

VNIVERSITAT  VALÈNCIA



UNIVERSITAT DE VALÈNCIA

---

AMPLIACIÓN DE ARQUITECTURA DE  
COMPUTADORES

INGENIERÍA INFORMÁTICA

Profesor: Fernando Pardo Carpio

Valencia, 8 de octubre de 2001



# Prefacio

El temario recogido en estos apuntes se corresponde con la asignatura de *Ampliación de Arquitectura de Computadores* de la Ingeniería Informática de la Universidad de Valencia. Esta asignatura consta de 3 créditos teóricos y 1.5 prácticos, y se cursa en quinto. Esta asignatura es la continuación de la asignatura de *Arquitectura e Ingeniería de Computadores* de cuarto curso.

El contenido principal de la asignatura está formado por sistemas de altas prestaciones como los multicomputadores, sin olvidar otras arquitecturas de alto rendimiento como los procesadores vectoriales y matriciales. Por último se tratan algunas arquitecturas de aplicación específica.

Estos apuntes son un subconjunto de los apuntes de la asignatura de plan viejo *Arquitecturas Avanzadas*. Los contenidos de esta asignatura de plan viejo se han dividido entre dos asignaturas, por un lado la de *Ampliación de Arquitectura de Computadores*, y por el otro en parte (segundo cuatrimestre) de la asignatura de *Arquitectura e Ingeniería de los Computadores*. Por esta razón existen tres versiones de estos apuntes:

- *Arquitecturas Avanzadas*. Esta asignatura ya no se da pero son los apuntes más completos, ya que recogen todos los temas preparados hasta el momento.
- *Ampliación de Arquitectura de Computadores*. Se recogen los temas de vectoriales, matriciales, multicomputadores y otras arquitecturas avanzadas. Se incluye también una introducción al paralelismo como apéndice.
- *Arquitectura e Ingeniería de los Computadores* (segundo cuatrimestre). En esta asignatura se recogen el resto de temas de Arquitecturas Avanzadas como son el rendimiento de los sistemas paralelos, redes de interconexión y multiprocesadores.

Estos apuntes se empezaron a preparar en 1998. Posteriormente, a principios del 2000, el profesor Antonio Flores Gil del Departamento de Ingeniería y Tecnología de Computadores de la Facultad de Informática de la Universidad de Murcia, aportó las fuentes de sus mucho más completos apuntes de Arquitectura de Computadores con el fin de que se pudiesen adaptar a las necesidades del temario. Con los apuntes que ya se tenían y con los del profesor Antonio Flores, se ha realizado esta nueva versión que todavía está en construcción.

Se incluyen las referencias bibliográficas utilizadas en la elaboración de los apuntes, incluyéndose un pequeño apéndice donde se comentan los diferentes libros utilizados. Con esto se ha pretendido dar una visión global a las arquitecturas avanzadas, ofreciéndole al alumno un abanico amplio de posibilidades para ampliar información.

