

Memoria Cache



Iker Rejano Martínez
Jonathan Echeverría Domínguez

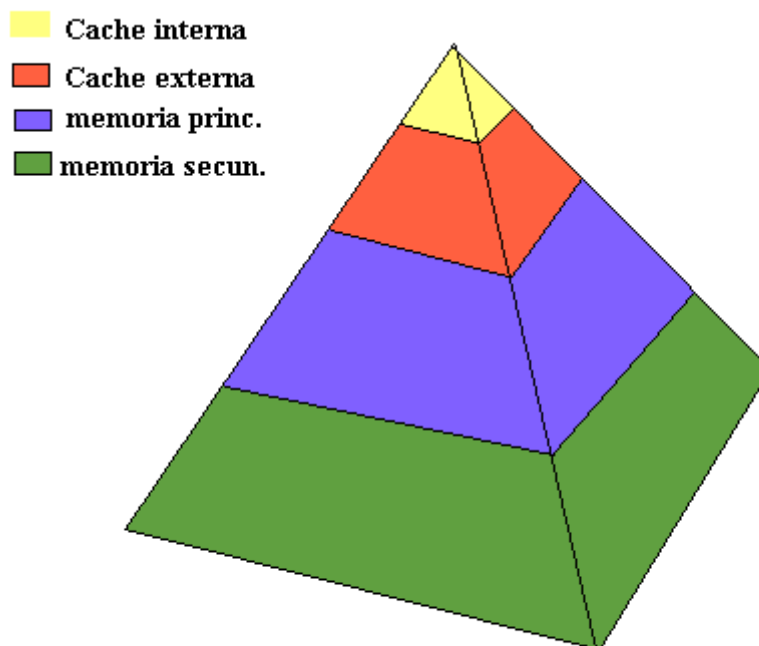
INDICE

1	Introducción Memorias Cache.....	3
2	Cache AMD.....	5
2.1	AMD 32 bits: Athlon y Duron.....	5
2.1.1	Micro-arquitectura AMD Athlon-Duron x86.....	5
2.1.1.1	Instrucciones de la cache.....	6
2.1.1.2	Cache de datos.....	6
2.1.1.3	Unidad LSU (Load-Store Unit).....	6
2.1.2	AMD Athlon-Duron x86 con L2 on-chip.....	7
2.1.2.1	Ventajas de una arquitectura dedicada.....	7
2.1.2.2	Parámetros que definen el rendimiento.....	8
2.1.3	Procesador AMD Duron.....	11
2.1.4	Resumen.....	11
2.2	AMD 64 bits: Athlon y Opteron.....	13
2.2.1	Micro-arquitectura Athlon y Opteron.....	13
2.2.1.1	El TLB.....	13
2.2.1.2	La unidad Load- Store (LSU).....	15
2.2.1.3	La tabla Branch-Prediction.....	15
2.2.1.4	La Unidad Fetch-Decode.....	15
2.2.1.5	La unidad de control de instrucción.....	16
2.2.1.6	El diagrama de bloques del procesador.....	16
2.2.2	La cache de instrucción L1.....	17
2.2.3	La cache de datos L1.....	18
2.2.4	La cache L2.....	18
2.2.5	Optimizaciones de la cache y la memoria.....	19
3	Cache UltraSparc.....	23
3.1	Micro-arquitectura	23
3.1.1	Descripción funcional.....	23
3.1.2	CPU-Cache Crossbar (CCX).....	24
3.1.3	SPARC Core.....	25
3.1.4	La unidad Load/Store (LSU).....	26
3.1.5	Data Translation Lookaside Buffer (DTLB).....	26
3.2	L2-Cache.....	27
3.2.1	Descripción Funcional de la cache L2.....	28
3.2.1.1	Arbitro (ARB).....	29
3.2.1.2	L2 Tag (etiqueta).....	29
3.2.1.3	L2 Data (scdata).....	30
3.2.1.4	La cola de entrada (IQ).....	30
3.2.1.5	La cola de salida (OQ).....	30
3.2.1.6	Búfer de fallo (MB).....	30
3.2.1.7	Búfer de llenado (FB).....	31
3.2.1.8	Write-back Buffer.....	31
3.2.1.9	Remote DMA Write Buffer.....	31
3.2.1.10	Directorio L2-Cache (DIR).....	31
3.2.1.11	Transacciones de la cache L2.....	31
3.3	Level 1 Instruction Cache.....	32
4	Conclusión.....	33
5	Bibliografía.....	34



1 Introducción Memorias Cache

La memoria cache es un componente fundamental de los procesadores de hoy en día, ya que aporta a la máquina un alto grado de eficiencia en los accesos a memoria para lecturas y escrituras. La función principal de la memoria cache es albergar datos que son usados frecuentemente por el procesador (basándose en el principio de localidad temporal y espacial de los programas), de forma que no se tenga que traer el dato desde memoria principal que es más lenta. Hace unos 10 años los procesadores contaban con una pequeña cache dentro del núcleo del procesador y una segunda cache de segundo nivel que estaba fuera del chip, conectada en la placa base. Actualmente todos los procesadores del mercado incluyen dentro del propio chip del procesador una cache de segundo nivel, la cuál ocupa una gran parte del área del chip. De esta forma se ha conseguido reducir bruscamente los tiempos de latencia de las caches, ya que se accede muchísimo más rápido a una cache integrada con el procesador que a una externa.



Jerarquía de accesos

Generalmente las memorias caches de datos están soportadas por tecnología SRAM (Static Random Access Memory), y son accedidas por el procesador para evitar los altos tiempos de acceso a memoria principal. Para comprobar que no se produzcan errores en los datos almacenados en las caches, suelen tener Códigos de Corrección de Errores (ECC), así como otros bits de control para proporcionar robustez y fiabilidad a los datos almacenados en las caches.

Como se explicará más adelante con procesadores concretos, existen varios parámetros que describen una memoria cache:

- Tamaño total
- Nivel que ocupa
- Tamaño del bloque
- Tamaño de cada entrada
- Si es exclusiva para datos o instrucciones, o si es unificada
- Correspondencias: directa, asociativa por conjuntos o totalmente asociativa.
- Algoritmos de reemplazo: random, FIFO, LRU...
- Políticas de escritura: write-back, write-through...
- Rendimiento / precio
- Prefetch

Cada fabricante se decanta por unos u otros parámetros, en función del producto final que quiera obtener y al uso que se va a dar de esa máquina. Cada procesador lleva asociada una cache específica que se complementa con el juego de instrucciones del procesador, tiempos de latencia, frecuencias y tiempos de ciclo... Es por ello, que el estudio y diseño de las memorias cache va totalmente ligado a la arquitectura del procesador al que van destinadas.



2 Cache AMD

2.1 AMD 32 bits: Athlon y Duron

Los términos que definen el diseño de un computador son la arquitectura, la micro-arquitectura y la implementación del diseño. Así pues, los procesadores Athlon y Duron funcionan bajo una arquitectura x86. Pero lo que verdaderamente nos interesa en nuestro caso es la micro-arquitectura (técnicas de diseño empleadas para conseguir bajo coste, gran funcionalidad y rendimiento), así como el diseño de la implementación empleado (circuitos lógicos, registros, buffers...).

2.1.1 Micro-arquitectura AMD Athlon-Duron x86

Los procesadores AMD están diseñados de forma que los decodificadores operan de forma independiente a las unidades de ejecución, y el núcleo del procesador tiene un número de instrucciones reducido y está implementado con circuitos sencillos que pueden funcionar con altas frecuencias y un ciclo de reloj muy rápido. De esta forma un procesador Athlon es capaz de procesar 3 instrucciones x86 por cada ciclo de reloj con una unidad de control de instrucciones central y dos unidades de procesamiento para integers y floats.

