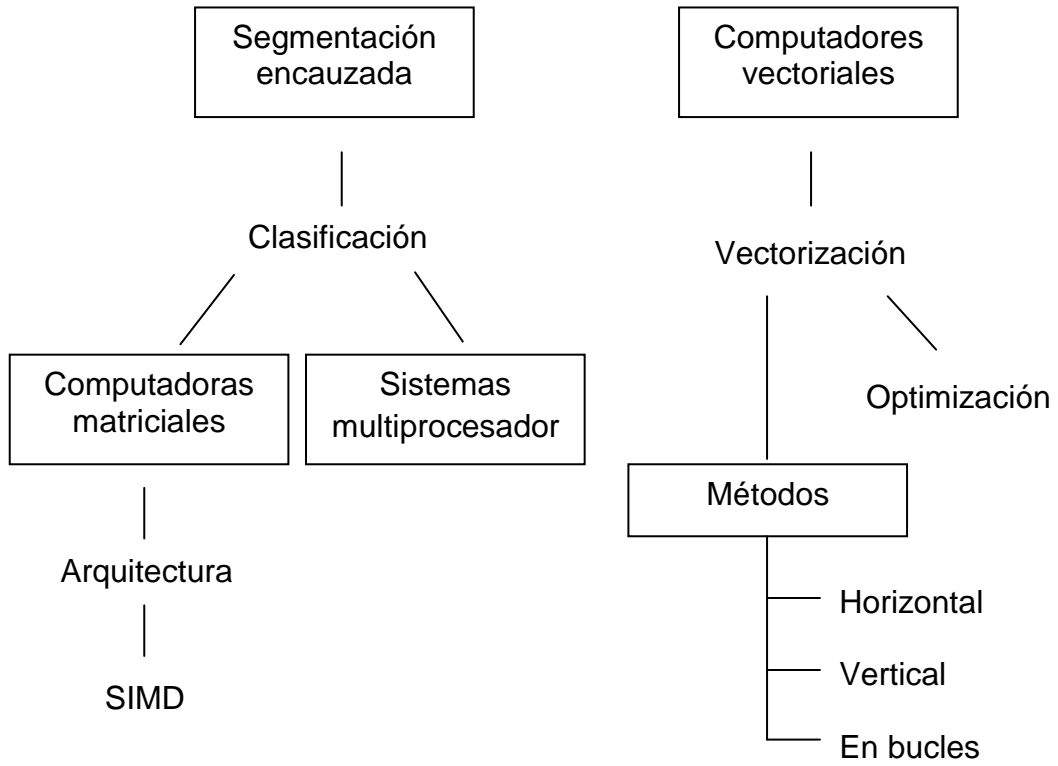
#### TEMARIO

3.1 LOS COMPUTADORES DE SEGMENTACIÓN ENCAUZADA

3.2 PROCESADORES VECTORIALES INICIALES Y ACTUALES

3.3 MÉTODOS DE VECTORIZACIÓN Y OPTIMIZACIÓN

# MAPA CONCEPTUAL



## INTRODUCCIÓN

En esta unidad reconoceremos los computadores que se encuentran dentro de la clasificación de la segmentación encauzada, así como sus características y arquitectura que los constituye.

Durante este curso, a partir del procesamiento paralelo y sus aplicaciones estamos revisando y encauzando el aprendizaje hacia la segmentación, nos daremos cuenta cómo influye y provoca mejora de recursos en los computadores matriciales y sistemas multiprocesadores.

En temas anteriores revisamos los fundamentos del procesamiento vectorial, lo cual nos servirá para que en esta unidad aprendamos los distintos métodos de vectorización y optimización en los computadores de segmentación encauzada.

Los métodos de procesamiento vertical, procesamiento horizontal y operaciones vectoriales en bucles son los que nos ayudan a realizar una buena vectorización al trabajar con grandes cantidades de datos donde requerimos ejecutar distintas operaciones vectoriales.

### 3.1 LOS COMPUTADORES DE SEGMENTACIÓN ENCAUZADA

La segmentación encauzada es una técnica que se usa en el diseño e implantación de microprocesadores. ¿Qué ventajas tienen los computadores de segmentación encauzada? Mejoran el rendimiento incrementando la productividad de las instrucciones en lugar del tiempo de ejecución en cada una de ellas, es decir que no reduce el tiempo que tarda una instrucción en ejecutarse, si no que incrementa el número de instrucciones para ser ejecutadas simultáneamente.<sup>22</sup>

En la figura 1 podemos observar que la ejecución de una instrucción está constituida por cuatro etapas, las cuales gracias a la segmentación encauzada, permiten mejorar la velocidad (cuatro veces más rápido) que una arquitectura sin esta tecnología.

En el ejemplo podemos distinguir las etapas mencionadas: 1) traer la instrucción, 2) decodificarla, 3) traer operandos y 4) ejecutar la instrucción. Cada una de estas etapas son independientes y trabajan de manera paralela; en el mismo ciclo de reloj se estaría trayendo la instrucción 1, se está decodificando, trayendo los operandos y terminando de ejecutar la instrucción, esto quiere decir que cuando el cauce se encuentre lleno en cada ciclo de reloj se estaría terminando la ejecución de cada instrucción.

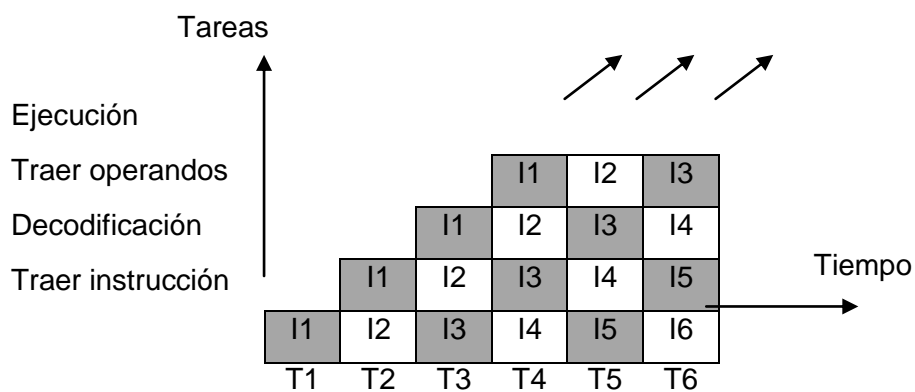


Figura 1. Representación gráfica de tiempo/tareas de una instrucción.<sup>23</sup>

<sup>22</sup> Fuente: <http://www.wisis.ufg.edu.sv/www.wisis/documentos/EB/004.16-S921d/004.16-S921d-Segmentacion%20encauzada.pdf>

<sup>23</sup> Fuente: <http://es.scribd.com/doc/7996920/Diseno-De-Microprocesadores->

La duración de ciclo de reloj es importante en la segmentación encauzada, ya que es la encargada de sincronizar el total de las etapas que constituyen la segmentación.

La figura 2 muestra una estructura de computador típico de segmentación encauzada, está conformado por un cauce aritmético escalar y uno vectorial, podemos observar que la unidad de procesamiento de instrucciones se encuentra encauzada y conformada por tres etapas. La de búsqueda de operandos tiene dos etapas independientes, una para extraer los operandos escalares y otra para extraer los operandos vectoriales.

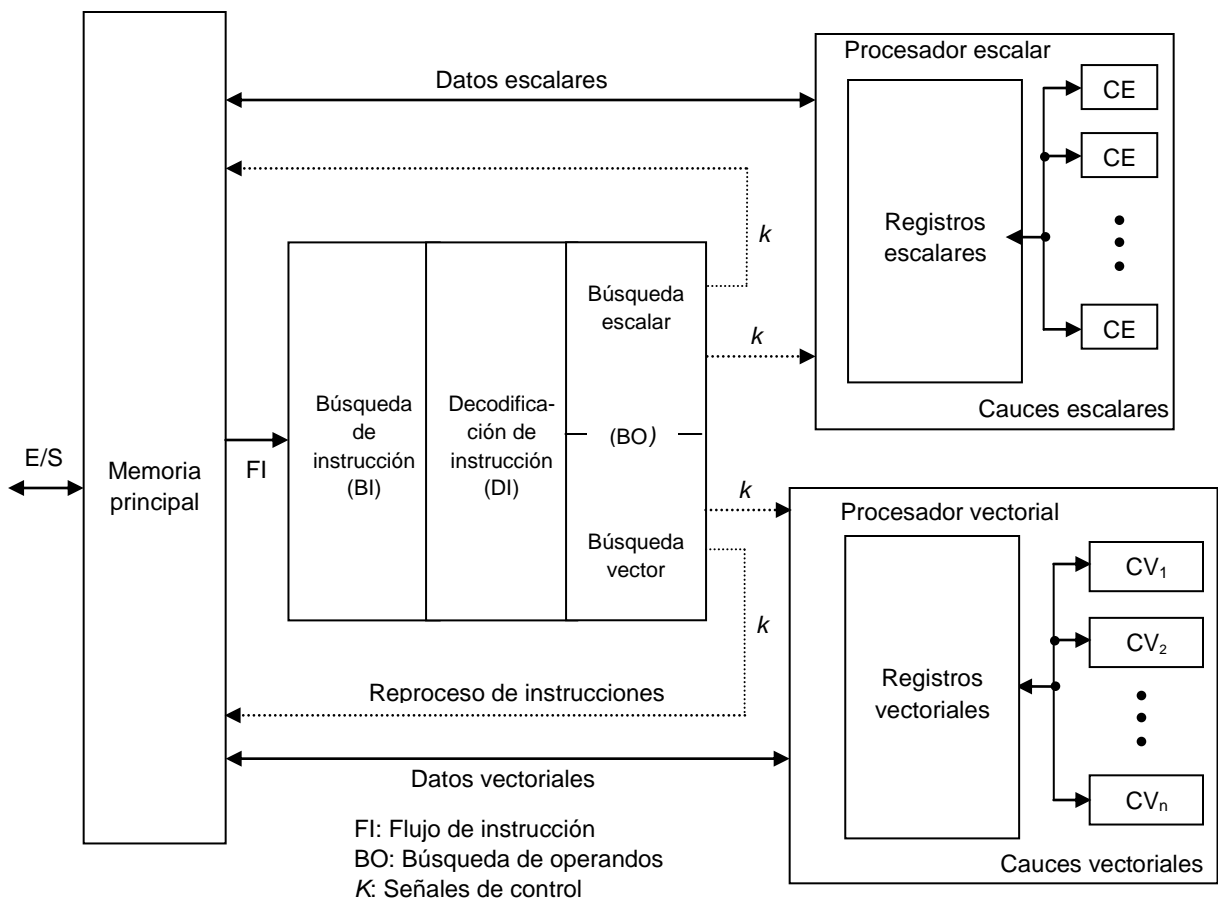


Figura 2. Estructura funcional de un computador encauzado con capacidades escalares y vectoriales.<sup>24</sup>

<sup>24</sup> Fuente: <http://www.wisis.ufg.edu.sv/www.wisis/documentos/EB/004.16-S921d/004.16-S921d-Segmentacion%20encauzada.pdf>



¿Cómo están clasificados los computadores de segmentación encauzada? Los podemos clasificar en computadores matriciales, donde las instrucciones escalares y de control se ejecutan en la unidad de control, encargada de realizar la búsqueda y decodificación de las instrucciones; cada elemento de proceso consta de una ALU con registros y una memoria local.

La arquitectura SIMD (Single-Instruction Stream, Multiple-Data Stream / flujo único de instrucciones, flujo múltiple de datos) corresponde con esta clasificación de computadores. Este tipo de arquitectura es indispensable en los computadores matriciales debido a su gran capacidad de manejo de datos vectoriales y matriciales en poco tiempo. Cuenta con varios procesadores idénticos que poseen una memoria local, trabaja bajo un solo flujo de instrucciones originado por una unidad central de control. La figura 3 muestra el esquema típico de la arquitectura SIMD.

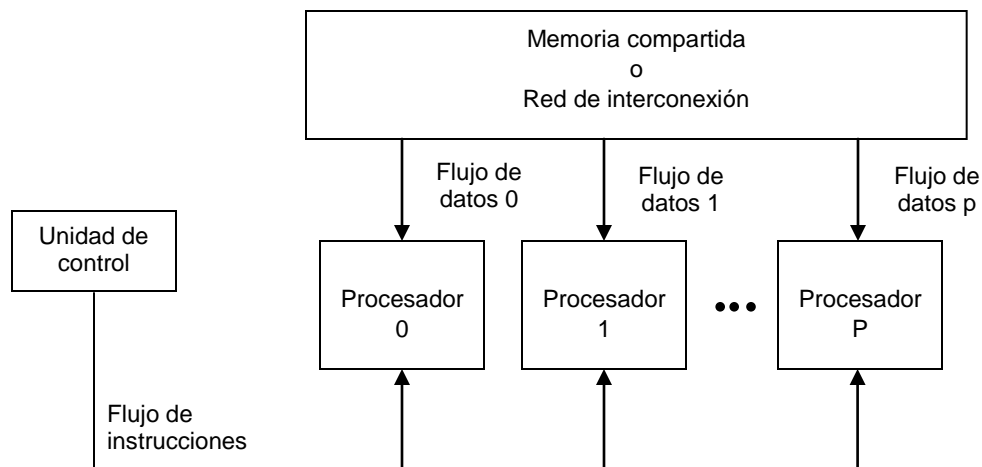


Figura 3. Arquitectura SIMD.<sup>25</sup>

Otra de las clasificaciones son los sistemas multiprocesadores, que consisten en que están orientados a mejorar la productividad, fiabilidad, flexibilidad y disponibilidad de los sistemas. Contienen dos o más procesadores que comparten el acceso común a memoria, canales de entrada y salida y dispositivos periféricos.

<sup>25</sup> Fuente: <http://mixteco.utm.mx/~merg/AC/pdfs/Multiprocesadores.pdf>

## ACTIVIDAD DE APRENDIZAJE

En la figura 2 se muestra la estructura del computador típico de segmentación encauzada, analiza el cauce aritmético escalar y el vectorial y explícalo.

### 3.2 PROCESADORES VECTORIALES INICIALES Y ACTUALES

Los procesadores vectoriales tienen características y exigencias propias para la ejecución segmentada y, como se revisó anteriormente, no cuentan con riesgos estructurales, de datos o control. Este tipo de procesadores son arquitecturas que tienen cauces para procesamiento de vectores.

Algunos procesadores vectoriales aún utilizan instrucciones SIMD, las cuales realizan operaciones enteras y de punto flotante. Como vimos anteriormente, este tipo de arquitectura es extraordinaria por su buen manejo de datos vectoriales y matriciales.