Fernando Pardo en Valencia a 8 de octubre de 2001



# Índice General

<b>1</b>	<b>Procesadores vectoriales</b>	<b>1</b>
1.1	Procesador vectorial básico . . . . .	2
1.1.1	Arquitectura vectorial básica . . . . .	2
1.1.2	Instrucciones vectoriales básicas . . . . .	4
1.1.3	Ensamblador vectorial DLXV . . . . .	6
1.1.4	Tiempo de ejecución vectorial . . . . .	8
1.1.5	Unidades de carga/almacenamiento vectorial . . . . .	10
1.2	Memoria entrelazada o intercalada . . . . .	11
1.2.1	Acceso concurrente a memoria (acceso C) . . . . .	13
1.2.2	Acceso simultáneo a memoria (acceso S) . . . . .	14
1.2.3	Memoria de acceso C/S . . . . .	15
1.2.4	Rendimiento de la memoria entrelazada y tolerancia a fallos . . . . .	15
1.3	Longitud del vector y separación de elementos . . . . .	18
1.3.1	Control de la longitud del vector . . . . .	18
1.3.2	Cálculo del tiempo de ejecución vectorial . . . . .	19
1.3.3	Separación de elementos en el vector . . . . .	21
1.4	Mejora del rendimiento de los procesadores vectoriales . . . . .	23
1.4.1	Encadenamiento de operaciones vectoriales . . . . .	23
1.4.2	Sentencias condicionales . . . . .	23
1.4.3	Matrices dispersas . . . . .	25
1.5	El rendimiento de los procesadores vectoriales . . . . .	27
1.5.1	Rendimiento relativo entre vectorial y escalar . . . . .	27
1.5.2	Medidas del rendimiento vectorial . . . . .	28
1.6	Historia y evolución de los procesadores vectoriales . . . . .	32
<b>2</b>	<b>Procesadores matriciales</b>	<b>35</b>
2.1	Organización básica . . . . .	35
2.2	Estructura interna de un elemento de proceso . . . . .	36
2.3	Instrucciones matriciales . . . . .	38
2.4	Programación . . . . .	38
2.4.1	Multiplicación SIMD de matrices . . . . .	38
2.5	Procesadores asociativos . . . . .	38
2.5.1	Memorias asociativas . . . . .	38
2.5.2	Ejemplos de procesadores asociativos . . . . .	38
<b>3</b>	<b>Multicomputadores</b>	<b>39</b>
3.1	Redes de interconexión para multicomputadores . . . . .	41
3.1.1	Topologías estrictamente ortogonales . . . . .	43
3.1.2	Otras topologías directas . . . . .	45
3.1.3	Conclusiones sobre las redes directas . . . . .	50
3.2	La capa de conmutación o control de flujo ( <i>switching</i> ) . . . . .	52
3.2.1	Elementos básicos de la conmutación . . . . .	52
3.2.2	Conmutación de circuitos . . . . .	56
3.2.3	Conmutación de paquetes . . . . .	58

3.2.4	Conmutación de paso a través virtual, <i>Virtual Cut-Through</i> (VCT)	59
3.2.5	Conmutación de lombriz ( <i>Wormhole</i> ) . . . . .	61
3.2.6	Conmutación cartero loco . . . . .	63
3.2.7	Canales virtuales . . . . .	66
3.2.8	Mecanismos híbridos de conmutación . . . . .	68
3.2.9	Comparación de los mecanismos de conmutación . . . . .	68
3.3	La capa de encaminamiento ( <i>routing</i> ) . . . . .	70
3.3.1	Clasificación de los algoritmos de encaminamiento . . . . .	71
3.3.2	Bloqueos . . . . .	72
3.3.3	Teoría para la evitación de bloqueos mortales ( <i>deadlocks</i> ) . . . . .	75
3.3.4	Algoritmos deterministas . . . . .	81
3.3.5	Algoritmos parcialmente adaptativos . . . . .	84
3.3.6	Algoritmos completamente adaptativos . . . . .	87
3.3.7	Comparación de los algoritmos de encaminamiento . . . . .	88
<b>4</b>	<b>Otras arquitecturas avanzadas</b>	<b>93</b>
4.1	Máquinas de flujo de datos . . . . .	94
4.1.1	Grafo de flujo de datos . . . . .	94
4.1.2	Estructura básica de un computador de flujo de datos . . . . .	96
4.2	Otras arquitecturas . . . . .	97
<b>A</b>	<b>Introducción a las arquitecturas avanzadas</b>	<b>99</b>
A.1	Clasificación de Flynn . . . . .	100
A.2	Otras clasificaciones . . . . .	101
A.3	Introducción al paralelismo . . . . .	108
A.3.1	Fuentes del paralelismo . . . . .	108
A.3.2	El paralelismo de control . . . . .	109
A.3.3	El paralelismo de datos . . . . .	109
A.3.4	El paralelismo de flujo . . . . .	111
<b>B</b>	<b>Comentarios sobre la bibliografía</b>	<b>113</b>
	<b>Bibliografía</b>	<b>115</b>
	<b>Índice de Materias</b>	<b>117</b>