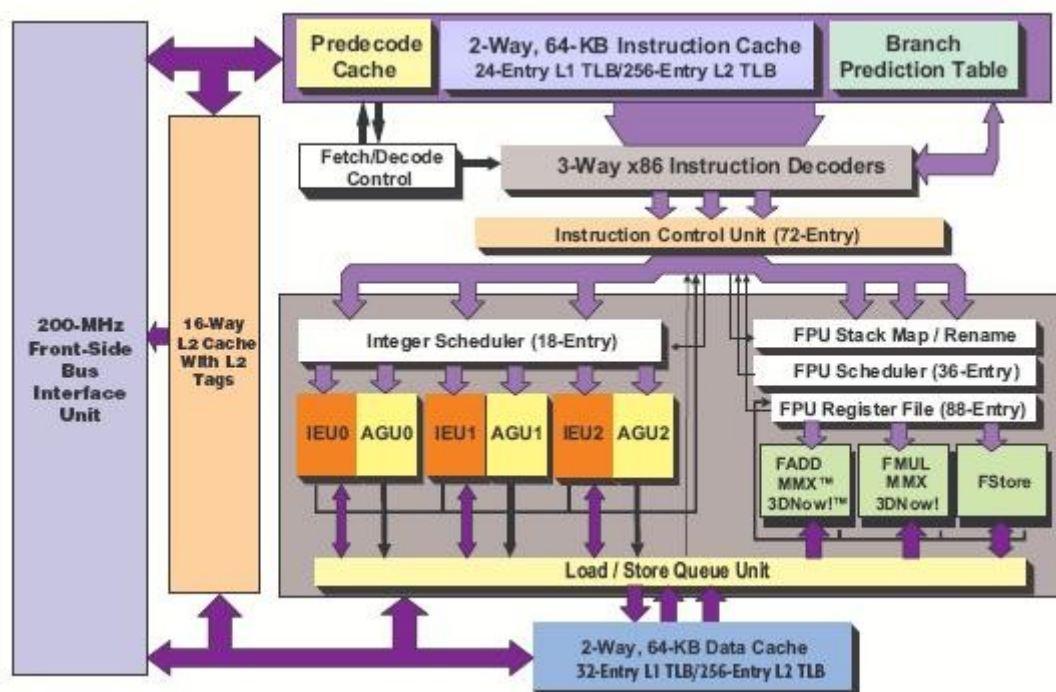


Diagrama de procesamiento de bloque

2.1.1.1 Instrucciones de la cache

La cache L1 es totalmente asociativa con 128kb divididos para instrucciones y datos. Por tanto, para instrucciones tenemos 64kb en total, con entradas de 64bytes de longitud. Estas entradas son reemplazadas mediante LRU (least recently used). En este nivel se tienen las instrucciones cargadas, así como las pre-seleccionadas, las pre-codificadas y las apuestas de salto (suelen ser muy efectivos, ya que se basan en el principio de localidad de los programas). Si se produce un fallo, es decir si la instrucción no ha sido traída al L1 ni por carga “ordinaria” ni por prever que va a ser utilizada en breve, se recurre al L2 de cache al que se accede mediante el BUI (Bus Interface Unit).

El L1 tiene dos niveles de traducción, dos TLBs. El primero es totalmente asociativo y contiene 24 entradas (16 entradas para páginas de 4kb y 8 para páginas de 2Mb o 4Mb). El segundo nivel de TLB tiene 4 formas de asociatividad y contiene 256 entradas que pueden direccionar páginas de 4Kb.

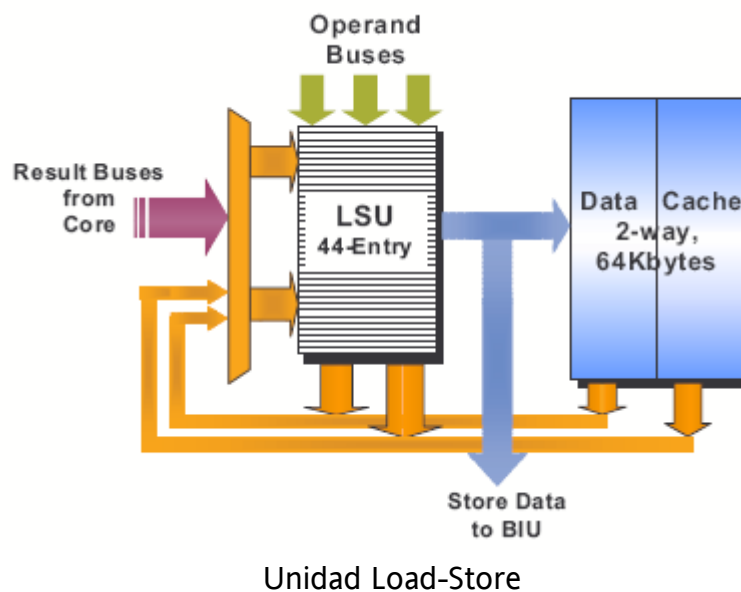
2.1.1.2 Cache de datos

El nivel 1 de la cache contiene dos puertos de 64bits con una escritura directa y una política de escritura en la cache write-back con reemplazo LRU. Los datos soportan 5 bits de estado para mantener la coherencia y paridad de datos entre caches. Estos bits son denominados en inglés MOESI (Modificado, Propio, Exclusivo, Compartido e Inválido).

2.1.1.3 Unidad LSU (Load-Store Unit)

Es una unidad especial que se encarga del control de accesos para carga y escritura de datos a la cache L1 y si no está el dato a la cache L2. Es una unidad intermedia entre el núcleo del procesador, las unidades de coma flotante y enteros, y las caches. Ésta unidad dispone de dos búferes de almacenamiento: uno de lectura y otro de escritura. Cada uno de dichos búferes es capaz de dar respuesta a dos operaciones por ciclo, es decir, dos lecturas o dos escrituras respectivamente.





2.1.2 AMD Athlon-Duron x86 con L2 on-chip

Inicialmente eran procesadores con memoria de 128kb de nivel1, dentro del núcleo del procesador, y con una memoria externa, instalada en la placa base, de nivel 2 de 512kb que ofrecía velocidades superiores a la mitad de la velocidad del núcleo del procesador.

Posteriormente integraron el 2º nivel de cache dentro del propio procesador. De esta forma, la nueva gama de procesadores AMD conseguían velocidades máximas de rendimiento que multiplicaban por 3 la de los Athlon anteriores. Las caches empleadas para ello sufrieron un ligero cambio para dar una gran mejora en cuanto al rendimiento. Dados los problemas existentes de integración dentro del propio procesador (una cache de nivel2 ocupa un 20% del procesador) y el precio económico de una cache, no era posible introducir caches de gran tamaño.

Por ello, se mantuvo 128kb (64kb para datos y 64kb para instrucciones) en el L1, mientras que el segundo nivel se redujo a la mitad: 256kb, dando al sistema 384kb de memoria cache en total. Con esos 384kb de memoria cache totales, se conseguía un rendimiento muy superior, al que nos daban los 640kb de cache instalados en los AMD anteriores. Esto se debe a que se puede conseguir dar respuesta a las peticiones rápidas del procesador, ya que casi hay una velocidad de 1 a 1, debida sobre todo a la integración del nivel 2. La integración del nuevo nivel supone un pequeño problema espacial dentro de la placa de procesador, ya que hay que emplear un 20% de la base.



2.1.2.1 Ventajas de una arquitectura dedicada

Los procesadores athlon dieron un cambio en cuanto al diseño de la arquitectura del PC. Hasta el momento, las memorias cache se dividían en las de primer nivel (integradas dentro del procesador) y las de segundo nivel que iban en la placa madre. Con este nuevo diseño de arquitectura, el segundo nivel de cache también es integrado dentro del procesador.