¿Qué computadores vectoriales existieron anteriormente? El Cray-1 de Cray Research, constituido por tres secciones, la de computación, la de memoria y la de entrada-salida. La estructura de la sección de computación es de tres tipos de registros: de dirección, escalares y vectoriales. Dispone de 12 unidades funcionales segmentadas independientes, divididas en cuatro grupos: vectoriales enteras, vectoriales de punto flotante, escalares y de direcciones.

El computador Cyber-205 (figura 4) es otro ejemplo de computador vectorial memoria-memoria, tiene una memoria muy rápida y está enlazada con cuatro vías. La ventaja de que esta máquina sea memoria-memoria radica en que la longitud de los vectores puede ser muy grande. El control de la ejecución se encuentra en la unidad escalar, que recibe y decodifica las instrucciones procedentes de la memoria y las reparte entre los distintos bloques de función de su tipo: escalar, vectorial o de cadena. La unidad de flujo sirve para controlar el tráfico de datos entre la memoria y los cauces.

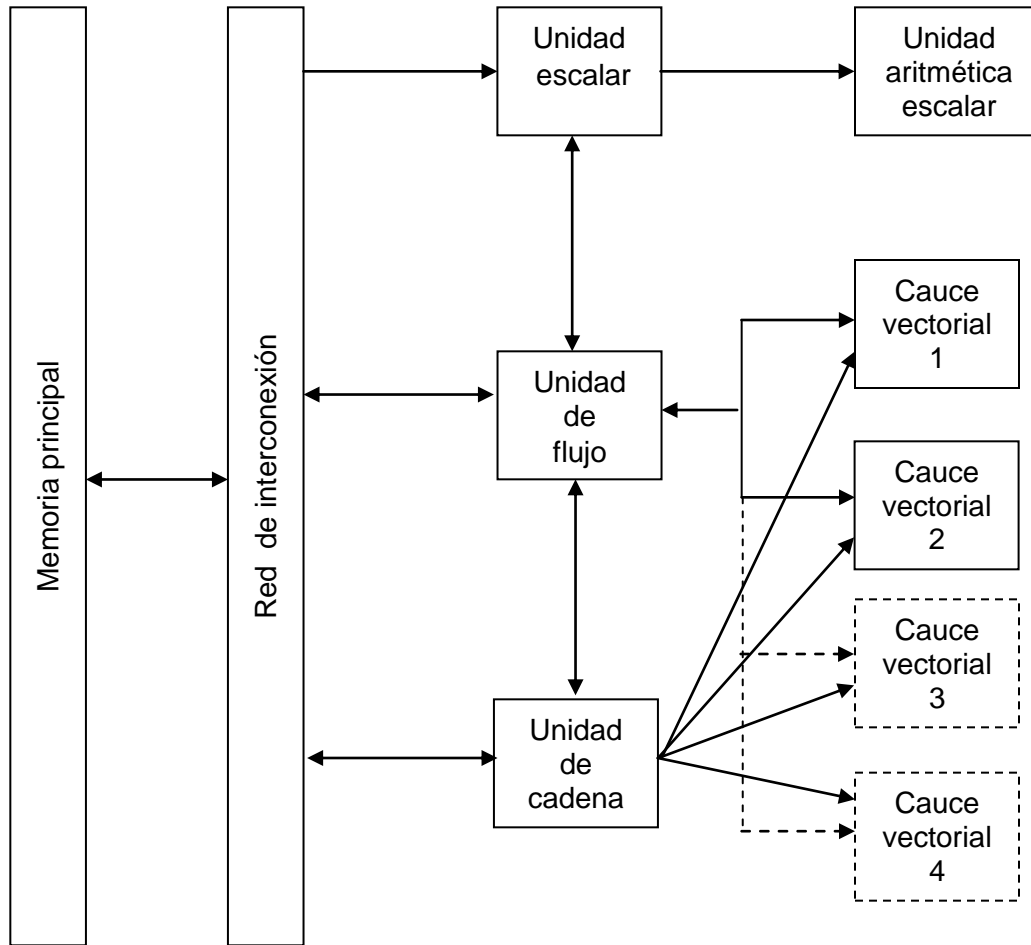


Figura 4. Diagrama general de bloques del Cyber-205.<sup>26</sup>

El computador IBM-3090 (figura 5) es una computadora donde el procesamiento vectorial es una opción llamada *vector facility*, en el diseño de la arquitectura de este equipo la longitud de los registros vectoriales oscila entre 8 y 512. Una de las características más importantes de esta máquina es su juego de instrucciones vectoriales de tipo funciones vectoriales compuestas, ya que esto da mucha agilidad y velocidad a las operaciones sin recurrir al encadenamiento.

<sup>26</sup> Fuente: <http://www.infor.uva.es/~bastida/Arquitecturas%20Avanzadas/Vectoriales.pdf>

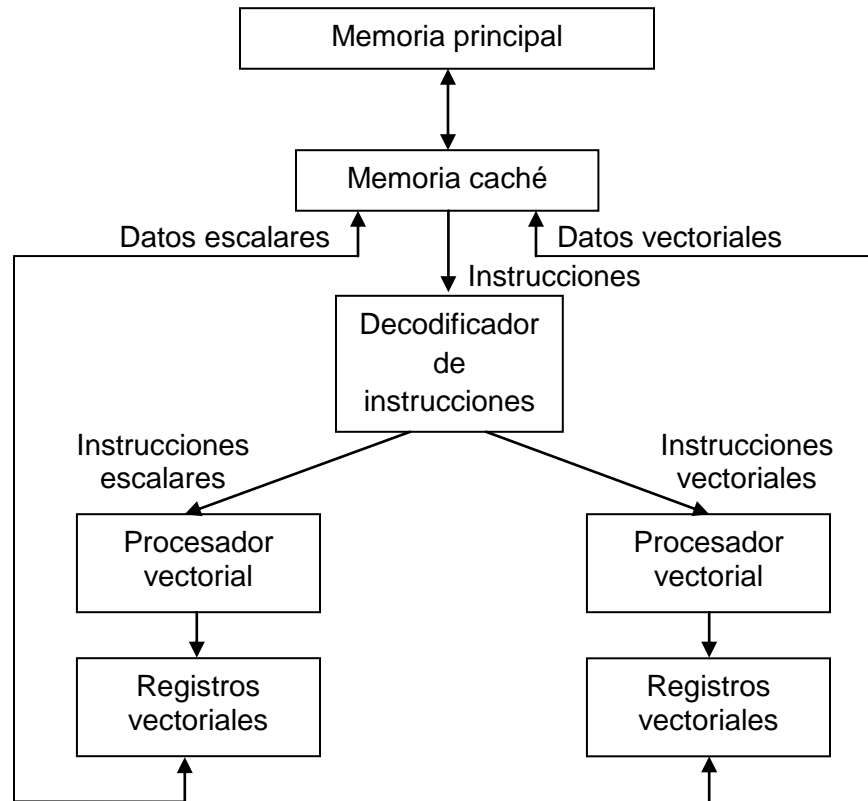


Figura 5. Diagrama de bloques del procesador IBM-3090 con la opción de proceso vectorial.<sup>27</sup>

¿Cómo está construido el computador Earth Simulator? Es un multicomputador construido por 640 nodos interconectados, cada nodo es un multiprocesador con 8 procesadores vectoriales, comparten una memoria de 16 Gigabytes, un procesador de entrada-salida y una unidad de control para el acceso remoto, su unidad de procesamiento superescalar es capaz de emitir cuatro instrucciones por ciclo de reloj. Las unidades superescalares disponen de 128 registros de uso general y cada unidad vectorial incluye 72 registros vectoriales de 256 componentes.

<sup>27</sup> Fuente: <http://www.infor.uva.es/~bastida/Arquitecturas%20Avanzadas/Vectoriales.pdf>

En los últimos años los fabricantes de procesadores como Intel y AMD integraron instrucciones SIMD en los chipsets. El chipset o circuito integrado auxiliar es un conjunto de circuitos integrados diseñado con base en la arquitectura del procesador y en algunos casos se diseña como parte integral de la arquitectura (figura 6).



Figura 6. Ejemplo de un chipset Intel.<sup>28</sup>

A partir del 2011 los procesadores actuales como Sandy Bridge de Intel, integran instrucciones AVX (extensiones vectoriales avanzadas), que están enfocadas básicamente en mejorar operaciones de cálculo, entre las ventajas está el incremento de x2 de operaciones de punto flotante por segundo (FLOP) y mejorar y organizar el acceso a la información.

Las instrucciones AVX vienen a reemplazar las instrucciones SSE (Streaming SIMD / Single Instruction, Multiple Data-Extensions), las cuales fueron desarrolladas por Intel a finales de los años noventa e integradas en los procesadores de la misma marca. A lo largo de casi 10 años, Intel ha realizado mejoras en la diferentes versiones de SSE que se han ido introduciendo en los diferentes procesadores como Intel Core 2 Duo, Intel Core i7 y Xeon 5500, todos basados en la arquitectura Nehalem.<sup>29</sup>

---

<sup>28</sup> Fuente: [http://i258.photobucket.com/albums/hh263/kaux/2984101741\\_ca7e7a109f\\_o.jpg](http://i258.photobucket.com/albums/hh263/kaux/2984101741_ca7e7a109f_o.jpg)

<sup>29</sup> Fuente: <http://www.madboxpc.com/amd-soportara-instrucciones-avx-de-intel/>

## ACTIVIDAD DE APRENDIZAJE

Los procesadores vectoriales no cuentan con riesgos estructurales, de datos o de control. Analiza las complicaciones y discútelas con tu grupo..

### 3.3 MÉTODOS DE VECTORIZACIÓN Y OPTIMIZACIÓN

En la medida que los computadores crecían en potencia y versatilidad, se tuvo que pensar en una forma más eficiente de realizar las operaciones, a este proceso se le llamó vectorización. Hacer una conversión de una operación escalar a una vectorial significa que en la escalar se toma elemento por elemento para finalizar con el proceso y en una vectorial se toman bloques mayores en la memoria y se realiza la operación de forma rápida y eficiente. En la figura 7 podemos observar la utilidad de usar la vectorización, ya que para calcular la operación  $Y = A * X + Y$ , se muestran las instrucciones a seguir desde un programa escalar contra las instrucciones desde un programa vectorial. A simple vista podemos observar que los pasos que tiene el programa vectorial son menos, ya que el proceso no se realiza con todos los elementos, sino por medio de una agrupación en bloques, de manera que la operación se vuelve más sencilla, al seguir menos pasos para finalizar con el procedimiento.

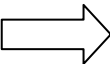
Programa escalar		Programa vectorial
<pre>ld    fo,a addi  r4, rx, #512 Loop:ld  f2, 0 (rx) multd f2, f0, f2 ld    f4, 0 (ry) add   f4, f2, f4 sd    f4, 0 (ry) addi  rx, rx, #8 addi  ry, ry, #8 sub   r20, r4, rx bnz   r20, loop</pre>		<pre>ld    fo,a lv    v1, rx multsv v2, f0, v1 lv    v3, ry addv  v4, v2, v3 sv    ry, v4</pre>

Figura 7. Ejemplo del uso de la vectorización.<sup>30</sup>

Ahora que tenemos claros conceptos generales sobre la vectorización, podremos revisar los métodos con los cuáles se puede realizar. Con frecuencia los usamos al procesar grandes cantidades de datos donde se tienen que realizar cálculos vectoriales. Estos métodos los podemos clasificar en tres tipos:

1. Procesamiento horizontal. Los cálculos vectoriales se tienen que realizar horizontalmente, de izquierda a derecha y por filas.
2. Procesamiento vertical. Los cálculos se tienen que realizar verticalmente, de arriba hacia abajo y por columnas.
3. Operaciones vectoriales en bucles. Los cálculos se realizan de izquierda a derecha y de arriba hacia abajo; es una combinación de los procesamientos horizontal y vertical.<sup>31</sup>

Existen varios sistemas de optimización utilizados en los procesadores vectoriales, Mem-bancos ORY es usado para reducir la carga y la latencia, opencast mining (minería a cielo abierto) se utiliza para generar y optimizar código usado para la operación vectorial, el encadenamiento de vector, ayuda a duplicar el número de operaciones en punto flotante que se realizan en las unidades de tiempo.

## ACTIVIDAD DE APRENDIZAJE

Realizar un cuadro sinóptico estructurando y sintetizando la información de los métodos de vectorización y optimización.

---

<sup>30</sup> Fuente: <http://atc2.aut.uah.es/~nacho/All/tema6.2.pdf>

<sup>31</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

## AUTOEVALUACIÓN

19. Lee cuidadosamente y completa la siguiente oración:

La segmentación encauzada no reduce el tiempo que tarda una instrucción en ejecutarse, sino que \_\_\_\_\_.

20. Completa el siguiente concepto escribiendo la palabra que falte en el espacio correspondiente:

La arquitectura \_\_\_\_\_ cuenta con más de dos procesadores idénticos que poseen una memoria local, trabaja bajo un solo flujo de instrucciones originado por una unidad central de control.

21. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

\_\_\_\_\_ están orientados a mejorar la productividad, la fiabilidad, la flexibilidad y la disponibilidad de los sistemas.

- a) Los sistemas multiprocesadores.
- b) Los métodos de vectorización.
- c) Los métodos de optimización.
- d) Los computadores vectoriales.

22. ¿Qué son los chipset?

23. En los procesadores vectoriales actuales se integran las instrucciones AVX (extensiones vectoriales avanzadas), sus principales ventajas son:

24. Observa la siguiente tabla y relaciona cada uno de los métodos de vectorización con la descripción que corresponda, marca con una **X** sólo uno de los incisos con la respuesta que establezca la relación correcta.

Método	Descripción
4. Procesamiento vertical.	d) Los cálculos vectoriales se realizan de



	izquierda a derecha y por filas.
5. Procesamiento horizontal.	e) Los cálculos vectoriales se realizan de izquierda a derecha y de arriba hacia abajo.
6. Operaciones vectoriales en bucles.	f) Los cálculos vectoriales se realizan de arriba hacia abajo y por columnas.

a) 1a; 2c; 3b.

b) 1a; 2b; 3c.

c) 1b; 2a; 3c.

d) 1c; 2a; 3b.

25. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

Dentro de la clasificación de los procesadores vectoriales podemos encontrar aquel del que gracias al juego de instrucciones vectoriales se obtiene agilidad y velocidad de las operaciones sin recurrir al encadenamiento. ¿A qué procesador se hace referencia?

a) IBM-3090.

b) Earth Simulator.

c) Cyber-205.

d) Cray-1.