# Índice de Figuras

1.1	La arquitectura de un supercomputador vectorial. . . . .	3
1.2	Estructura básica de una arquitectura vectorial con registros, DLXV. . .	4
1.3	Características de varias arquitecturas vectoriales. . . . .	5
1.4	Instrucciones vectoriales del DLXV. . . . .	7
1.5	Penalización por el tiempo de arranque en el DLXV. . . . .	9
1.6	Tiempos de arranque y del primer y último resultados para los convoys 1-4. . . . .	10
1.7	Dos organizaciones de memoria entrelazada con $m = 2^a$ módulos y $w = 2^a$ palabras por módulo. . . . .	12
1.8	Tiempo de acceso para las primeras 64 palabras de doble precisión en una lectura. . . . .	13
1.9	Acceso segmentado a 8 palabras contiguas en una memoria de acceso C. . . . .	14
1.10	Organización de acceso S para una memoria entrelazada de $m$ vías. . . . .	14
1.11	Organización de acceso C/S. . . . .	15
1.12	Dos organizaciones de memoria entrelazada usando 8 módulos: (a) 2 bancos y 4 módulos entrelazados, (b) 4 bancos y 2 módulos entrelazados. . . . .	17
1.13	Un vector de longitud arbitraria procesado mediante seccionamiento. Todos los bloques menos el primero son de longitud MVL. En esta figura, la variable $m$ se usa en lugar de la expresión $(n \bmod MVL)$ . . . . .	19
1.14	Tiempo de ejecución por elemento en función de la longitud del vector. . . . .	21
1.15	Temporización para la ejecución no encadenada y encadenada. . . . .	24
1.16	Rendimiento relativo escalar/vectorial. . . . .	27
1.17	Formación de convoys en el bucle interior del código DAXPY. . . . .	28
1.18	Comparación del rendimiento de los procesadores vectoriales y los microprocesadores escalares para la resolución de un sistema de ecuaciones lineales denso (tamaño de la matriz= $n \times n$ ). . . . .	33
2.1	Computador matricial básico. . . . .	36
2.2	Componentes de un elemento de proceso (EP) en un computador matricial. . . . .	37
3.1	Arquitectura básica de un multicomputador. . . . .	40
3.2	Clasificación de las redes de interconexión . . . . .	42
3.3	Variaciones de mallas y toros. . . . .	44
3.4	Topologías estrictamente ortogonales en una red directa. . . . .	45
3.5	Hipercubos, ciclo cubos y $n$ -cubos $k$ -arios. . . . .	46
3.6	Otras topologías directas. . . . .	47
3.7	Matriz lineal, anillos, y barril desplazado. . . . .	48
3.8	Algunas topologías árbol. . . . .	49
3.9	Árbol, estrella y árbol grueso. . . . .	50
3.10	Distintas unidades de control de flujo en un mensaje. . . . .	53
3.11	Un ejemplo de control de flujo asíncrono de un canal físico. . . . .	53
3.12	Un ejemplo de control de flujo síncrono de un canal físico. . . . .	54
3.13	Modelo de encaminador ( <i>router</i> ) genérico. (LC = Link controller.) . . . . .	55