Con una arquitectura exclusiva, la cache de segundo nivel contiene solamente bloques modificados o víctimas, que van a ser escritos en memoria principal empleando el algoritmo write-back. De este modo, dichos bloques hacen referencia a bloques de cache que han sido previamente mantenidos en el nivel 1, y que tienen que ser reemplazados por un nuevo bloque. Para ello el nivel 2 tiene 256kb, mientras que el nivel 1 tiene 128kb.

El objetivo de una arquitectura exclusiva para la cache, es dedicar el nivel 1 para almacenar los datos que más frecuentemente son usados para que puedan ser accedidos fácilmente por el procesador, proporcionando con ello una gran tasa de aciertos, definida como el porcentaje de tiempo que requiere un dato para ser encontrado en la cache, frente a lo que se tardaría si tuviera que ir a memoria principal. Contra mayor es la cache de primer nivel mayor es la tasa de aciertos, y más rendimiento se saca al procesador, ya que no tiene que ir a memoria. El problema es que contra mayor es la cache, mayor es el precio del procesador. Cuando se produce un fallo en la cache de primer nivel, no hay que ir a memoria principal, si no que se recurre a la cache de segundo nivel a por el dato, que es de suponer que se encontrará allí por dos motivos: la localidad espacial de los programas, y que al ser de mayor tamaño es capaz de albergar mayor número de bloques. La cache de segundo nivel no gasta espacio por los contenidos duplicados de la cache de primer nivel, dada la arquitectura exclusiva.

2.1.2.2 Parámetros que definen el rendimiento.

2.1.2.2.1 Latencia

El AMD Athlon proporciona una mayor velocidad de respuesta ante la petición de un dato, respecto a sus predecesores. Esto se debe básicamente al aumento del tamaño de la cache de primer nivel y separación entre datos e instrucciones, y de dedicar una arquitectura exclusiva a la cache de segundo nivel



dentro del propio procesador. Los datos que son encontrados en la cache son usados más pronto que los residentes en memoria principal, con lo que la latencia disminuye, dando un rendimiento mayor al sistema.

Integrando el segundo nivel dentro del procesador se reduce significativamente la latencia, ya que los tiempos de transferencia de datos dentro del chip del procesador son muchísimo menores, que en una lectura de memoria principal que está en la placa madre.

Según un estudio de AMD, con este tipo de arquitectura de caches embebidas, se consigue reducir la latencia en un 45% respecto a los sistemas de cache anteriores. Dicho estudio afirma que las latencias no deben ser medidas como se median tradicionalmente: probando los peores casos que se pueden dar para ver como responde la cache. Según AMD, dichos tests no son fiables, puesto que esas situaciones son demasiado improbables en la realidad. En sus tests lo que prueban es la latencia del L2 para programas habituales, bucles con una localidad espacial lógica y "normal"... Ahí es donde se notan las grandes ventajas que aporta la introducción de un segundo nivel de cache dentro del procesador.

El sistema de buffer para bloques víctimas contiene los datos que han sido quitados del nivel 1. Presenta ocho entradas de 64 bytes, donde cada entrada se corresponde con una línea de la cache de 64 bytes (L1), es decir, emplean correspondencia directa. Cabe destacar que las caches L1 y L2 del AMD operan con de 64 bytes, el doble que las caches del Pentium III (32bytes).

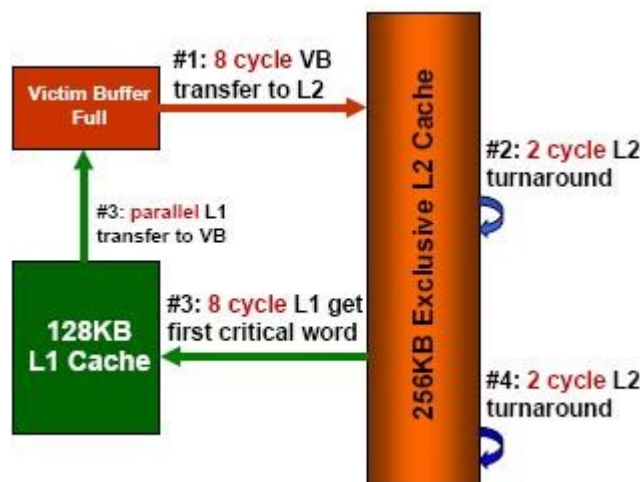
Con un tamaño de 128 el KB, la cache L1 del Athlon es capaz de satisfacer más demandas múltiples para los datos. Como consecuencia, el buffer un elemento clave en la reducción de los tiempos de latencia. En aplicaciones del mundo real, el buffer de bloques víctimas raramente se llena, por lo que es capaz de dar en casi todos los casos cobertura al primer nivel. Así pues, se estima la latencia de uso de carga del segundo nivel de cache en 11 ciclos, que incluyen los tres ciclos cuando se produce un fallo en L1. La cache L2 provee la primera palabra crítica hacia el primer nivel con una latencia de 8 ciclos, es decir, tarda 8 ciclos desde que se le solicita la palabra crítica hasta que la entrega.





Buffer de bloques víctimas no lleno

Pero aun así, hay ocasiones en que el buffer se llena y los tiempos de latencia suben, ya que hay que traer el bloque desde el L2. La siguiente figura muestra el caso en que se produzca un fallo en el nivel 1 de cache y que ese bloque se encuentre en el nivel 2, estando el buffer de víctimas lleno. Los pasos que se siguen ante esta situación son:



#1: El bloque es copiado desde el buffer al L2, ya que tenemos write-back y hay que actualizar. Tiempo necesario para ello: 8 ciclos.

#2: Se comprueba si el bloque se encuentra en L2. Tiempo empleado: 2 ciclos.

#3: Se transfiere la palabra crítica desde L2 hasta L1. En paralelo se transmite desde L1 al buffer de víctimas. Tiempo necesario: 8 ciclos.

#4: Se actualiza el segundo nivel (write-through). Tiempo: 2 ciclos.

En resumen, los tiempos de latencia en la arquitectura de caches del Athlon están entre 11 y 20 ciclos de reloj, dependiendo de la actividad del procesador. En usos reales del procesador, se ha estimado que los tiempos medios de latencia son de aproximadamente 11ciclos de reloj. Con el tamaño

establecido de L1, el buffer de bloques víctimas, y la arquitectura dedicada para el nivel L2, se consiguen tiempos de respuesta y rendimiento superiores al Pentium III.

2.1.2.2.2 Ancho de banda de la cache

Con la integración en el procesador de la cache de nivel L2 se consiguió un aumento considerable de la velocidad, así como un gran incremento del ancho de banda. Dicho incremento se estima en un 300% con respecto a los modelos anteriores de AMD (k6). El aumento del ancho de banda permite al procesador trabajar con más datos en un menor tiempo. Por ejemplo, los procesadores anteriores al Athlon necesitaban para transferir 64 bytes (1 línea de la cache), 8 transferencias por 3 ciclos de reloj cada una de ellas, es decir, 24 ns en total. En cambio, en el Athlon se reduce ese tiempo a 8ns en total.

AMD realizó un estudio en el que determinó que no merece la pena de momento expandir el bus de L2 a más de 64 bits, ya que no nos ofrecería grandes ventajas ni mejoras en el rendimiento. Para ello compararon un L2 con 64 bits, y un L2 con 256bits, y apenas se notaba la diferencia de rendimiento, notándose considerablemente el aumento del precio del procesador por la integración de dicho bus. La otra conclusión a la que llegaron fue que en procesadores con un L1 pequeño, y un gran L2 no se producen buenos resultados, puesto que hay una alta tasa de fallos en L1 y hay que ir al segundo nivel constantemente, con la consiguiente pérdida de tiempo que ello conlleva. Por ello, en el Athlon incluye un L1 grande, para proporcionar una alta tasa de aciertos y con ello reducir y minimizar las demandas de bloques que se producen al L2, minimizando el tráfico por el bus.