26. Observa la siguiente tabla y relaciona la columna de sistemas de optimización con el uso que corresponda utilizados en los procesadores vectoriales, marca con una **X** sólo uno de los incisos con la respuesta que establezca la relación correcta.

Sistemas de optimización	Uso
4. Mem-bancos ORY.	a) Genera y optimiza el código usado para la operación vectorial.
5. Opencast mining (minería a cielo)	b) Reduce carga y la latencia.

abierto).	
6. Encadenamiento de vector.	c) Duplica el número de operaciones en punto flotante realizadas en las unidades de tiempo.

e) 1c: 2b; 3a.

f) 1b: 2a; 3c.

g) 1a: 2b; 3c.

h) 1b: 2c; 3a.

27. Lee con atención y contesta la siguiente pregunta:

El sistema H-PP11N es un sistema implementado en una reconocida fábrica textil al norte de la Ciudad de México, el sistema cuenta con 20 procesadores idénticos y cada uno de ellos maneja un sistema de memoria autónomo e independiente, la producción es de alta calidad y debido a esto la importación de telas es superior a los 10 mdp mensuales, la cantidad de datos que se manejan es incuantificable, ¿A qué clase de computadores de segmentación encauzada se hace referencia?

28. Observa la siguiente tabla y relaciona los tipos de procesadores vectoriales y sus características, marca con una **X** sólo uno de los incisos con la respuesta que establezca la relación correcta.

Procesador vectorial	Características
1. Cray-1.	a) Está construido por 640 nodos interconectados, comparten una memoria de 16 Gigabytes, su unidad de procesamiento superescalar emite cuatro instrucciones por ciclo de reloj.
2. Cyber-205.	b) Está constituido por la secciones de computación, memoria y entrada-salida.
3. Earth Simulator.	c) Su principal característica es que tiene una

	memoria muy rápida y está enlazada con cuatro vías.
--	---

- a) 1c; 2b; 3a.
- b) 1b; 2c; 3a.
- c) 1a; 2b; 3c.
- d) 1b; 2a; 3c.

## RESPUESTAS

1. Incrementa el número de instrucciones para ser ejecutadas simultáneamente.
2. SIMD.
3. a.
4. Es un conjunto de circuitos integrados diseñados con base en la arquitectura del procesador.
5. Incrementar las operaciones de punto flotante por segundo (FLOP) y mejorar y organizar el acceso a la información.
6. d.
7. a.
8. b.
9. Computadores matriciales.
10. b.

## UNIDAD 4

### ESTRUCTURAS Y ALGORITMOS PARA PROCESADORES MATRICIALES

#### OBJETIVO

El alumno comprenderá las estructuras y decisiones de diseño de algoritmos para procesadores matriciales, así como la interconexión de los procesadores SIMD.

#### TEMARIO

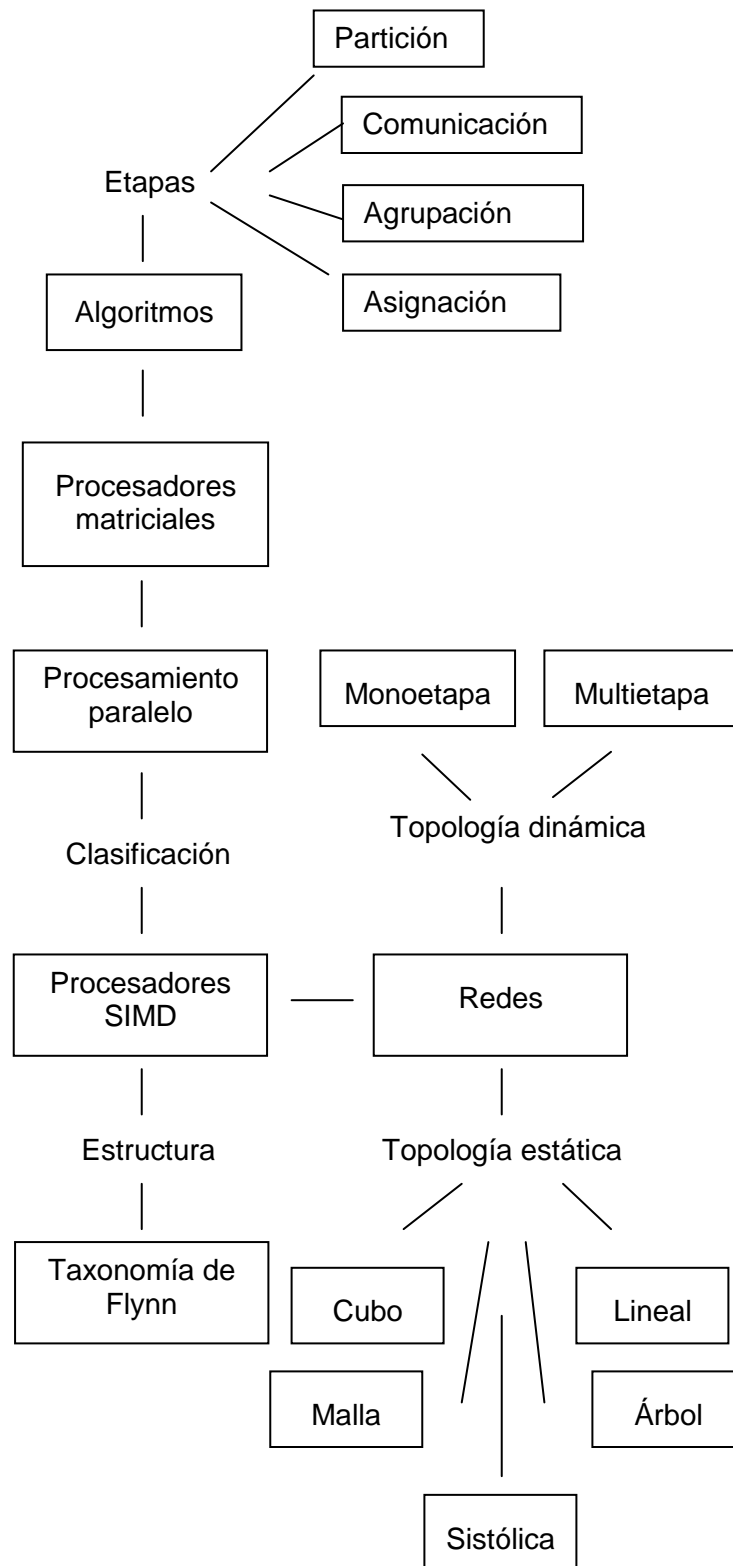
4.1 PROCESADORES SIMD

4.2 REDES DE INTERCONEXIÓN SIMD

4.3 ALGORITMOS PARALELOS PARA PROCESADORES MATRICIALES

4.4 PROCESAMIENTO MATRICIAL ASOCIATIVO

# MAPA CONCEPTUAL



## INTRODUCCIÓN

En esta unidad revisaremos cómo fueron diseñados los procesadores SIMD y conoceremos la clasificación de su arquitectura y las ventajas y desventajas del uso de estas máquinas.

Conoceremos las topologías de las redes de interconexión para la arquitectura, ya que existen varios tipos de éstas que se pueden utilizar para que los procesadores siempre estén conectados con el fin de realizar la transferencia de datos en el momento que se requiera. Debemos saber que las redes de interconexión son uno de los aspectos más significativos dentro de los sectores que se aplican en la selección de la arquitectura.

Es importante conocer los algoritmos con los que se construyen los procesadores matriciales, veremos los diferentes tipos que existen, así como el proceso matricial de tipo asociativo.

En unidades anteriores revisamos conceptos que nos sirven ahora muy bien para comprender estos temas. Con esta unidad se pretende que el alumno comprenda y asocie el aprendizaje adquirido en esta unidad con las unidades ya vistas.

#### 4.1 PROCESADORES SIMD

Anteriormente los diseños de las unidades de control era de muy alto costo, ésta fue la razón principal por la que se empezaron a diseñar los procesadores con arquitectura SIMD. Michel J. Flynn en 1972 propuso una manera de clasificar el procesamiento paralelo; aunque existen varias maneras de hacerlo, él consideró que la organización de un sistema se da mediante la cantidad de instrucciones y unidades de datos que se manipulan en forma simultánea, de esa forma se creó la taxonomía de Flynn.

Uno de los grupos que pertenecen a esta clasificación es el flujo único de instrucciones, flujo múltiple de datos, SIMD (por sus siglas en inglés Single-Instruction Stream, Multiple-Data Stream). Las primeras máquinas paralelas correspondieron a este tipo por ser más simples de construir y a menor costo.<sup>32</sup>

En la unidad 3 revisamos la estructura de la arquitectura SIMD, recordemos que está formada por un gran número de procesadores ejecutando la misma operación sobre un conjunto de datos y bajo la supervisión de una sola unidad de control (CU), que es la encargada de interpretar y distribuir la misma instrucción a todos los procesadores.

Esta estructura representó un cambio en el estilo de la programación consiguiendo resultados especulares de velocidad de proceso, de hecho una de sus ventajas es la habilidad que tiene para manejar grandes vectores y matrices de datos en tiempos muy cortos.

Un arreglo de procesadores se compone de múltiples elementos de procesamiento (PE), éstos son controlados por la unidad de control, la cual es común para todos los procesadores y como resultado de ello se ejecuta la misma operación simultáneamente. La unidad de control decodifica todas las instrucciones y decide dónde se ejecutan. En la figura 1 vemos gráficamente esta estructura.

Cada elemento de procesamiento lo interpretamos como una unidad lógica-aritmética con registros que caen en la memoria local para el almacenamiento de los datos distribuidos, la unidad de control cuenta con una

---

<sup>32</sup> M. Morris Mano, *Arquitectura de computadoras*, México, Pearson Education, 1993, p. 321.

memoria propia para el almacenamiento de programas y aquí también se ejecutan directamente las instrucciones de control de tipo escalares.

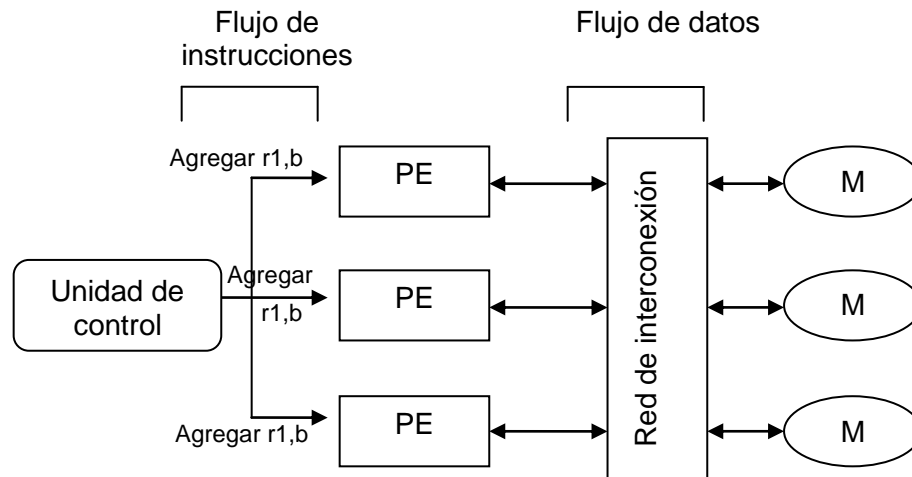


Figura 1. Ejecución SIMD.<sup>33</sup>

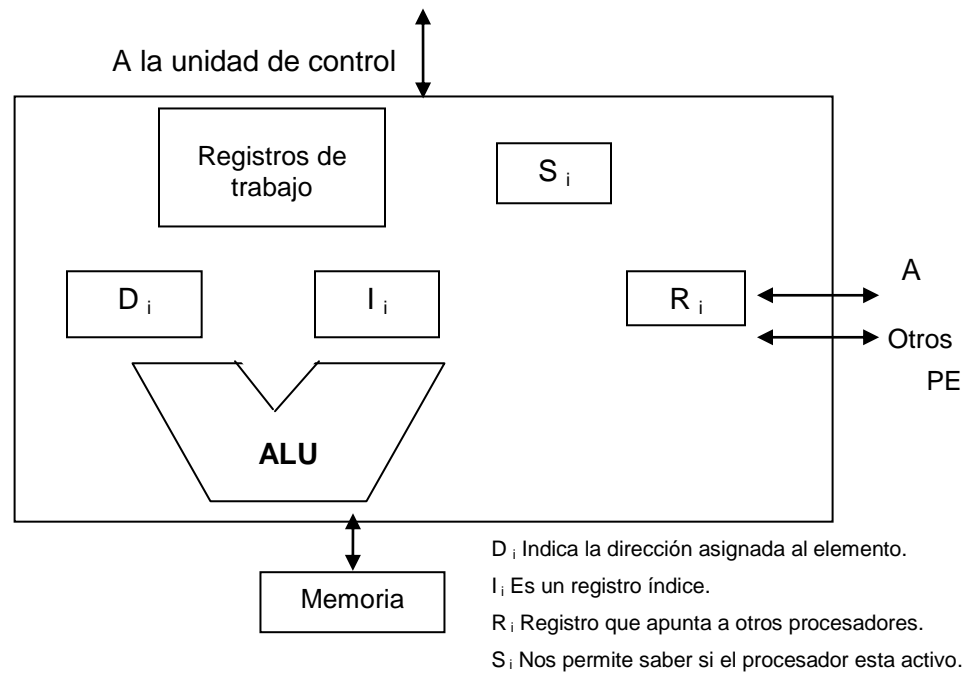


Figura 2. Estructura interna de un elemento de procesamiento (EP).<sup>34</sup>

<sup>33</sup> Fuente: <http://msanchez.usach.cl/lcc/Arquitectura-SIMD.pdf>

<sup>34</sup> Fuente: <http://msanchez.usach.cl/lcc/Arquitectura-SIMD.pdf>



Todos los elementos de procesamiento realizan la misma función sincrónicamente en forma esclava bajo el control de la unidad de control. Veamos la figura 2 y revisemos cada uno de los componentes que lo conforman.

## ACTIVIDAD DE APRENDIZAJE

Consulta la página <http://msanchez.usach.cl/lcc/Arquitectura-SIMD.pdf> e identifica el esquema que se implementó en la computadora ILLIAC IV y expón a tus compañeros el tipo de configuración.

### 4.2 REDES DE INTERCONEXIÓN SIMD

La mayoría de las aplicaciones que usan las máquinas SIMD utilizan redes síncronas, pueden ser estáticas o dinámicas, su control es centralizado y con conmutación de circuitos.