3.14	Cálculo de la latencia en una red para el caso de ausencia de carga (R = Encaminador o <i>Router</i> ). . . . .	56
3.15	Cronograma de un mensaje por conmutación de circuitos. . . . .	57
3.16	Un ejemplo del formato en una trama de creación de un circuito. (CHN = Número de canal; DEST = Dirección destino; XXX = No definido.) . . . . .	57
3.17	Cronograma de un mensaje por conmutación de paquetes. . . . .	59
3.18	Un ejemplo de formato de la cabecera del paquete. (DEST = Dirección destino; LEN = Longitud del paquete en unidades de 192 bytes; XXX = No definido.) . . . . .	59
3.19	Cronograma para un mensaje conmutado por VCT. ( $t_{blocking}$ = Tiempo de espera en un enlace de salida.) . . . . .	60
3.20	Cronograma de un mensaje conmutado mediante wormhole. . . . .	61
3.21	Un ejemplo de mensaje bloqueado con la técnica wormhole. . . . .	62
3.22	Formato de los paquetes conmutados mediante wormhole en el Cray T3D. . . . .	62
3.23	Cronograma de la transmisión de un mensaje usando la conmutación del cartero loco. . . . .	63
3.24	Un ejemplo del formato de un mensaje en la técnica de conmutación del cartero loco. . . . .	64
3.25	Ejemplo de encaminamiento con la conmutación del cartero loco y la generación de flits de dirección muertos. . . . .	65
3.26	Canales virtuales. . . . .	67
3.27	Un ejemplo de la reducción del retraso de la cabecera usando dos canales virtuales por canal físico. . . . .	68
3.28	Latencia media del paquete vs. tráfico aceptado normalizado en una malla $16 \times 16$ para diferentes técnicas de conmutación y capacidades de buffer. (VC = Virtual channel; VCT = Virtual cut-through.) . . . . .	69
3.29	Una taxonomía de los algoritmos de encaminamiento . . . . .	71
3.30	Configuración de bloqueo en una malla 2-D. . . . .	73
3.31	Una clasificación de las situaciones que pueden impedir el envío de paquetes. . . . .	74
3.32	Configuración ilegal para $R$ . . . . .	76
3.33	Redes del ejemplo anterior. . . . .	78
3.34	Red del ejemplo anterior. . . . .	79
3.35	El algoritmo de encaminamiento $XY$ para mallas 2-D. . . . .	82
3.36	El algoritmo de encaminamiento por dimensiones para hipercubos. . . . .	83
3.37	Grafo de dependencias entre canales para anillos unidireccionales. . . . .	83
3.38	El algoritmo de encaminamiento por dimensiones para toros 2-D unidireccionales. . . . .	85
3.39	Caminos permitidos en el encaminamiento totalmente adaptativo y adaptativo por planos. . . . .	86
3.40	Redes crecientes y decrecientes en el plano $A_i$ para el encaminamiento adaptativo por planos. . . . .	86
3.41	Latencia media del mensaje vs. tráfico normalizado aceptado para mallas $16 \times 16$ para una distribución uniforme del destino de los mensajes. . . . .	89
3.42	Latencia media del mensaje vs. tráfico normalizado aceptado para mallas $8 \times 8 \times 8$ para una distribución uniforme del destino de los mensajes. . . . .	90
3.43	Latencia media del mensaje vs. tráfico normalizado aceptado para toros $16 \times 16$ para una distribución uniforme del destino de los mensajes. . . . .	91
3.44	Latencia media del mensaje vs. tráfico normalizado aceptado para toros $8 \times 8 \times 8$ para una distribución uniforme del destino de los mensajes. . . . .	92

---

3.45	Desviación estándar de la latencia vs. tráfico normalizado aceptado en un toro $8 \times 8 \times 8$ para una distribución uniforme del destino de los mensajes	92
4.1	Grafo de flujo de datos para calcular $N!$	95
4.2	La máquina MIT de flujo de datos.	96
4.3	Ejemplo de máquina de flujo de datos estática.	97
4.4	Ejemplo de máquina de flujo de datos dinámica.	98
A.1	Clasificación de Flynn de las arquitecturas de computadores.	101
A.2	Clasificación de las arquitecturas paralelas.	102
A.3	El modelo UMA de multiprocesador.	103
A.4	El modelo NUMA de multiprocesador.	104
A.5	El modelo COMA de multiprocesador.	104
A.6	Diagrama de bloques de una máquina de flujo de datos.	106
A.7	Paralelismo de control.	109
A.8	Ejemplo de paralelismo de control.	110
A.9	Paralelismo de datos.	110
A.10	Ejemplo de la aplicación del paralelismo de datos a un bucle.	111
A.11	Paralelismo de flujo.	111