2.1.2.2.3 Asociatividad

El Athlon tiene un grado 16 de asociatividad en la cache, frente al grado 2 que ofrecían sus predecesores. Aumentando la asociatividad se consigue una mayor tasa de aciertos debido a la reducción de los conflictos entre datos, es decir se consigue que en la cache de primer nivel se tenga capacidad para albergar mayor cantidad de datos, reduciendo con ello la probabilidad de que se tenga que expulsar un bloque que se usa con relativa frecuencia de L1 a L2. Puesto que la memoria principal es limitada y sus tiempos de acceso lentos (133MHz) la cache tiene gran importancia en el sistema: optimiza al máximo el



rendimiento de la memoria principal y reduce los tiempos de obtención de datos del procesador, consiguiendo un alto rendimiento del sistema.

2.1.2.2.4 Buffers de servicio, colas de entradas al bus y buffers de write-back

Los Athlon incluyen 8 buffers de servicio, 8 colas de entrada al bus de lectura y 8 buffers para la escritura write-back. Con ello logran un rendimiento muy superior a los Pentium, que cuentan con 6 para servicio, 8 de lectura y 4 para write-back. El conjunto de estos buffers están dedicados al abastecimiento rápido de datos al procesador, a reducir los tiempos de espera del procesador por un dato, y los tiempos de actualización de bloques en las caches. De esta forma se incrementa notablemente la velocidad de procesamiento de datos del procesador. Con todo ello, el Athlon es capaz de ejecutar instrucciones de un ancho de banda de aproximadamente 1,6 GB/s.

2.1.3 Procesador AMD Duron

La única diferencia con el Athlon es que su cache de segundo nivel es de 64Kb, mientras que la del Athlon es de 256Kb.

2.1.4 Resumen

Los procesadores de AMD Athlon y Duron son los más potentes y los que ofrecen los mayores altos rendimientos del mercado (segunda mitad año 2000).

Con su cache de segundo nivel integrada en el procesador ofrece baja latencia, una arquitectura exclusiva para la gestión de la cache (buffers, buses de 64bits...), y un alto grado de asociatividad. Están destinados tanto a su uso personal como de negocios, software de entretenimiento y multimedia, aplicaciones de tratamiento de vídeo, ya que tiene un rendimiento muy bueno.



2.2 AMD 64 bits: Athlon y Opteron

Al plantear el diseño del procesador, es importante entender los términos de la arquitectura, micro-arquitectura, y la implementación del diseño.

La arquitectura consta de instrucciones y éstas características de un procesador que son visibles para programas software que se ejecutan en el procesador. La arquitectura determina qué software puede ejecutar. Las arquitecturas del AMD64, de los procesadores AMD Athlon 64 y AMD Opteron, son compatibles con las instrucciones estándar de la arquitectura del x86.

El término micro-arquitectura se refiere a las características del diseño, las funciones que desempeña, y las metas de funcionabilidad del procesador. La arquitectura del AMD64 utiliza un diseño de decodificación/ejecución desparejado. En otras palabras, los decodificadores y las unidades de ejecución esencialmente operan independientemente; El centro de ejecución usa un número pequeño de instrucciones y un diseño de circuito simplificado para una rápida ejecución de un solo ciclo y con rápidas frecuencias en las operaciones.

La implementación del diseño se refiere a una combinación particular de elementos físicos lógicos del circuito y que abarcan las especificaciones de la micro-arquitectura que reúne un procesador.

2.2.1 Micro-arquitectura Athlon y Opteron

Los procesadores AMD Athlon 64 y AMD Opteron incluyen muchas características diseñadas para mejorar la ejecución del software. El diseño interno, o micro-arquitectura, de estos procesadores proporciona las siguientes características significativas:

- La cache de instrucción L1 de 64 KBytes y la cache de datos de 64 KBytes.
- La memoria cache L2 On-chip.
- El predecodificado de la instrucción y predicción del salto durante el llenado de la cache.
- Núcleo desparejado de decodificación/ejecución.
- El controlador integrado de memoria DDR.
- Tecnología HyperTransport.

...



2.2.1.1 El TLB

El búfer de translation-lookaside (TLB) es una memoria cache especial on-chip que administra en una tabla los emparejamientos de la mayoría de las direcciones virtuales recientemente usadas con sus direcciones físicas. El AMD Athlon 64 y AMD Opteron utilizan una estructura TLB de dos niveles. Un filtro en el AMD Athlon 64 y AMD Opteron elimina los registros innecesarios del TLB al cargar el registro CR3.

En las siguientes tablas podemos ver el tipo de asociatividad de los distintos TLB, así como el número de entradas que corresponde a cada memoria.

A.-) Especificaciones del TLB de la cache de instrucción L1

La tabla proporciona las especificaciones del TLB de instrucción de L1 para diversos procesadores AMD.

Processor Name	Family	Model	Associativity	Number of Entries	
				2-Mbyte Pages ¹	4-Kbyte Pages
AMD Athlon™ XP Processor	6	6	Full	8	16
AMD Athlon™ 64 Processor	15	All	Full	8	32
AMD Opteron™ Processor	15	All	Full	8	32

Note:
1. The number of entries available for 4-Mbyte pages is one-half this value (4-Mbyte pages require two 2-Mbyte entries).

Especificaciones del TLB de la L1I

B.-) Especificaciones del TLB de cache de datos L1.

La tabla proporciona las especificaciones del TLB de datos de L1 para diversos procesadores AMD.

Processor Name	Family	Model	Associativity	Number of Entries	
				2-Mbyte pages ¹	4-Kbyte pages
AMD Athlon™ XP Processor	6	6	Full	8	32
AMD Athlon™ 64 Processor	15	All	Full	8	32
AMD Opteron™ Processor	15	All	Full	8	32

Note:
1. The number of entries available for 4-Mbyte pages is one-half this value (4-Mbyte pages require two 2-Mbyte entries).

Especificaciones del TLB de la L1D

C.-) Especificaciones del TLB de la cache de L2



La tabla proporciona las especificaciones en el L2 TLB para procesadores diversos AMD.

Processor Name	Family	Model	Associativity	Number of Entries (4-Kbyte Pages)
AMD Athlon™ XP Processor	6	6	4 ways	256
AMD Athlon™ 64 Processor	15	All	4 ways	512
AMD Opteron™ Processor	15	All	4 ways	512

Especificaciones del TLB de la L2

2.2.1.2 La unidad Load- Store (LSU)

La unidad load-store es mostrada en la siguiente figura. Maneja la carga de datos y almacena los accesos para la cache de datos L1 y, si necesario, para la cache L2 o para la memoria del sistema. Las 44 entradas de la LSU proporcionan una interfaz de datos para los scheduler de enteros y de coma flotante. Consta de dos colas, una cola de 12 entradas para los accesos de carga y almacenamiento de cache L1 y una cola de 32 entradas para la cache L2 o los accesos de carga y almacenamiento de la memoria del sistema. La cola de 12 entradas puede solicitar un máximo de dos operaciones de la cache L1 (mezcla de loads y stores) por ciclo. Hasta dos stores de 64 bits pueden ser realizados por ciclo. En otras palabras, 16 bytes por ciclo de reloj es la velocidad máxima a la cual el procesador puede transferir datos. La cola de 32 entradas soporta eficazmente las peticiones perdidas en la solicitud de la cache L1 por la cola de 12 entradas. Finalmente, las ayudas LSU aseguran que las normas arquitectónicas de ordenación de carga y almacenamiento son conservadas (un requisito para la compatibilidad de arquitectura del AMD64).