Las redes estáticas emplean enlaces directos fijos entre los nodos. Los nodos son todos aquellos dispositivos que se quieren conectar a la red, tales como elementos de proceso, módulos de memoria, procesadores de entrada/salida, etc. En este tipo de enlaces una vez fabricado el sistema es difícil de cambiar, ya que la escalabilidad es baja. Las principales clases de redes de interconexión estáticas son de formación lineal, anillo, estrella, árbol, malla, cubo, red sistólica, entre otras.

La siguiente figura muestra algunas estructuras gráficas de las redes mencionadas.

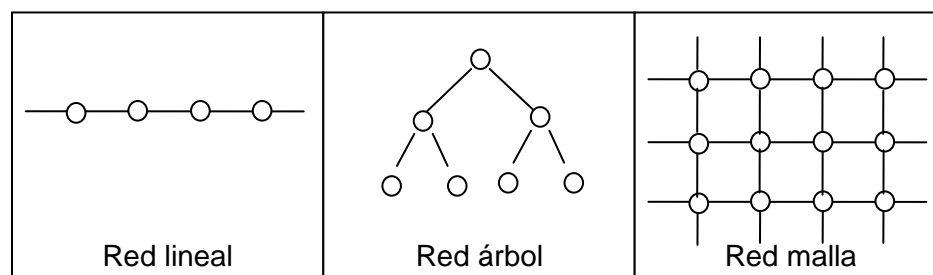


Figura 3. Ejemplos de redes estáticas.<sup>35</sup>

<sup>35</sup> Fuente: <http://atc2.aut.uah.es/~nacho/All/tema6.2.pdf>

Las redes de conexión dinámicas son aquellas cuya topología es fácil de modificar; cuando dos nodos necesitan ser conectados, la red puede cambiar de tal forma que se puedan conectar, esto pasa gracias a que facilitan mucho la escalabilidad. ¿Qué tipos de redes dinámicas existen? Las redes monoetapa son aquellas que realizan conexiones entre elementos de proceso en una sola etapa, en este tipo de redes es importante usar la red de barras cruzadas.

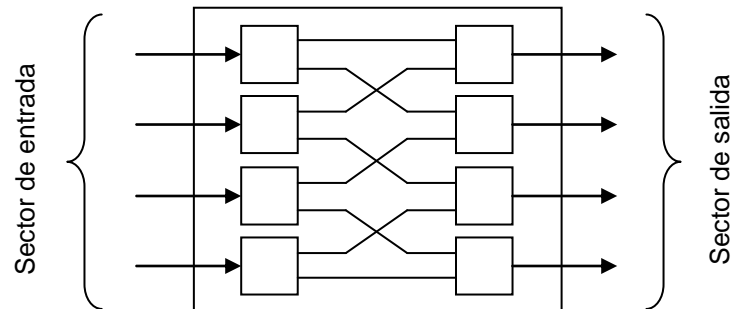


Figura 4. Ejemplo de red monoetapa.<sup>36</sup>

Y las redes multietapa, son aquellas que realizan conexiones entre los elementos de proceso en más de una etapa, capaces de conectar un elemento de proceso arbitrario de entrada a otro arbitrario de salida; se describen mediante cajas de conmutación, topología de red y estructura de control. La figura 5 muestra los cuatro tipos de configuraciones posibles de las cajas de conmutación con dos entradas.

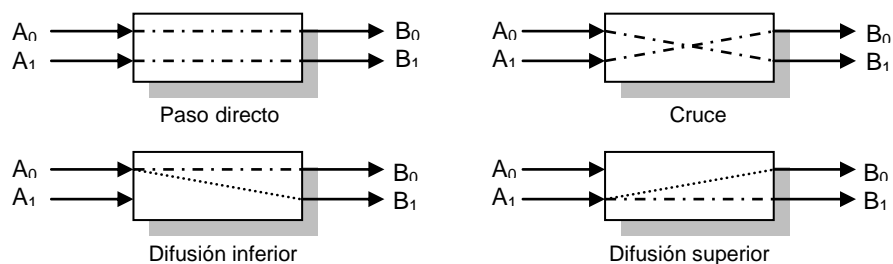


Figura 5. Configuraciones posibles de cajas de conmutación con dos entradas.<sup>37</sup>

<sup>36</sup> Fuente: <http://atc2.aut.uah.es/~nacho/All/tema6.2.pdf>

<sup>37</sup> Fuente: <http://atc2.aut.uah.es/~nacho/All/tema6.2.pdf>

Las redes multietapa tienen una subclasificación que corresponde a 3 grupos, el grupo 1, con bloque, se refiere a la conexión simultánea de varias parejas de terminales para producir conflictos; el grupo 2, reordenable, reordenando la configuración establece una línea de comunicación para una nueva pareja, y el grupo 3, sin bloqueo, como su nombre lo indica, puede manejar todas las conexiones posibles sin que se produzca ningún bloqueo.

## ACTIVIDAD DE APRENDIZAJE

Realizar un cuadro sinóptico con la clasificación de las redes de interconexión SIMD.

### 4.3 ALGORITMOS PARALELOS PARA PROCESADORES MATRICIALES

Los algoritmos los podemos definir como un conjunto de sentencias o instrucciones que expresan la lógica de un programa y nos permiten obtener un resultado determinado. El diseño de estos algoritmos es una tarea muy difícil y se requiere de mucha creatividad. La figura 6 nos muestra las cuatro etapas que involucran el diseño de unos de estos algoritmos.

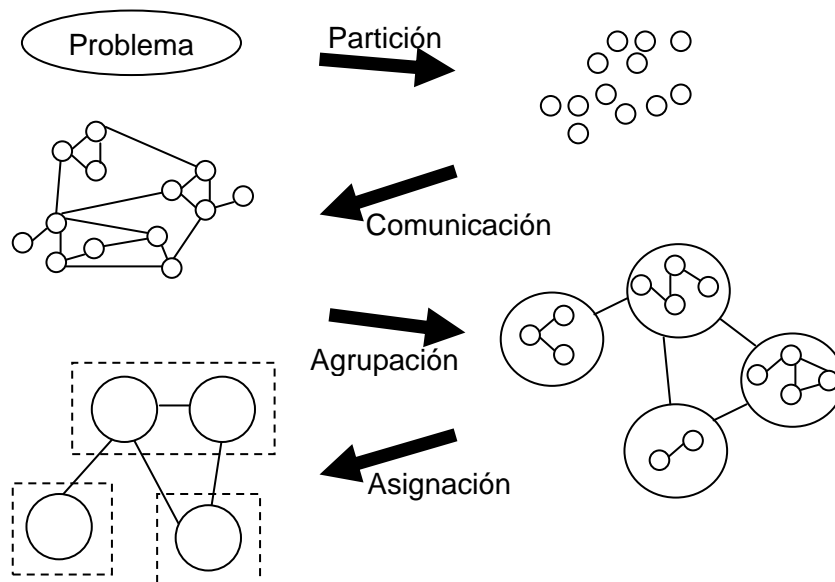


Figura 6. Etapas en el diseño de algoritmos paralelos.<sup>38</sup>

<sup>38</sup> Fuente: [http://www.saber.ula.ve/bitstream/123456789/15969/1/com\\_par.pdf](http://www.saber.ula.ve/bitstream/123456789/15969/1/com_par.pdf)

La primera etapa es la partición, en ella se descomponen el cómputo y los datos en tareas, se trata de subdividir el problema lo más finamente posible, no se toma en cuenta el número de procesadores que la máquina usará y sólo se concentra la atención en buscar oportunidades de paralelismo. Para proceder con la descomposición existen dos formas de hacerlo, la descomposición de dominio donde el centro de atención son los datos y la descomposición funcional donde el centro de atención lo ocupa el cómputo. Es importante que al particionar tareas se tome en cuenta que el número debe ser proporcional al tamaño del problema, de esta manera se logra que el algoritmo sea escalable, es decir, que el grado de paralelismo aumente con el tamaño del problema.

La etapa de comunicación atiende a la relación que se requiere determinar para realizar la coordinación, la transferencia y el compartimiento de cada una de las tareas, es importante saber que las tareas definidas en la etapa anterior pueden correr concurrentemente pero no de forma independiente. En esta etapa se llegan a definir las estructuras y los algoritmos de comunicación, lo cual puede suceder en dos fases: en la primera se definen los canales que conectan las tareas donde se encuentran los datos a compartir y en la segunda fase se especifica el mensaje que debe ser enviado y recibido a través de estos canales. En la mayoría de los computadores paralelos la comunicación se da mediante el pase de mensajes y regularmente los equipos se paran para enviar y recibir mensajes

En la etapa de la agrupación podemos observar que se encuentra el resultado de las dos etapas anteriores, sin embargo el algoritmo resultante es abstracto debido a que aún no se considera en que procesador correrá, así que la etapa de agrupación nos sirve para producir un algoritmo que corra eficientemente sobre una clase de computadores en especial. También se evalúan términos de eficiencia y costos de la implementación, esto quiere decir que al agrupar las tareas se puede reducir la cantidad de datos a enviar y así se reduce la cantidad de mensajes y el costo de la comunicación.

Por último, se encuentra la etapa de la asignación, aquí se determina en qué procesador se ejecutará cada tarea. La asignación puede ser estática o dinámica, en la estática las tareas son asignadas a un procesador al comienzo de la ejecución del algoritmo paralelo y corren ahí hasta el final; y la de tipo dinámico hace cambios en la distribución de las tareas entre los procesadores a tiempo de ejecución, esto se realiza con el fin de balancear la carga del sistema y reducir el tiempo de ejecución, sin embargo el costo de balanceo puede ser significativo e incrementar el tiempo de ejecución. Entre los algoritmos de balanceo de carga se encuentran el balanceo centralizado, el balanceo completamente distribuido y el balanceo semidistribuido.

Las siguientes son características propias de los algoritmos matriciales:

- ✓ Reproducen el comportamiento natural de la estructura de datos.
- ✓ Son algoritmos muy complejos.
- ✓ Tienen una ganancia significativa sobre homólogos secuenciales.

Una de las ventajas de los algoritmos paralelos para procesadores matriciales es que la eficiencia es óptima cuando se manejan arreglos de datos en ciclos for; sin embargo también nos encontramos con algunas desventajas: el rendimiento se compromete en el uso de las sentencias case y en los ciclos while; en algunos procesadores se puede encontrar la condición de salida del ciclo antes de que ocurra en otros procesadores, en este caso los procesadores que hayan completado el ciclo se deben ir deshabilitando hasta que el resto concluya con la condición de salida.<sup>39</sup>

## ACTIVIDAD DE APRENDIZAJE

Consulta [http://www.saber.ula.ve/bitstream/123456789/15969/1/com\\_par.pdf](http://www.saber.ula.ve/bitstream/123456789/15969/1/com_par.pdf) y prueba el código correspondiente al ejemplo de multiplicación de matrices; envía tus comentarios al profesor.

---

<sup>39</sup> Fuente: [http://www.saber.ula.ve/bitstream/123456789/15969/1/com\\_par.pdf](http://www.saber.ula.ve/bitstream/123456789/15969/1/com_par.pdf)

#### 4.4 PROCESAMIENTO MATRICIAL ASOCIATIVO

El procesamiento asociativo es un caso particular en los procesadores matriciales, su principal característica es el uso de la memoria de tipo asociativa (CAM, Content Addressable Memory), donde los datos son direccionables por contenido, permitiendo el acceso a numerosas palabras de la memoria, su nivel de paralelismo se mide en bit- $\{serie|paralelo\}$ , palabra- $\{serie|paralela\}$ , es decir cada palabra de memoria asociativa, que generalmente tiene una gran cantidad de bits, se asocia con registros especiales y una lógica de comparación que funcionalmente constituye un procesador, por tanto, un procesador asociativo con 7098 palabras tiene efectivamente 7098 elementos de procesamiento.

La figura 7 muestra una comparación orientada a fila para una arquitectura genérica de bit-serial.

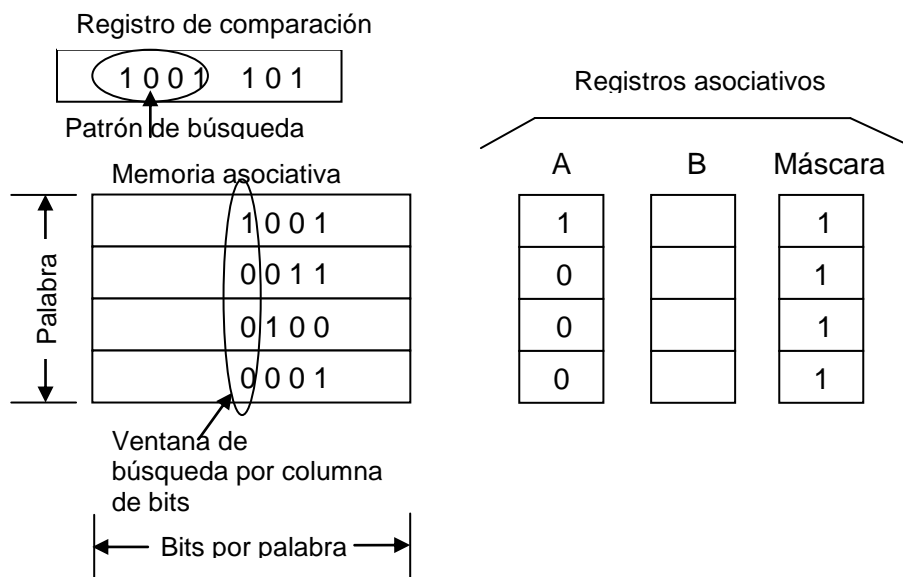


Figura 7. Comparación orientada a fila.<sup>40</sup>

De las primeras arquitecturas que fueron diseñadas con procesamiento asociativo está STARAN y Ensamblaje de Elementos Paralelos de Procesamiento (PEPE), ambos construidos en 1972, el primero construido por Goodyear Aerospace e instalado en varios lugares incluyendo Johnson Space Center de la NASA.

<sup>40</sup> Fuente: <http://msanchez.usach.cl/lcc/Arquitectura-SIMD.pdf>

STARAN es un procesador asociativo bit-serie y de costo moderado. Está construido por 32 módulos asociativos de matrices, cada módulo tiene una memoria de 256 palabras de 256 bits cada una, de acceso multidimensional, una red de permutación y un selector; esta estructura la podemos observar en la figura 8.

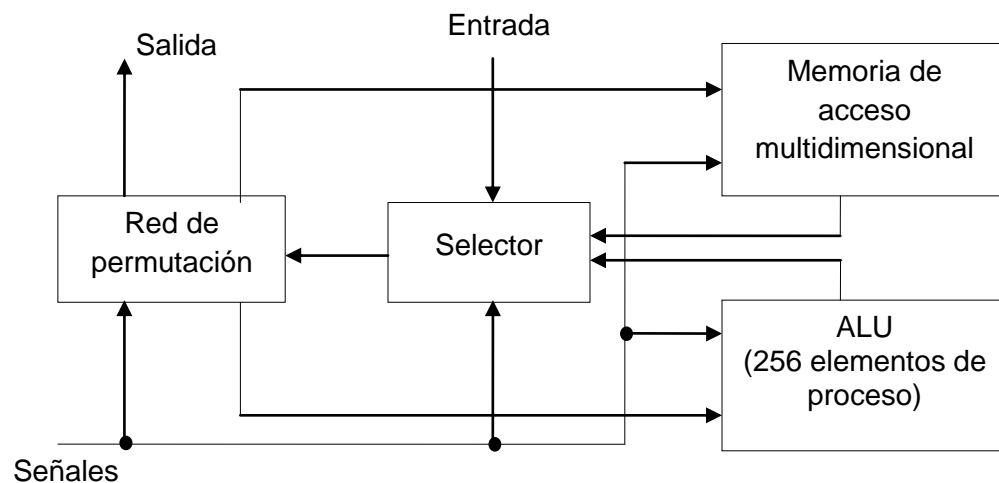


Figura 8. Módulo asociativo de matriz de STARAN.<sup>41</sup>

¿Qué diseño tiene PEPE? Es un procesador matricial asociativo basado en la configuración lógica distribuida, fue desarrollado en Bell Laboratories para aplicaciones de procesamiento de señales de radar. Está diseñado por los subsistemas funcionales, las cuales se muestran en la figura 9. Tiene 288 elementos de proceso organizados en ocho bastidores de elementos, y porciones de la carga de trabajo se cargan desde un computador principal (host) CDC-7600 en los EP. El proceso de selección de carga es determinado por el paralelismo inherente de la tarea y la capacidad de la arquitectura PEPE. Cada elemento de proceso tiene la responsabilidad de un objeto bajo observación por el sistema de radar, mantiene un archivo de datos que por su capacidad aritmética-asociativa actualiza constantemente el archivo específico.<sup>42</sup>

<sup>41</sup> Fuente: <http://www.dte.eis.uva.es/docencia/etsii/smp/bak/tema5/tema5.pdf>

<sup>42</sup> Fuente: [http://www.snip.gob.ni/xdc/amc3/Seminario/PEPE\\_5T1.pdf](http://www.snip.gob.ni/xdc/amc3/Seminario/PEPE_5T1.pdf)

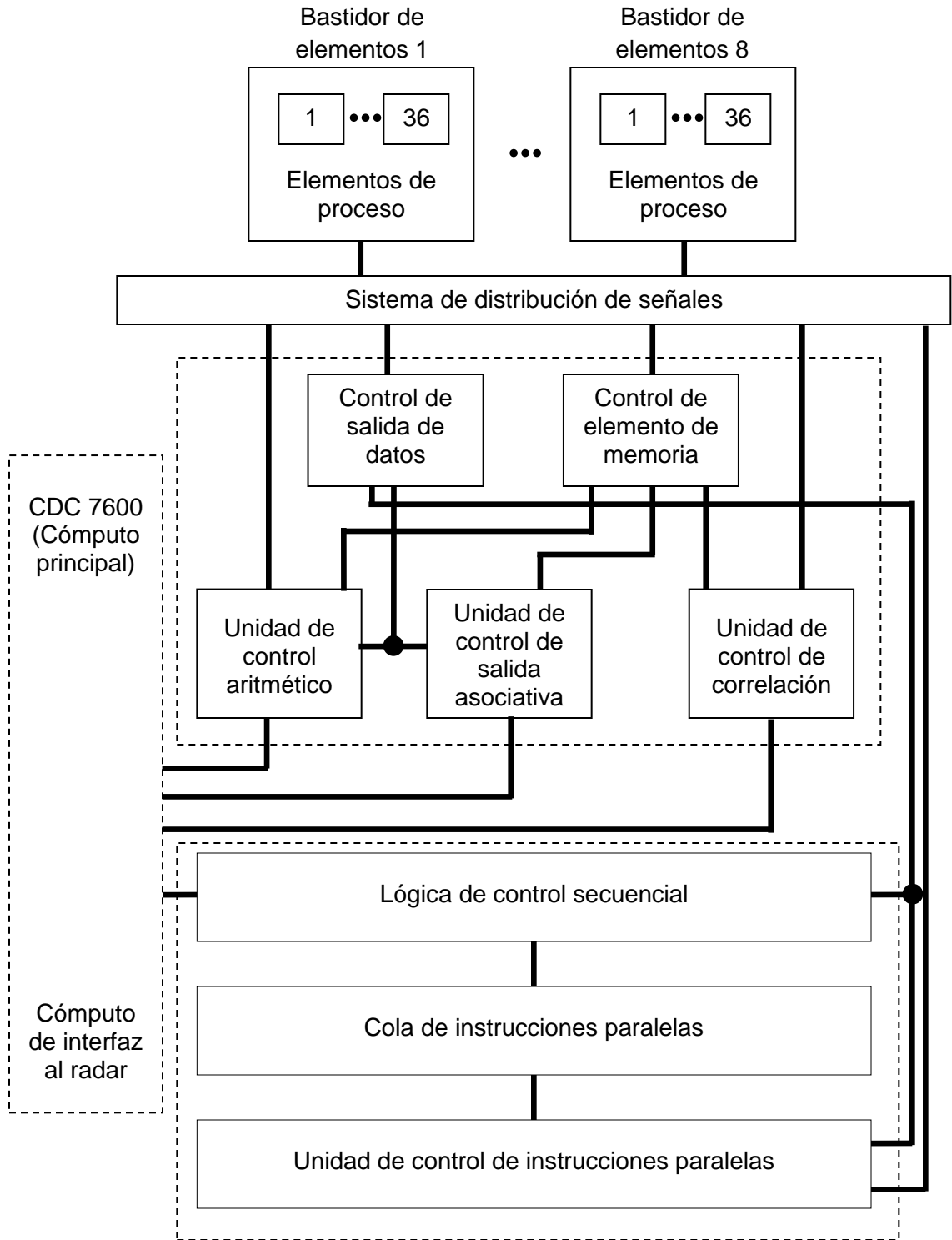


Figura 9. Sistema de control de PEPE.<sup>43</sup>

<sup>43</sup> Fuente: [http://www.snip.gob.ni/xdc/amc3/Seminario/PEPE\\_5T1.pdf](http://www.snip.gob.ni/xdc/amc3/Seminario/PEPE_5T1.pdf)



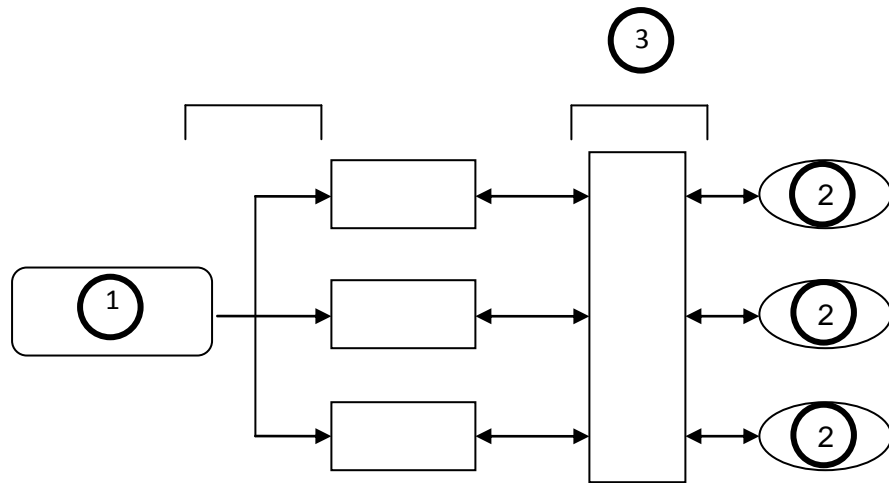
En la actualidad estas arquitecturas han dado pie a nuevos paradigmas computacionales con soporte de hardware para bases de datos y aplicaciones que demandan procesamiento paralelo de conjuntos de datos.

### ACTIVIDAD DE APRENDIZAJE

Analiza las características de los procesadores STARAN o PEPE e identifica cuáles permanecen en las nuevas arquitecturas.

## AUTOEVALUACIÓN

29. La siguiente figura muestra la estructura de la ejecución SIMD. Reconoce los elementos que se encuentran enumerados dentro del círculo y marca con una **X** sólo uno de los incisos donde se encuentre la respuesta correcta.

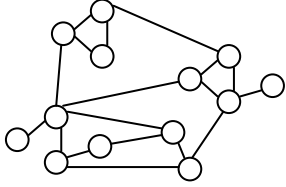
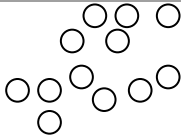
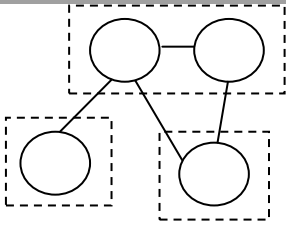
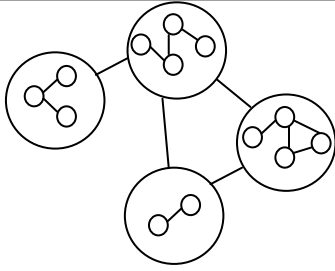


- e) Flujo de instrucciones, flujo de datos, memoria.
- f) Memoria, flujo de datos, elementos de proceso.
- g) Elementos de proceso, memoria, unidad de control.
- h) Unidad de control, flujo de datos, memoria.

30. Explica con tus propias palabras la estructura interna de los elementos de proceso (PE).

31. Observa la siguiente tabla y relaciona la figura con la descripción de cada etapa del diseño de los algoritmos, marca con una **X** sólo uno de los incisos con la respuesta que establezca la relación correcta.

Etapa	Descripción
-------	-------------

<p>7.</p> 	<p>g) Partición.</p>
<p>8.</p> 	<p>h) Agrupación.</p>
<p>9.</p> 	<p>i) Asignación.</p>
<p>10.</p> 	<p>j) Comunicación</p>

e) 1a; 2c; 3d; 4b.

f) 1d; 2a; 3c; 4b.

g) 1c; 2d; 3a; 4b.

h) 1d; 2c; 3b; 4a.

32. Lee con atención la siguiente oración y escribe la respuesta correcta.

Las siguientes características, ¿a qué tipo de algoritmos corresponden? Reproducen el comportamiento de la estructura de datos, son complejos y tienen una ganancia significativa.

33. ¿Cuáles fueron los primeros procesadores matriciales asociativos, diseñados en 1972?

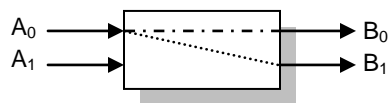
34. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

Dentro de la clasificación de las redes de interconexión SIMS se encuentra las redes estáticas, éstas emplean enlaces directos fijos entre \_\_\_\_\_.

Son dispositivos tales como elementos de proceso, módulos de memoria, procesadores de entrada/salida, etcétera.

- e) Nodo.
- f) Redes.
- g) Switch.
- h) Topología.

35. Las redes de conexión dinámicas se clasifican en dos tipos: la monoetapa y la multietapa, esta última usa cajas de conmutación y éstas a su vez utilizan diferente tipo de configuración. Observa la siguiente figura y determina la configuración que usa esta caja.



36. Los algoritmos paralelos están diseñados con cuatro etapas, un sistema está compuesto por 10 procesadores y en cada uno de ellos se encuentran procesos ejecutándose simultáneamente, necesitamos que tareas y procesos se reacomoden de forma diferente con el fin realizar operaciones vectoriales de alto grado de complejidad. ¿A qué etapa del algoritmo le corresponde realizar esta actividad?

## RESPUESTAS

1. *d.*
2. Cada elemento es como una unidad lógica-aritmética, los datos se registran en la memoria local y la unidad de control cuenta con una memoria propia para

el almacenamiento de programas. Todos los elementos de procesamiento realizan la misma función sincrónicamente en forma esclava bajo el control de la unidad de control.

3. *b.*

4. Los algoritmos matriciales.

5. STARAN y PEPE.

6. *a.*

7. Difusión inferior.

8. Etapa de partición.

## UNIDAD 5

### MULTIPROCESAMIENTO

#### OBJETIVO

El alumno comprenderá la arquitectura y la programación de los multiprocesadores, y mencionará y explicará ejemplos de un sistema multiprocesador.

#### TEMARIO

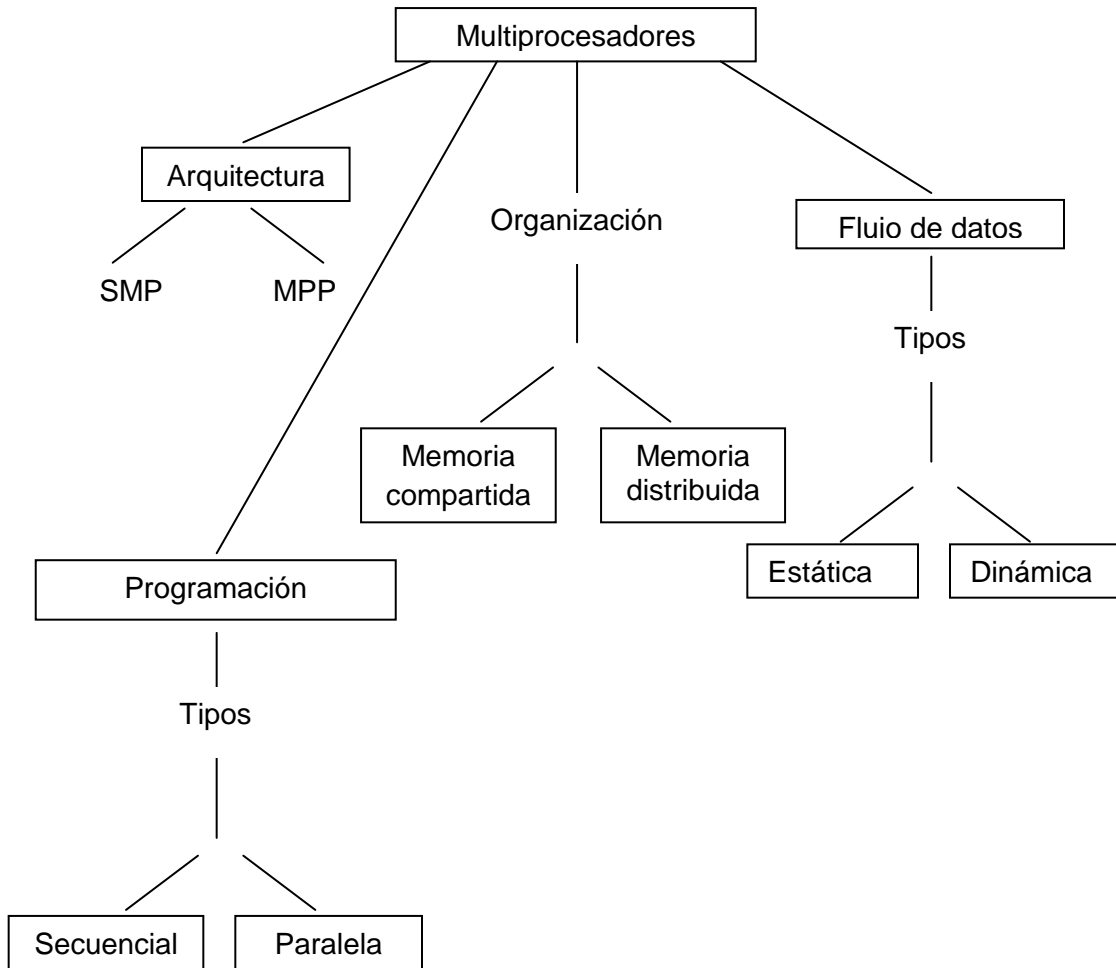
5.1 ARQUITECTURA Y PROGRAMACIÓN DE LOS MULTIPROCESADORES

5.2 MULTIPROCESAMIENTO: CONTROL Y ALGORITMOS

5.3 EJEMPLOS DE SISTEMAS MULTIPROCESADOR

5.4 COMPUTADORES DE FLUJOS DE DATOS

# MAPA CONCEPTUAL



## INTRODUCCIÓN

La programación y el diseño arquitectónico con el que se construyen los sistemas multiprocesadores van de la mano para poder lograr un buen funcionamiento de multiprocesamiento. Ya que tanto el hardware como el software son puntos importantes para lograrlo, la programación para este tipo de sistemas debe ir perfectamente bien encaminada al desarrollo que se pretende lograr, sin embargo si el hardware no es el adecuado, no podremos obtener el resultado deseado.