2.2.1.3 La tabla Branch-Prediction

La información almacenada en la tabla de branch-prediction se usa para predecir la dirección de las instrucciones de salto. Cuando las líneas de instrucción de la cache están libres para la cache L2, los selectores de salto y la información predecodificada es también almacenada en la cache L2.

Los procesadores AMD Athlon 64 y AMD Opteron utilizan un "búfer de la dirección branch target" (BTB), un "historial global contador bimodal" (GHBC), y una "pila de dirección de retorno" (RAS) para predecir y acelerar los saltos. Esto incurre solo un ciclo de espera para reencauzar la instrucción. En el caso de una mala predicción, la pena mínima es 10 ciclos.



El BTB es una tabla de 2048 entradas que las caches en cada entrada apuntan a la dirección de un salto. La tabla GHBC de 16384 entradas contiene 2 bits contadores usados para predecir si un salto está “tomado”. La tabla GHBC es indexada usando el resultado (tomada o no tomada) de los últimos ocho saltos condicionales y 4 bits de la dirección del salto. La tabla GHBC da a los procesadores permiso de predecir patrones de salto de hasta ocho divisiones.

Además, los procesadores implementan una pila de dirección de retorno de 12 entradas para predecir direcciones de retorno de una llamada cercana o lejana.

2.2.1.4 La Unidad Fetch-Decode

La unidad fetch-decode realiza la decodificación de instrucciones del AMD64 en macro-ops. Las salidas de los decodificadores guardan las instrucciones en el orden del programa (DirectPath o VectorPath). Esta decodificación produce tres macro-ops por ciclo de cualquier ruta. Las salidas de ambos decodificadores son multiplexadas a la vez y pasan a la siguiente etapa, la unidad de control de instrucción.

Decodificar una instrucción VectorPath puede impedir simultáneamente la decodificación de una instrucción DirectPath. Los bytes de instrucción son examinados para determinar si el tipo de decodificación pertenece a DirectPath o VectorPath.

2.2.1.5 La unidad de control de instrucción

La unidad de control de instrucción (ICU) es el centro de control para los procesadores AMD Athlon 64 y AMD Opteron. Controla el búfer centralizado, el scheduler de enteros, y el scheduler de coma flotante. Además, el ICU tiene a cargo las siguientes funciones: La transferencia de macro-ops, (mandar a ejecución) dispatch macro-ops, registra, señala, renombra y administra los recursos de ejecución, interrupciones, excepciones, y malos pronósticos de salto (apuestas de salto). La unidad de control de instrucción soporta tres macro-ops por ciclo de los decodificadores y las coloca en un búfer centralizado. Este búfer está organizado en 24 filas de tres macro-ops cada uno. La unidad de control de instrucción simultáneamente puede enviar múltiples macro-ops desde el búfer (reordenado) para ambos planificadores (scheduler de entero y de coma flotante) para la decodificación final y la ejecución como micro-ops. Además, la unidad de



control de instrucción trata excepciones y administra la retirada de macro-ops.

2.2.1.6 El diagrama de bloques del procesador

En el diagrama de bloques de los procesadores AMD Athlon 64 y AMD Opteron mostrado en la siguiente figura podemos ver los dos niveles de cache.

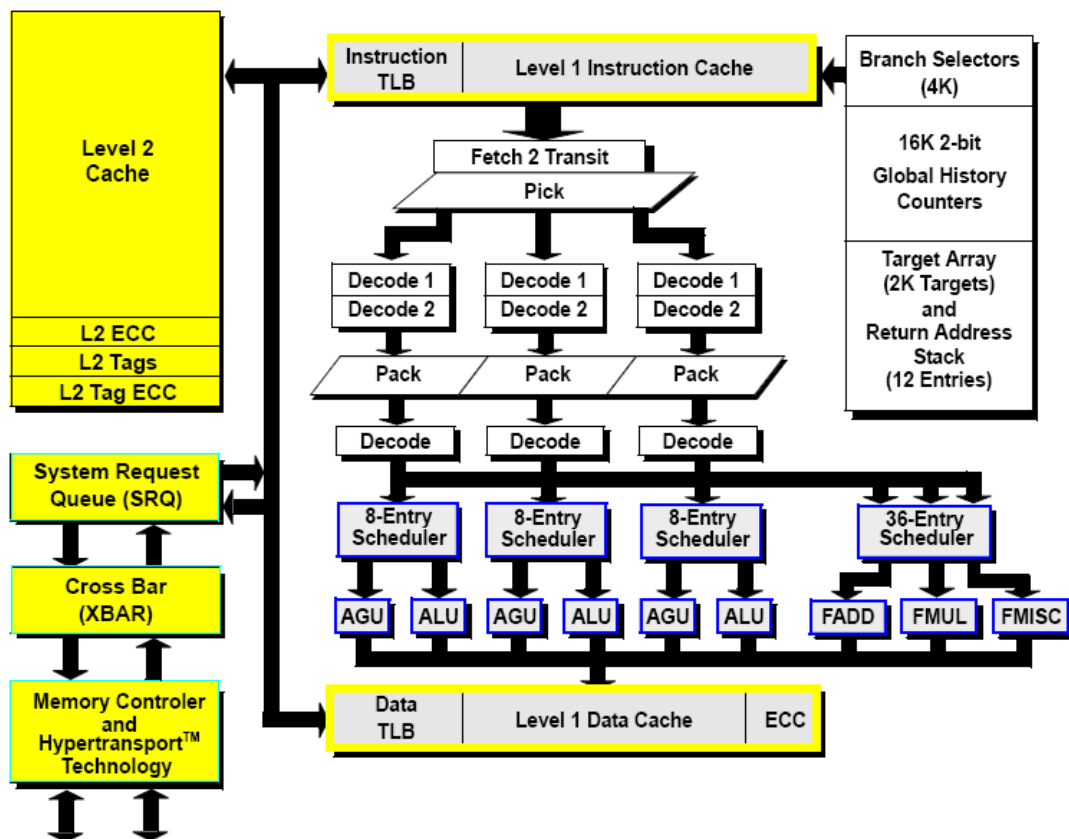


Diagrama de procesamiento de bloque

2.2.2 La cache de instrucción L1

El motor de ejecución de los procesadores AMD Athlon 64 y AMD Opteron contiene una importante cache de instrucción L1. Cada línea en esta cache tiene 64 bytes de largo.

Las funciones asociadas con la cache de instrucción L1 son cargas de instrucción, prefetch de instrucción (lectura anticipada), predecodificado de

instrucción, y las apuestas de salto.

Las peticiones que se pierden en la cache de instrucción L1 son retomadas desde la cache L2 o, posteriormente, desde la memoria local usando el controlador integrado de memoria.

La cache de instrucción L1 genera unas lecturas (fetchs) en los 64 bytes naturalmente alineados, conteniendo las instrucciones y la siguiente línea secuencial de 64 bytes (prefetch). Siguiendo el principio de localidad de programa-espacial hace el código prefetching muy efectivo y evita o reduce ejecución causada por la cantidad de tiempo requerido para leer el código necesario. El reemplazo de la línea de cache se basa en un algoritmo de reposición de least-recently-use ("uso menos reciente").

La siguiente tabla muestra las especificaciones de la cache de instrucción L1 para diversos procesadores AMD.

Processor name	Family	Model	Associativity	Size (Kbytes)
AMD Athlon™ XP processor	6	6	2 ways	64
AMD Athlon™ 64 processor	15	All	2 ways	64
AMD Opteron™ processor	15	All	2 ways	64

En ésta tabla, así como en la siguiente (L1 cache de datos) con la misma información podemos ver el tipo de correspondencia de las caches de los tres tipos de procesadores y su tamaño, además de la familia y modelos que corresponde la información. En los tres casos es asociativa 2 vías (conjuntos) y su tamaño es de 64 KBytes.

El predecodificado (Predecoding) comienza a medida que la cache de instrucción L1 está llena. La información predecodificada es generada y almacenada a lo largo de la cache de instrucción. Esta información se usa para ayudar eficientemente a identificar los límites entre la longitud de las variables de instrucción de los AMD64.

2.2.3 La cache de datos L1

La cache de datos L1 contiene dos puertos de 64 bits. Uno es de write-allocate y otro writeback que usa la política del reemplazo de least-recently-used (menos recientemente usada). Está dividido en 8 bancos, cada uno de 8 bytes.



Además, la cache L1 da soporte a MOESI protocolo cache-coherency y a la paridad de datos (Modified, Owner, Exclusive, Shared, and Invalid).

La tabla proporciona especificaciones de la cache de datos L1 para procesadores diversos AMD.

Processor name	Family	Model	Associativity	Size (Kbytes)
AMD Athlon™ XP Processor	6	6	2 ways	64
AMD Athlon™ 64 Processor	15	All	2 ways	64
AMD Opteron™ Processor	15	All	2 ways	64

Especificaciones de la cache de datos L1

2.2.4 La cache L2

Los procesadores AMD Athlon 64 y AMD Opteron contienen una memoria cache integrada L2. Esto nos presenta una arquitectura exclusiva de cache. La cache L2 contiene solo víctimas o bloques de cache duplicados que deben escribirse al subsistema de memoria como consecuencia de un fallo de conflicto. Estos términos, víctima o duplicados, se aplican a los bloques de la memoria cache que estuvieron previamente alojados en la cache L1 pero que han tenido que ser sobre-escritos (desalojados) para hacer sitio a datos más nuevos. El búfer de la víctima contiene los datos de los desalojados de la cache L1.

La cache L2 en el AMD Athlon XP, AMD Athlon™ 64, y procesadores AMD Opteron tiene un grado de asociatividad de 16.

El tamaño de la cache L2 ha aumentado respecto a su predecesor de 32 bits a 1MByte. Esta modificación se traduce en una mayor tasa de aciertos, debido a que la probabilidad de que la CPU encuentre en este subsistema de memoria la palabra buscada es a priori mayor que si dispone de 512 KBytes.

Para comprenderlo mejor, solo debemos recordar que, según el principio de localidad de las referencias, cuando se transfiere a la cache un bloque de datos desde la memoria principal para satisfacer la petición de una palabra por parte de la CPU, es probable que la siguiente petición haga referencia a una palabra almacenada en el mismo bloque. Lógicamente, en alguna ocasión será necesario actualizar el contenido de la cache ante la ocurrencia de un error



conocido como "fallo de cache". No obstante, es sencillo percibir que cuantas menos operaciones de refresco de esta rápida memoria sea necesario llevar a cabo, mayor será el rendimiento.

2.2.5 Optimizaciones de la cache y la memoria

Las optimizaciones recogen las ventajas de las caches L1 y del gran ancho de banda de los buses de los procesadores AMD Athlon 64 y AMD Opteron.

1. Alineamiento natural de objetos de datos.

Un objeto es naturalmente alineado si esta localizado en una dirección que es múltiplo de su tamaño.

Una palabra en una dirección divisible por 2, una doble-palabra por 4, una Cuádruple-palabra por 8,....

2. Cache-Coherencia Nonuniform Accesos a memoria (ccNUMA).

Para aplicaciones con hilos múltiples, usan las funciones del SISTEMA OPERATIVO para ejecutar un hilo en un nodo particular y dejar ese hilo asignar la memoria que precisa a fin de que la memoria usada sea local para ese nodo. En el entorno Microsoft Windows, la función para ejecutar un hilo en un nodo particular es `SetThreadAffinityMask ()`.

Todas las versiones de Microsoft Windows XP para AMD64 y Windows Server para AMD64 están correctamente configuradas para soportar ccNUMA.

3. Incompatibilidades de tamaño-memoria.

Evita incompatibilidades de tamaño-memoria cuando diferentes instrucciones operan con los mismos datos. Cuando una instrucción almacena y una siguiente instrucción carga los mismos datos que la anterior, se guarda la información de los operandos alineados (loads/stores, tamaño...).

4. Consideraciones del multiprocesador.

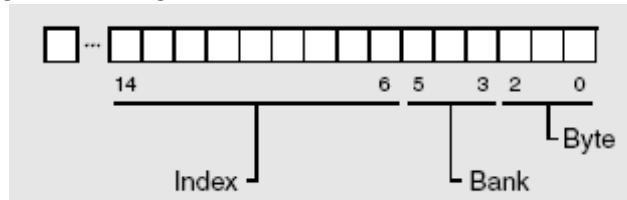
En un sistema del multiprocesador, los datos en una sola línea de cache que es compartida entre procesadores puede reducir el rendimiento. En ciertos casos (por ejemplo, los semáforos), este tipo de uso compartido de datos de la línea cache no puede ser evitado, pero es (o debería ser) minimizado dentro de lo posible. Los datos a menudo son reestructurados para evitar que esto ocurra.

5. Conflictos de banco de la cache de datos L1.

Se utilizan cargas pares que no tienen conflicto para mejorar la carga.



Campos usados para direccionar el multibanco de la cache de datos L1.
La cache de datos L1 es un diseño multibanco con un total de 8 bancos, donde cada banco es de 8 bytes de ancho. Para poner la dirección en la cache de datos L1, el procesador utiliza campos dentro de la dirección como se muestra en el siguiente diagrama:



Cómo saber si un conflicto del banco existe.
La existencia de un conflicto del banco entre dos cargas vecinas depende de su banco y los valores del índice. En la tabla podemos ver cuando se produce un conflicto de banco:

When the bank is	And the index is	Then a bank conflict
Different	Either the same or different	Does not exist
The same	The same	Does not exist
The same	Different	Exists

En otras palabras, con tipos comunes de datos, los elementos consecutivos de un array no pueden tener conflicto de banco. Si los elementos del array son de 4 Bytes o menos, las dos cargas son para el mismo índice y el mismo banco y no hay conflicto. Si los elementos del array son de 8 Bytes, las cargas son para el mismo índice pero para bancos diferentes y tampoco hay conflicto.

6. Prefetch Instructions.

El uso de instrucciones prefetch (lectura anticipada) incrementa el ancho de banda efectivo de los procesadores AMD Athlon y AMD Opteron.

Streaming- Store Non-Temporal Instructions.

El uso de instrucciones como MOVNTPS y MOVNTQ al escribir en arrays o búferes que no necesitan radicar en cache, da al procesador permiso de realizar una escritura sin primeramente leer los datos de memoria o caches de otro procesador. Esto libera el tiempo necesario para leer una línea de la memoria cache, y también previene el desalojo de datos de la cache que pueden ser necesarios. Ésta es una ventaja significativa de rendimiento. Estas instrucciones están disponibles en la mayoría de compiladores.



7. Write-combining (escritura combinada).

El sistema operativo y los controladores de dispositivos entre otros, se ven favorecidos de las capacidades del write-combining del AMD Athlon 64 y AMD Opteron.

8. Colocación de código y datos en la misma línea cache de 64 Bytes.

Evita juntar código y datos e una misma línea cache ya que en caso de tener que modificar los datos el rendimiento se ve afectado.

9. Ordenación de las variables locales.

Ordena variables locales según su tamaño, tipo....