A lo largo de esta materia hemos aprendido cómo están contruidos los procesadores diseñados con una sola CPU, llamados monoprocesadores, estudiamos el comportamiento de los computadores de segmentación encauzada, los procesadores vectoriales con sus propios métodos de vectorización y optimización y, finalmente, en esta unidad revisaremos un panorama general del diseño, el control y la programación de los multiprocesadores, así como las ventajas y desventajas del uso de estos sistemas.



## 5.1 ARQUITECTURA Y PROGRAMACIÓN DE LOS MULTIPROCESADORES

Hoy en día el uso de los sistemas multiprocesadores es habitual en la mayoría de los sistemas grandes y medianos, incluso existen casos donde se aplica esta arquitectura en equipos personales. Pero, ¿cómo están contruidos los sistemas multiprocesadores? Son arquitecturas contruidas con dos o más procesadores, esto nos permite ejecutar de manera simultánea varios procesos, su ventaja reside en la operación llamada cambio de contexto que consiste en quitar un proceso de la CPU, ejecutar otro y volver a colocar el primero sin que se entere de nada. Los dos objetivos primordiales de los multiprocesadores son acelerar la ejecución de aplicaciones orientadas a la supercomputación y ejecutar más aplicaciones por unidad de tiempo.

Los sistemas multiprocesadores tienen dos modelos de organización:

1. Los de memoria compartida, que permiten, a través de la red de interconexión, que cualquier procesador acceda a cualquier posición de memoria, sin embargo en este tipo de multiprocesadores, la red de interconexión es un cuello de botella lo cual impide que sea escalable. En la figura 1 podemos observar la arquitectura de este modelo.

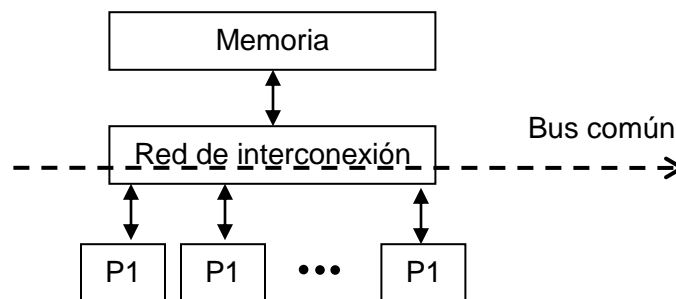


Figura 1. Multiprocesadores con memoria compartida.<sup>44</sup>

2. Los de memoria distribuida, donde la red de interconexión permite que cualquier procesador se comuniquen con cualquier otro del sistema. Los

<sup>44</sup> Fuente:

<http://epsc.upc.edu/projectes/usuaris/miguel.valero/materiales/docencia/doctorado/introduccion.pdf>

sistemas multiprocesadores, pero en especial los de memoria distribuida, tienen una gran potencia de cálculo y al paso del tiempo esta característica sigue aumentando. La figura 2 muestra la estructura de este tipo de sistemas.

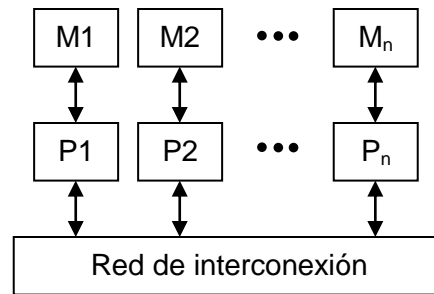


Figura 2. Multiprocesadores con memoria distribuida.<sup>45</sup>

La programación de los sistemas multiprocesadores se clasifica en dos tipos, la secuencial y la paralela. FORTRAN 77 y FORTRAN 77 + librerías paralelas son una subclasificación de la programación secuencial.

FORTRAN es un lenguaje de alto nivel con numerosas mejoras en cuanto a objetos como los de C++ e instrucciones para procesamiento paralelo. (C++ es un lenguaje de programación híbrido, creado con base en el lenguaje C, mejorado en la manipulación de objetos).

En el modelo FORTRAN 77 + librerías paralelas, se limita el uso del sistema multiprocesador por la estructura del modelo. La ventaja de que los desarrolladores usen este lenguaje es que están familiarizados con los lenguajes secuenciales, las bibliotecas están disponibles en todos los sistemas paralelos y éstas dan al desarrollador un control de bajo nivel.

El modelo de la programación paralela se subdivide en variables compartidas y paso de mensajes, en el modelo de variables compartidas las operaciones se descomponen en tareas, los datos son compartidos por todas las tareas y se requieren primitivas de sincronización, éstas se dividen en dos

<sup>45</sup> Fuente:

<http://epsc.upc.edu/projectes/usuaris/miguel.valero/materiales/docencia/doctorado/introduccion.pdf>

clasificaciones: a) las de bajo nivel que son dotadas por el sistema de operación, de esta clasificación dependen los semáforos, b) las primitivas de alto nivel que son dotadas por el lenguaje de programación como ADA, PASCAL o JAVA, de esta clasificación dependen los monitores y la ruta de expresión.

El semáforo es una variable entera, la cual se puede manipular con las siguientes llamadas del sistema (funciones atómicas):

```
wait(S) :  
    while (S <= 0) do;  
        S = S - 1;  
sinal(S): S = S + 1;  
Inicializa (S)
```

En las primitivas de sincronización para señalización, la manera más sencilla de esperar una señal de un subproceso es llamar al método Join que se bloquea hasta que se complete el otro subproceso, en la figura 3 se muestra gráficamente la expresión del paralelismo en el modelo de variables compartidas.

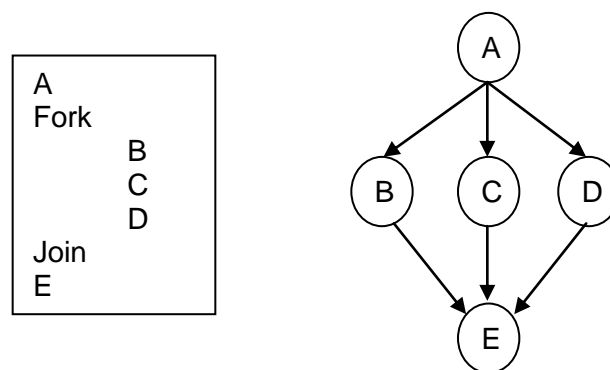


Figura 3. Llamadas de los métodos Fork y Join.<sup>46</sup>

El monitor es una construcción de concurrencia que contiene los datos y procedimientos necesarios para realizar la asignación de un recurso compartido determinado, o de un grupo de recursos compartidos.

<sup>46</sup> Fuente: <http://epsc.upc.edu/projectes/usuarios/miguel.valero/materiales/docencia/doctorado/modprog.pdf>

En el siguiente código se da un ejemplo del uso de monitores, usado en las primitivas de alto nivel:<sup>47</sup>

```
Producer-Consumer: monitor
begin
  buffer está compartido
  is_full es un boolean
  empty and full son semáforos
  procedure produce (data)
  begin
    if (is_full) then
      wait (empty)
    endif
    buffer := data
    is_full := true
    signal (full)
  end
  procedure consume (data)
  begin
    if (not is_full) then
      wait (full)
    endif
    data := buffer
    is_full := false
    signal (empty)
  end
  initialize
    is_full := false
end
```

---

<sup>47</sup> Fuente: <http://epsc.upc.edu/projectes/usuarios/miguel.valero/materiales/docencia/doctorado/modprog.pdf>

El modelo de programación basado en paso de mensajes consiste en dividir operaciones y datos en varios procesos que sean ejecutados en diferentes procesadores que no compartan memoria; la comunicación entre procesos se produce mediante mensajes de envío y recepción de datos, entendiéndose como mensaje a una transferencia de datos de un proceso a otro donde dicha transferencia puede ser de manera síncrona o asíncrona. En la figura 4 se muestra la estructura del modelo.

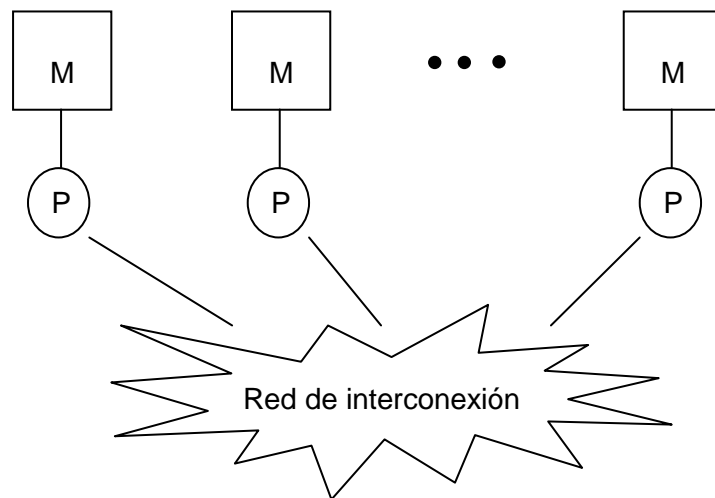


Figura 4. Estructura de la programación basada en paso de mensajes.<sup>48</sup>

La ventaja de este modelo es que es portable a cualquier tipo de arquitectura; la desventaja, es que son sistemas complicados para depurar la memoria compartida y difíciles de programar.

El proceso que se comunica por medio de mensajes está constituido por tres partes:

1. Acceso: donde se obtiene un número de procesos en el grupo que se da al enrolarse en el sistema de paso de mensajes.
2. Ejecución, es cuando el proceso realiza operaciones interactuando con el resto de proceso por medio de rutinas de paso de mensajes.

---

<sup>48</sup> Fuente:  
<http://yboon.net/~azamudio/PROCESAMIENTO%20PARALELO/Paso%20Mensajes/Paso%20Mensajes.pdf>

- Finalización, donde el proceso de desenrola del sistema de paso de mensajes. En la figura 5 se muestra gráficamente el proceso del modelo de paso de mensajes.

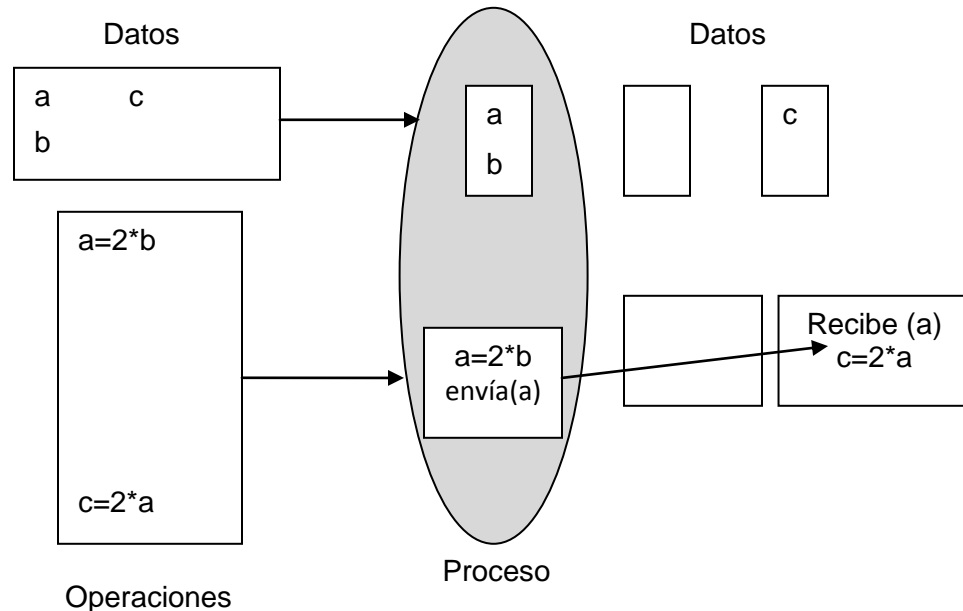


Figura 5. Proceso de comunicación mediante el paso de mensajes.<sup>49</sup>

Para hacer funcionar correctamente el multiprocesamiento es necesario tener buen conocimiento tanto de hardware como de software, y reconocer perfectamente la interconexión entre los procesadores y la programación que ha sido diseñada para escribir aplicaciones y software que se aproveche al máximo. Debido a esto, la arquitectura y la programación van de la mano para garantizar un buen funcionamiento de multiprocesamiento.

## ACTIVIDAD DE APRENDIZAJE

Corre el siguiente programa basado en el modelo de variables compartidas y comenta con tus compañeros.