10. Copiar de memoria.

Para realizar una copia rápida de memoria se llama a la función libc memcopy(), que incluye Windows o herramientas gcc. Esta función presenta optimizaciones para todos los tamaños del bloque y las alineaciones.



11. La organización de Loads y Stores.

Cunado se ejecuta una rutina con instrucciones loads y stores, la organización de la secuencia de dichas instrucciones puede afectar al rendimiento.



3 Cache UltraSparc

El procesador OpenSPARC T1 es el primer chip multiprocesador que implementa completamente el Sun Throughput Computing Initiative. El procesador OpenSPARC T1 es un procesador altamente integrado que implementa la arquitectura SPARC V9 de 64 bits. Este procesador apunta a las aplicaciones comerciales como a servidores de aplicación y servidores de bases de datos.

3.1 Micro-arquitectura

El procesador OpenSPARC T1 contiene ocho núcleos del procesador SPARC, y cada uno tiene soporte hardware para cuatro hilos de ejecución. Cada núcleo SPARC tiene una cache de instrucción, una cache de datos, y una instrucción completamente asociativa más un TLB. Los ocho núcleos SPARC están conectados a la cache de nivel 2 on-chip (L2-Cache).

Los cuatro controladores on-chip de acceso a la memoria dinámica (DRAM) directamente interconectan a los datos DRAM (DDR2 SDRAM). Adicionalmente, hay un controlador de J-Bus on-chip que provee una intercomunicación entre el procesador OpenSPARC T1 y el subsistema de E/S.

3.1.1 Descripción funcional

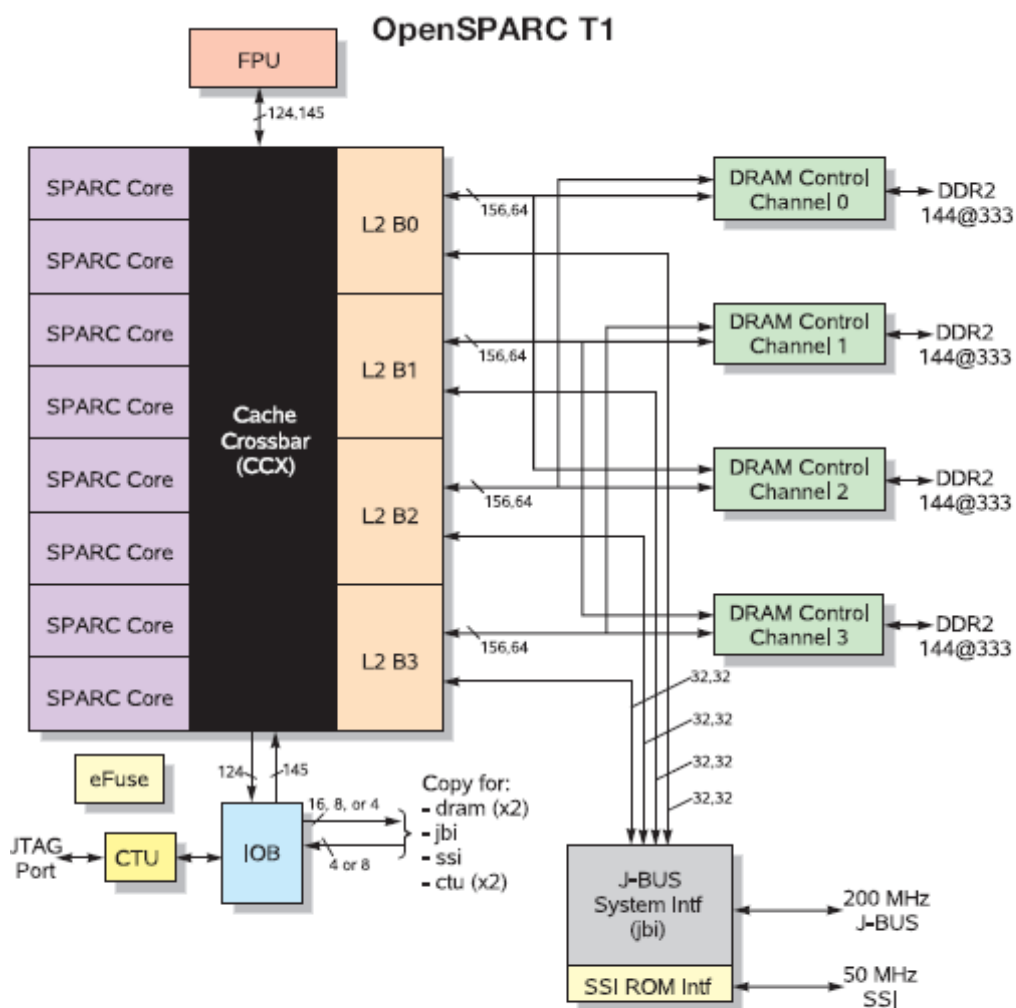
Las características del procesador OpenSPARC T1 incluyen:

- 8 núcleos SPARC V9 CPU, con 4 flujos de ejecución por núcleo, para un total de 32 flujos de ejecución.
- 132 GBytes/sec (crossbar) interconectan para la comunicación on-chip.
- 16 KBytes de cache primaria de instrucción (Level 1) por núcleo CPU.
- 8 KBytes de cache primaria de datos (Level 1) por núcleo CPU.
- 3 MBytes de cache secundaria (Level 2) – en 4 bancos, de 12 vías, es compartida por todos los núcleos CPU.
- 4 controladores DDR-II DRAM - la interfaz de 144 bits por canal, y 25 GBytes/sec de pico del ancho de banda total.
- IEEE 754 la unidad de coma flotante (FPU), compartido por todos los núcleos CPU.
- Interfaces externas:
 - + La interfaz de J-Bus (JBI) para E/S – 2,56 GBytes/sec de pico de



ancho de banda, más un bus multiplexado de dirección/datos de 128 bits.
+ La interfaz Serial System (SSI) para el auto-arranque PROM.

La figura 1 muestra un diagrama de bloques del procesador OpenSPARC T1 ilustrando las diversas interfaces y los componentes integrados del chip. El ancho de los buses está etiquetado con las cifras indicadas en la figura (#IN, #OUT). El tamaño de la imagen no se corresponde con la realidad.



Notes:

- Blocks are not scaled according to physical size!
- Bus widths are labelled as in#,out# where in is into CCX or L2

Diagrama de procesamiento de bloque

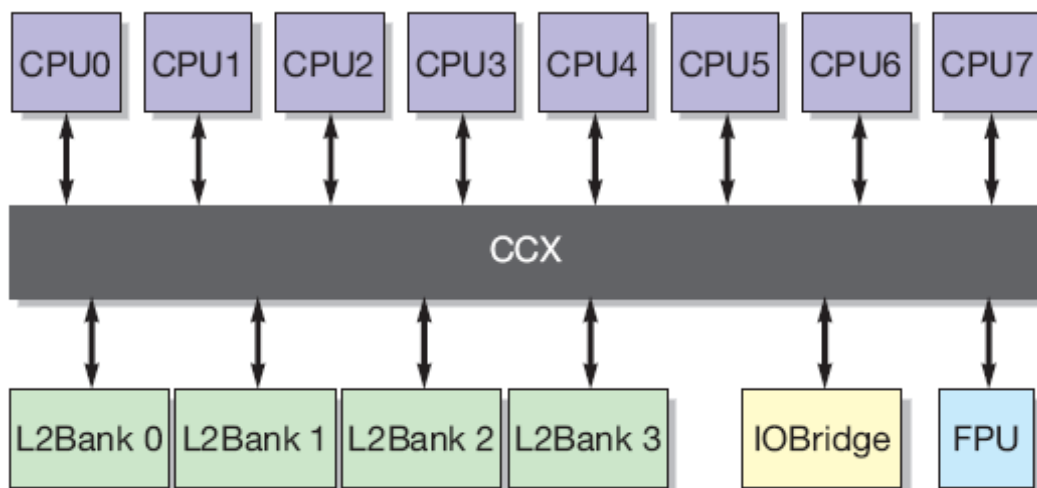
3.1.2 CPU-Cache Crossbar (CCX)



El CPU cache crossbar maneja la comunicación entre los ocho núcleos de la CPU, los cuatro bancos de L2-Cache, el puente de E/S, y la unidad de coma flotante (FPU). Estas unidades funcionales se comunican con cada una enviando paquetes, y el CCX arbitra el reparto del paquete.