<sup>49</sup> Fuente: <http://epsc.upc.edu/projectes/usuarios/miguel.valero/materiales/docencia/doctorado/modprog.pdf>

```

real data (N) SHARED
integer i,N,B

doall i=1,N,B
  do j=i, i+B-1
    if data [i] < data [j]
      then data [i] = data [j]
    endif
  enddo
enddo
do i=1,N,B
  if data [1] < data [i]
    then data [1] = data [i]
  endif
enddo
write (data [1])

```

## 5.2 MULTIPROCESAMIENTO: CONTROL Y ALGORITMOS

Los sistemas operativos tienen el control del multiprocesamiento, en un sistema multiprocesador su construcción se realiza por medio de más de dos procesadores y una interconexión que puede ser un bus común, sin embargo el sistema operativo es el encargado de realizar la sincronización entre los procesadores y la planificación de los procesos; lo más importante es que todo esto es un proceso transparente para el usuario.

Veremos algunos algoritmos que se usan en la programación de los multiprocesadores como los algoritmos Peterson y la instrucción hardware TSL: Test & Set Lock.

¿En qué consiste el algoritmo de Peterson? Es una solución para dos procesos, los cuales comparten las variables *int turn;* y *boolean flag[2];* donde la variable *turn* indica el turno de quién puede entrar en la sección crítica y el arreglo *flag* es usado para indicar que un proceso está listo para entrar a la

sección crítica. Este algoritmo asume que las primitivas de lectura y escritura son atómicas, es decir, que las primitivas no pueden ser interrumpidas.

En seguida se muestra el algoritmo de Peterson para el proceso  $P_i$ :

```
while (true) {
    flag[i] = true;
    turn = j;
    while ( flag[j] && turn == j);
    CRITICAL SECTION
    flag[i] = false;
    REMAINDER SECTION
```

La siguiente instrucción TestAndSet es parte de la mayoría de las arquitecturas:

```
boolean TestAndSet (boolean *target)
{
    boolean rv = *target;
    *target = TRUE;
    return rv;
}
```

La siguiente instrucción usa la solución TestAndSet con la variable compartida lock (booleana):

```
while (true) {
    while (TestAndSet(&lock))
        ; // do nothing
    CRITICAL SECTION
    lock = false;
    REMAINDER SECTION
```



¿Cuál es la función de los semáforos? La implementación de los semáforos debe garantizar que dos o más procesos puedan ejecutar *wait()* y *signal()* sobre el mismo semáforo al mismo tiempo, en la siguiente instrucción se muestra el uso correcto e incorrecto de semáforos:

Uso correcto:

```
signal (mutex) .... wait (mutex);
```

Uso incorrecto:

```
wait (mutex) ... wait (mutex);
```

```
signal (mutex) ... signal (mutex);
```

```
(mutex) ... wait (mutex);
```

```
signal (mutex) .... (mutex);
```

¿Cuál es la función de los monitores? La implementación de los monitores nos permite identificar una abstracción de alto nivel para proveer mecanismos de sincronización efectivos y simples, las siguientes líneas nos describen el algoritmo del uso del monitor:

```
monitor nombre-del-monitor {  
    // declaración de variables compartidas  
    condition c;  
    procedure P1 (...) { .... }  
    ...  
    procedure Pn (...) {.....}  
    Initialization code ( ....) { ... }  
    ...  
}
```

## ACTIVIDAD DE APRENDIZAJE

Desarrolla un algoritmo basado en la solución de Peterson para un proceso.

### 5.3 EJEMPLOS DE SISTEMAS MULTIPROCESADOR

La arquitectura SMP (Symmetric Multi-Processing) es creada con dos o más procesadores que comparten una única memoria central y dispositivos de E/S, esta arquitectura también es llamada UMA (Uniform Memory Access), por la característica de la memoria compartida a todos los procesadores. Estos sistemas permiten que cualquier procesador trabaje en cualquier tarea sin importar la localización en memoria, el sistema de interconexión que usan para comunicarse es un bus compartido.

Las principales características de configuraciones de los sistemas de trato simétrico son:

- El maestro flota de un procesador a otro.
- Mejor balance de cargas.
- El código debe ser reentrante al ser ejecutado en varios procesadores.
- Mayor disponibilidad y mejor uso eficiente de los recursos.

Los sistemas IBM 9020 y Carnegie Multi-Mini-Procesor son algunos ejemplos de sistemas de trato simétrico. La máquina IBM 9020 es un sistema multiprocesador de la familia de ordenadores IBM System/360 adaptados para el uso en Estados Unidos.



Figura 6. Sistema IBM 9020.<sup>50</sup>

---

<sup>50</sup> Fuente: <http://infolab.stanford.edu/pub/voy/museum/pictures/display/9020FAA.jpg>

Los sistemas con arquitectura MPP (Massively Parallel Processing) son un esquema de una máquina paralela que está constituida por varias unidades de procesamiento, pero debido a que cada una de las máquinas cuenta con su propio procesador y su propia memoria, se puede decir que son máquinas básicamente independientes (figura 7), ésta es una de las principales diferencias con la arquitectura SMP. La comunicación con las demás unidades de procesamiento es por medio de un canal exclusivo para ese propósito.

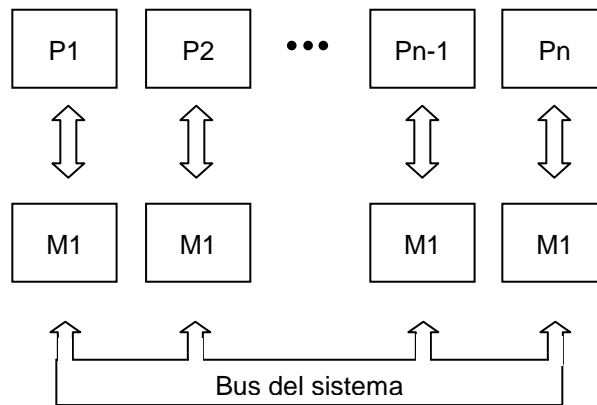


Figura 7. Arquitectura MPP.<sup>51</sup>

La desventaja de este tipo de sistemas es que es que la programación se vuelve muy difícil, ya que la memoria se rompe en pequeños espacios separados, así como la sincronización de los datos entre las tareas ampliamente distribuidas.

¿Cuáles son las tareas funcionales de los sistemas multiprocesadores? Algunas de son: administrar recursos, controlar y asignar memoria, planificar y asignar procesos, intercomunicar procesadores, balancear cargas de procesadores, proteger tablas y datos del sistema, balancear carga de trabajo de entrada/salida y realizar la reconfiguración.

Estos sistemas nos ofrecen grandes ventajas, sin embargo es necesario revisar algunas de las desventajas que podemos encontrar en este tipo de arquitecturas, como: son sistemas muy complejos, difíciles de diseñar, difíciles

<sup>51</sup> Fuente: <http://publiespe.espe.edu.ec/articulos/sistemas/arquitectura/arquitectura.htm>

de controlar, el desempeño puede aumentar al agregar un componente o una unidad funcional, pero no es lineal con el incremento de costos.<sup>52</sup>

## ACTIVIDAD DE APRENDIZAJE

Revisa las ventajas y desventajas de los sistemas multiprocesadores y genera una red semántica que sintetice la información.

### 5.4 COMPUTADORES DE FLUJOS DE DATOS

Existen dos formas de procesar la información, una de ellas es mediante la ejecución en serie de una lista de instrucciones y la segunda es por medio de la ejecución de las instrucciones según las pidan los datos disponibles; esta última forma desde el punto de vista de la paralelización resulta mucho más interesante, ya que las instrucciones se ejecutan en el momento que se tienen los datos necesarios y la ejecución de las instrucciones es demandada al mismo tiempo. Para este tipo de proceso es donde la arquitectura de flujo de datos nos es de gran ayuda.

Con esta arquitectura los datos están disponibles por medio de la canalización de los resultados de las instrucciones ejecutadas con anterioridad, esta canalización forma un flujo de datos que va disparando las instrucciones a ejecutar. Las instrucciones de flujo de datos no utilizan variables en una memoria compartida global, llevan los valores de las variables en ellas mismas. En estas máquinas las instrucciones pueden ser ejecutadas simultáneamente, lo que nos da la posibilidad de un alto grado de concurrencia y paralelización.

Las máquinas de flujo de datos están clasificadas de dos formas:

1. La estática, donde una instrucción se activa hasta que se reciban todos los operandos requeridos y haya alguna instrucción esperando recibir el resultado de esta instrucción (figura 8).
2. La dinámica, donde una instrucción se activa cuando se reciben todos los operandos requeridos, es decir que varios conjuntos de operandos pueden estar listos para una instrucción al mismo tiempo

---

<sup>52</sup> Fuente: <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>

(figura 9). Este tipo de máquinas permiten mayor paralelismo, sin embargo es aquí donde se hace necesario establecer un mecanismo que permita distinguir los valores de diferentes conjuntos de operandos para una instrucción.

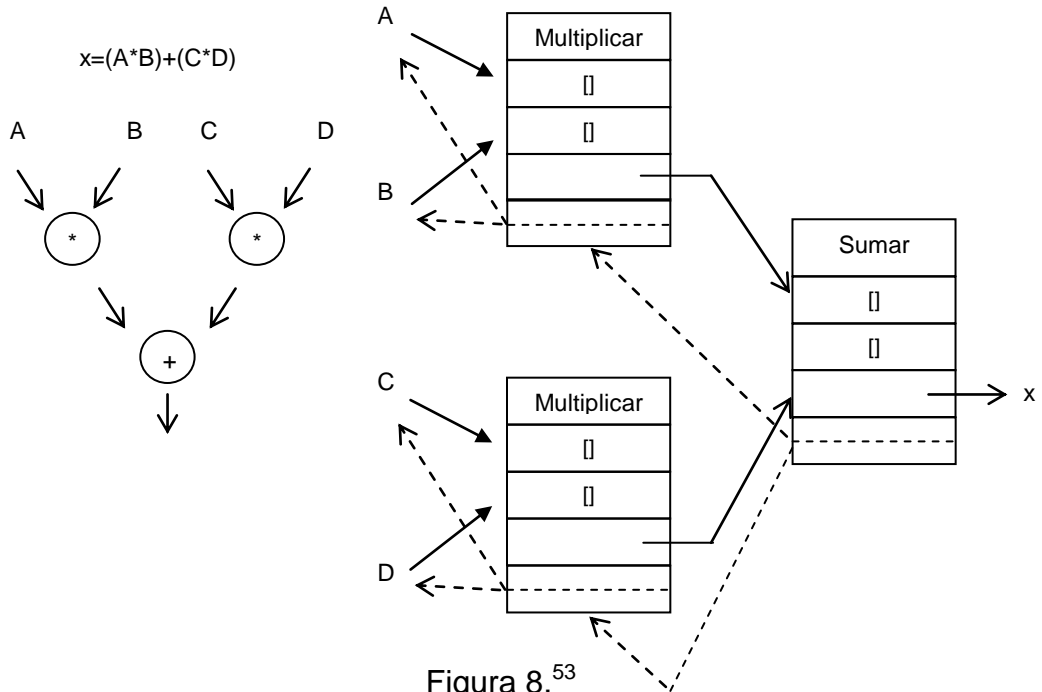


Figura 8.<sup>53</sup>

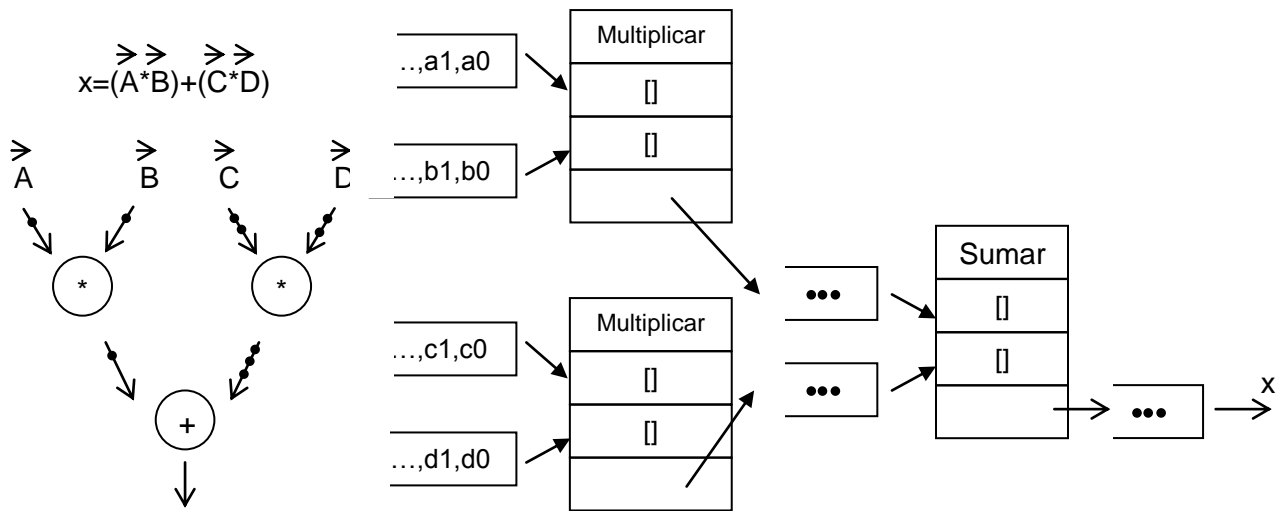


Figura 9.<sup>54</sup>

<sup>53</sup> Fuente: [http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac\\_otras.pdf](http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac_otras.pdf)

<sup>54</sup> Fuente: [http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac\\_otras.pdf](http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac_otras.pdf)

En las máquinas de flujo de datos se utiliza el grafo de flujo de datos, el cual muestra las dependencias de datos entre las instrucciones y representa los pasos de ejecución de un programa que sirve como interfaz entre la arquitectura y el lenguaje de programación. Los nodos o actores en el grafo representan los operandos y se encuentran interconectados mediante arcos de entrada y salida que llevan etiquetas que contienen algún valor, estas etiquetas están clasificadas en dos tipos: etiquetas de datos y etiquetas booleanas. Para distinguir el tipo de entrada a un actor se utilizan flechas simples, si lleva etiquetas de datos, y flechas dobles cuando las etiquetas son booleanas. En la figura 10 se muestra el grafo de flujo de datos para el cálculo de  $N!$ .

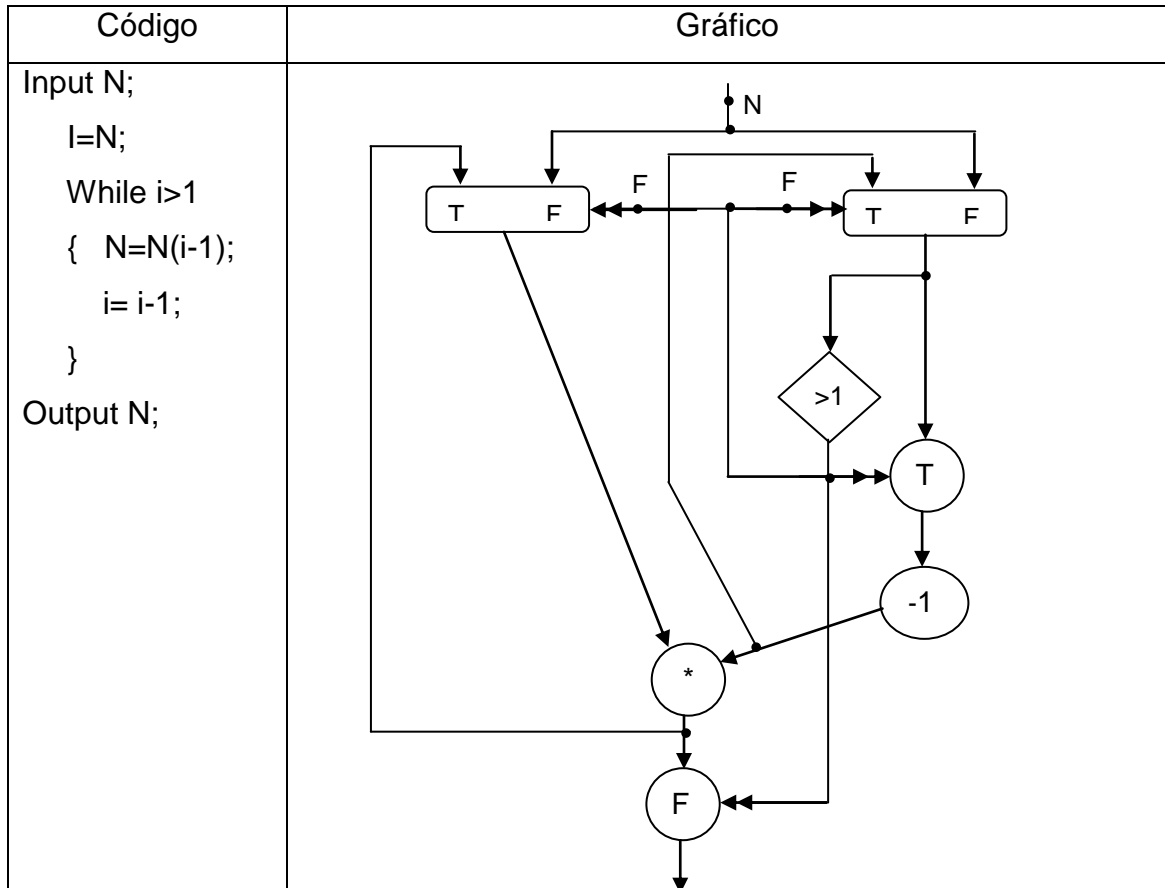


Figura10.<sup>55</sup>

<sup>55</sup> Fuente: [http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac\\_otras.pdf](http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac_otras.pdf)

La estructura básica de un computador de flujo de datos, como el caso de la que fue desarrollada en el MIT, está compuesta por cinco unidades (figura 11). Cada célula de instrucción guarda una instrucción que consiste en un código de operación, operandos y dirección de destino. La instrucción se activa cuando se reciben todos los operandos y señales de control requeridos. La red de arbitraje manda la instrucción activa como un paquete de operación a algunos de los elementos de proceso y una vez que la instrucción es ejecutada, el resultado se devuelve a través de la red de distribución al destino en memoria. Cada resultado se envía como un paquete que consiste en un resultado más una dirección de destino.

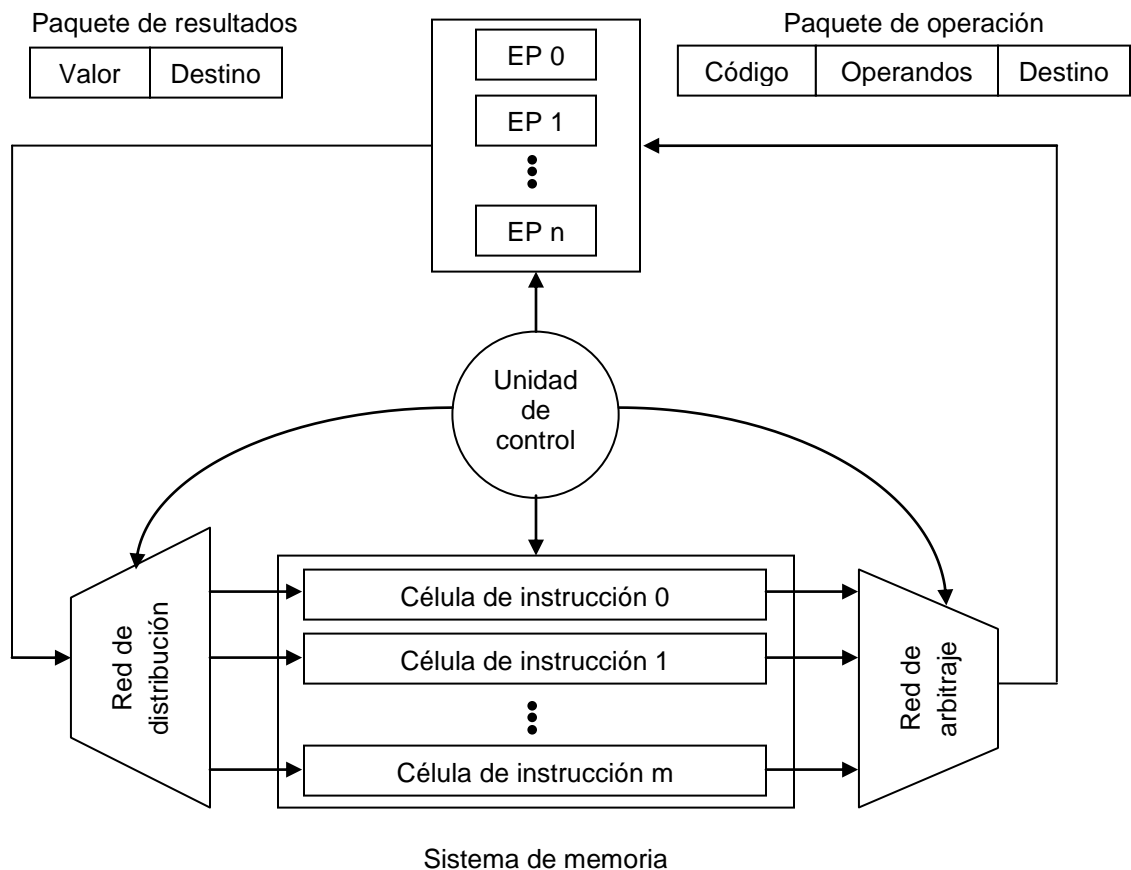


Figura11.<sup>56</sup>

<sup>56</sup> Fuente: [http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac\\_otras.pdf](http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac_otras.pdf)

## ACTIVIDAD DE APRENDIZAJE

SisteMex y asociados, S.A., es una empresa constituida desde 1996 dedicada a la consultoría e implementación de sistemas informáticos, los consultores son personas especialistas y capaces para proponer a cada uno de sus clientes la implementación de sistemas que más convenga a sus necesidades.

Danimex S.A. requiere hacer una implementación de sistemas con arquitectura de flujo de datos, los consultores de Systemex realizaron un análisis y decidieron implementar las máquinas con arquitectura de flujo de datos estáticas, esta clasificación se realiza cuando las instrucciones se activan hasta que se reciben la totalidad de los datos.

Analiza el caso, fundamenta porqué fue la mejor decisión implementar el flujo de datos estática e indica qué implicaría hacer la implementación con flujo de datos dinámica.



## AUTOEVALUACIÓN

37. Explica con tus propias palabras en qué consiste el multiprocesamiento.

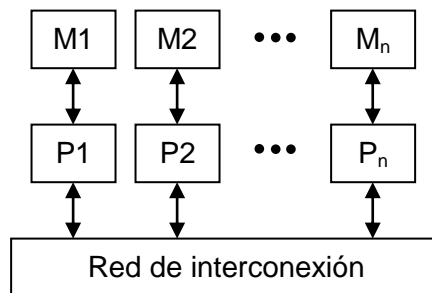
38. Lee con atención el siguiente enunciado y contesta correctamente.

¿Qué tipo de sistemas permiten que un procesador acceda a cualquier posición de memoria a través de la red de interconexión, la cual se convierte en un cuello de botella que impide que sea escalable?

39. Lee con atención la siguiente pregunta y contesta correctamente.

¿Qué ventaja importante tienen los sistemas multiprocesadores de memoria distribuida?

40. Observa el siguiente esquema e identifica a qué tipo de multiprocesador corresponde.

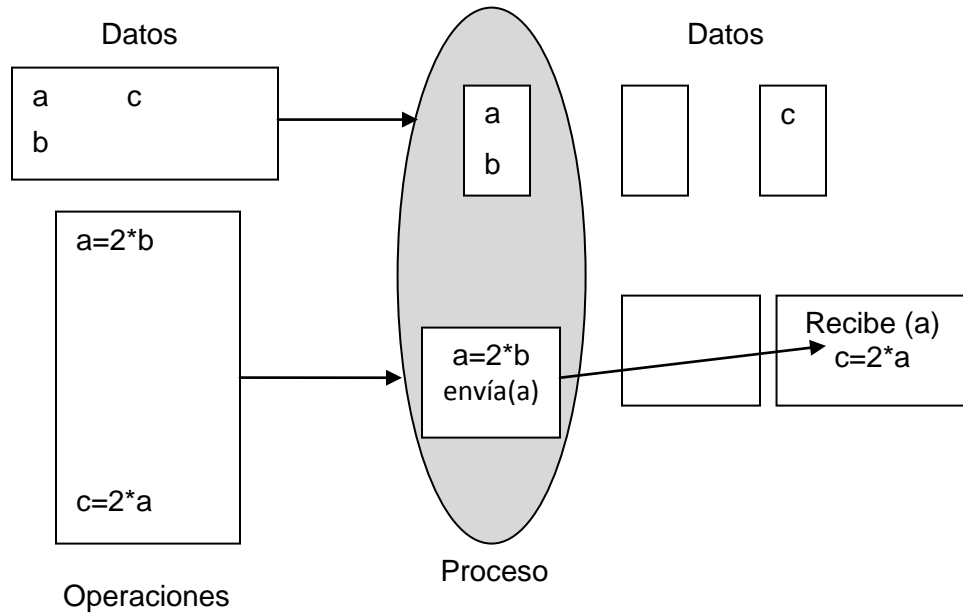


41. Lee con atención la siguiente oración y marca con una **X** sólo uno de los incisos con la respuesta correcta.

De las siguientes sentencias ¿cuál corresponde al uso correcto de los semáforos?

- i) *signal* (mutex) .... *wait* (mutex);
- j) (mutex) ... *wait* (mutex);
- k) *signal* (mutex) .... (mutex);
- l) *wait* (mutex) ... *wait* (mutex);

42. Crea la estructura de un algoritmo implementando el uso del monitor.
43. Observa la siguiente figura y explica con tus propias palabras el proceso a seguir a través del paso de mensajes.



## RESPUESTAS

1. Es un mecanismo que nos permite ejecutar varios procesos de forma simultánea.
2. Los sistemas multiprocesadores de memoria compartida.
3. Tienen una gran potencia de cálculo.
4. Memoria distribuida.
5. a).
6. monitor nombre {  
    condition c;  
    procedure P1 (...) { ..... }  
    ...  
    procedure Pn (...) {.....}  
    Initialization code ( ....) { ... }  
    ...  
}
7. Primero el proceso pasará por la etapa donde se dará acceso para enrolarse en el sistema de paso de mensajes, posteriormente se realiza la interacción con el resto de proceso por medio de rutinas de paso de mensajes y finalmente el proceso de desenrolamiento del sistema.

## BIBLIOGRAFÍA

Hwang, Kai y Faye Briggs, *Arquitectura de computadoras y procesamiento paralelo*, McGraw-Hill, México, 1988.

Mano, M. Morris, *Arquitectura de computadoras*, 3a. ed., México, Pearson Education, 1993.

## GLOSARIO

**Ancho de banda:** es la máxima cantidad de datos que pueden pasar por un camino de comunicación en un momento dado, normalmente medido en segundos. Cuanto mayor sea el ancho de banda, más datos podrán circular por segundo.

**Chipset o circuito integrado auxiliar:** es un conjunto de circuitos integrados diseñados con base en la arquitectura del procesador y en algunos casos se diseña como parte integral de ésta.

**Colisión:** es el uso simultáneo de varios segmentos por parte de ejecuciones distintas de la misma función o por varias de las funciones.

**Computadores paralelos:** son dispositivos que cuentan con dos o más procesadores interconectados, los cuales son capaces de intercambiar información y ejecutar instrucciones de forma simultánea.

**Etapa:** son circuitos combinatoriales que efectúan operaciones aritméticas o lógicas sobre el flujo de datos que circula a través del cauce.

**Latencia:** es el número de desplazamientos entre dos tablas que nos dará el número de ciclos que esperamos.

**Monoprocesador:** sistemas con un solo procesador.

**Multiprocesador:** sistemas con más de un procesador.

**Multiprocesamiento:** es el uso de múltiples procesos concurrentes en un sistema, en lugar de un único proceso en un instante determinado.

**Multiprogramación:** es la ejecución de múltiples tareas, las cuales comparten recursos de un mismo equipo de cómputo.

**Nodo:** son todos aquellos dispositivos que se quieren conectar a la red.

**Algoritmo:** es un conjunto de sentencias o instrucciones que expresan la lógica de un programa y nos permite obtener un resultado determinado.

**Paralelismo:** es un mecanismo que consiste en ejecutar más instrucciones en menos tiempo.

**Segmentación:** es una técnica de implementación de procesadores que desarrolla el paralelismo a nivel instrucción, es decir, divide una unidad funcional en varias etapas más rápidas, con el fin de mejorar su rendimiento.

**Segmentación encauzada:** es una técnica utilizada en el diseño y la implantación de microprocesadores, en la cual múltiples instrucciones pueden ejecutarse simultáneamente.

**Segmentación encauzada lineal:** es una técnica donde las etapas de la unidad funcional se ejecutan en orden secuencial.

**Tabla de reserva:** contiene información sobre la ocupación de cada segmento en cada uno de los ciclos máquina hasta que termina de ejecutarse la tarea.

**Vectorización:** es la conversión de un programa, el cual corresponde de un procesador escalar a uno vectorial.