Cada núcleo SPARC CPU puede enviar un paquete a cualquiera de los bancos de L2-Cache, puente de E/S, o al FPU. Inversamente, los paquetes también pueden ser enviados en dirección contraria, desde cualquier de los cuatro bancos de L2-Cache, del puente de E/S, o del FPU a cualesquiera de los ocho núcleos de la CPU.

La FIGURA muestra como cada uno de los ocho núcleos SPARC pueden comunicarse con cada uno de los cuatro bancos de L2-Cache, puente de E/S, y FPU.



3.1.3 SPARC Core

Cada núcleo SPARC tiene un soporte hardware para cuatro hilos de ejecución. Este soporte consta de un registro lleno por hilo de ejecución, con la mayoría de los identificadores del espacio de direcciones (ASI), registros auxiliares de estado (ASR), y los registros producidos por hilo. Los cuatro hilos comparten la instrucción, las caches de datos, y los TLBs. Cada cache de instrucción es 16KBytes con un tamaño de la línea de 32 bytes. La política de escritura de las caches de datos es write through (escritura directa), de 8 KBytes y tienen un tamaño de la línea de 16 bytes. Los TLBs incluyen un rasgo "autodemap" que posibilita los hilos múltiples para actualizar el TLB sin cerrar.

Cada núcleo SPARC tiene solo un número de seis fases. Estas seis fases son:

1. Lectura
2. Selección del hilo de ejecución
3. Decodificación
4. Ejecución
5. La memoria
6. Write Back (escritura retardada)

Cada núcleo SPARC tiene las siguientes unidades:

1. La unidad Fetch-Instruction (lectura de instrucción) (IFU), que incluye un sistema de cache de instrucción (Se encarga de gestionar los pc's de los procesos, la cache L1).

2. La unidad de ejecución (EXU) incluye las fases de ejecución del pipeline.

3. La unidad Load/store (LSU) incluye etapas de memoria, write-back, y un sistema de cache de datos.

4. La unidad lógica Trap (TLU).

5. La unidad de procesamiento de corriente (flujo) (SPU) es usada para funciones de aritmética modular.

6. La unidad de gestión de memoria (MMU).

7. La unidad frontend Floating- point (FFU).

3.1.4 La unidad Load/Store (LSU)

El complejo de cache de datos tiene unos datos 8 KBytes, en 4 bloques de direcciones, y el tamaño de la línea de 16 bytes. También tiene una etiqueta exportada de datos. Hay un bit válido de puerto dual (1R/1W) para mantener el estado de la línea cache de inválido o válido. El V-Bite invalida el acceso pero no la etiqueta de datos. El algoritmo de reemplazo que usa es pseudo aleatorio para reemplazar línea de la cache de datos. Las cargas son asignadas, en cambio los stores no son asignados. Los datos TLB operan de manera muy parecida para las instrucciones TLB.

La unidad load/store (LSU) tiene un búfer de 8 entradas por hilo de ejecución, la cual se une en un solo array de 32 entradas, con RAW bypassing. Sólo está permitida una única carga excepcional por hilo de ejecución. Las peticiones duplicadas para la misma línea no son enviadas a la cache L2. La LSU tiene interfaz lógica para interactuar por la CPU, es la cache crossbar (CCX). Esta interfaz realiza las siguientes operaciones:

- Fija las prioridades de las peticiones para las operaciones de: operación en coma flotante (Fpops), streaming (flujo), fallos de instrucción y datos,



etcétera.

- La prioridad de petición: lmiss ldmis stores, {fpu, stream, interrupción}.
- Ensambla paquetes para el processor-cache crossbar (PCX).

La LSU manipula los retornos del PCX y mantiene el orden para las actualizaciones de la cache y las invalidaciones.

3.1.5 Data Translation Lookaside Buffer (DTLB)

El búfer del lookaside de traducción de datos (DTLB) es el TLB para la D-Cache. Las caches DTLB almacenan 64 entradas (most-recently-used) y son completamente asociativas. El DTLB dispone de un puerto CAM y un puerto de lectura-escritura (1 RW). Todos los flujos de ejecución comparten el DTLB. Las entradas de la tabla de traducción de cada hilo son guardadas de las entradas de los otros hilos.

El DTLB da soporte a operaciones de traducción de direcciones de 32 bits.

3.2 L2-Cache

El nivel L2 de la cache opera en 4 bancos de 768 KB, con la selección del banco basada en el bit 7:6 de la dirección física. Cada banco de L2-Cache tiene 1024 sets (líneas). El tamaño de la cache es de 3 MBytes, de 12 vías asociativas con un algoritmo de reemplazo seudo – least recently used (LRU). El tiempo de acceso de descarga es de 23 ciclos para un fallo de cache de datos L1 y de 22 ciclos para un fallo de cache de instrucción L1.

L2-cache tiene un tamaño de la línea de 64 bytes, con 64 bytes interpaginados entre bancos. La latencia en el L2-Cache es 8 ciclos para una carga, 9 ciclos para un fallo (l-miss). Son soportados 16 fallos excepcionales (outstanding) por banco por un total de 64 fallos. El DMA de la E/S está serializado con respecto al tráfico de los núcleos en el nivel L2 de la cache.

La cache L2 es responsable de mantener la coherencia de la cache L1, conservando una copia de todas las “etiquetas” de la cache L1 en una estructura directorio.

La coherencia y el ordenamiento en el L2-Cache están descritos como:

- Las cargas actualizan el directorio y llenan la cache L1 en el retorno.
- Los stores no son asignados en la cache L1.



- Hay dos tipos de stores: total store order (TSO) y read memory order (RMO). Solamente está permitido un store destacado (excepcional) para la cache L2 por hilo para conservar el "store ordering" (TSO). No hay tal limitación en RMO stores.
- Ninguna comprobación de la etiqueta se hace en una inserción del búfer store
- Los stores comprueban el directorio y determinan un salto de L1-Cache
- El directorio envía acuses de recibo del store o invalida al núcleo SPARC.
- El crossbar ordena las respuestas a través de los bancos de la cache.

La cache L2 gestiona los posibles errores que se pueden producir. Para ello utiliza unos registros de control de errores, algunos de éstos podrán ser corregidos por la cache, otros serán incorregibles.

Cada banco de la cache contiene una serie de registros que almacenan la información de los errores. Estos registros son:

- Registro de control L2
- Registro de error habilitado (Error Enable)
- Registro de error de estados L2
- Registro de error de direcciones L2
- Registro de error de inserción L2

Cuando se produce un error, estos registros almacenan dicha información y dependiendo del error se podrá recuperar del error o no.

3.2.1 Descripción Funcional de la cache L2

Cada banco de L2-Cache consta de estos tres sub-bloques principales:

- El sctag (la etiqueta de la memoria cache) contiene el array de etiquetas, VUAD array, el directorio de L2-Cache, y el controlador de la memoria cache.
- El scbuf contiene el búfer write back (WBB), búfer de llenado (FB) y búfer de DMA.
- El scdata contiene el array scdata.

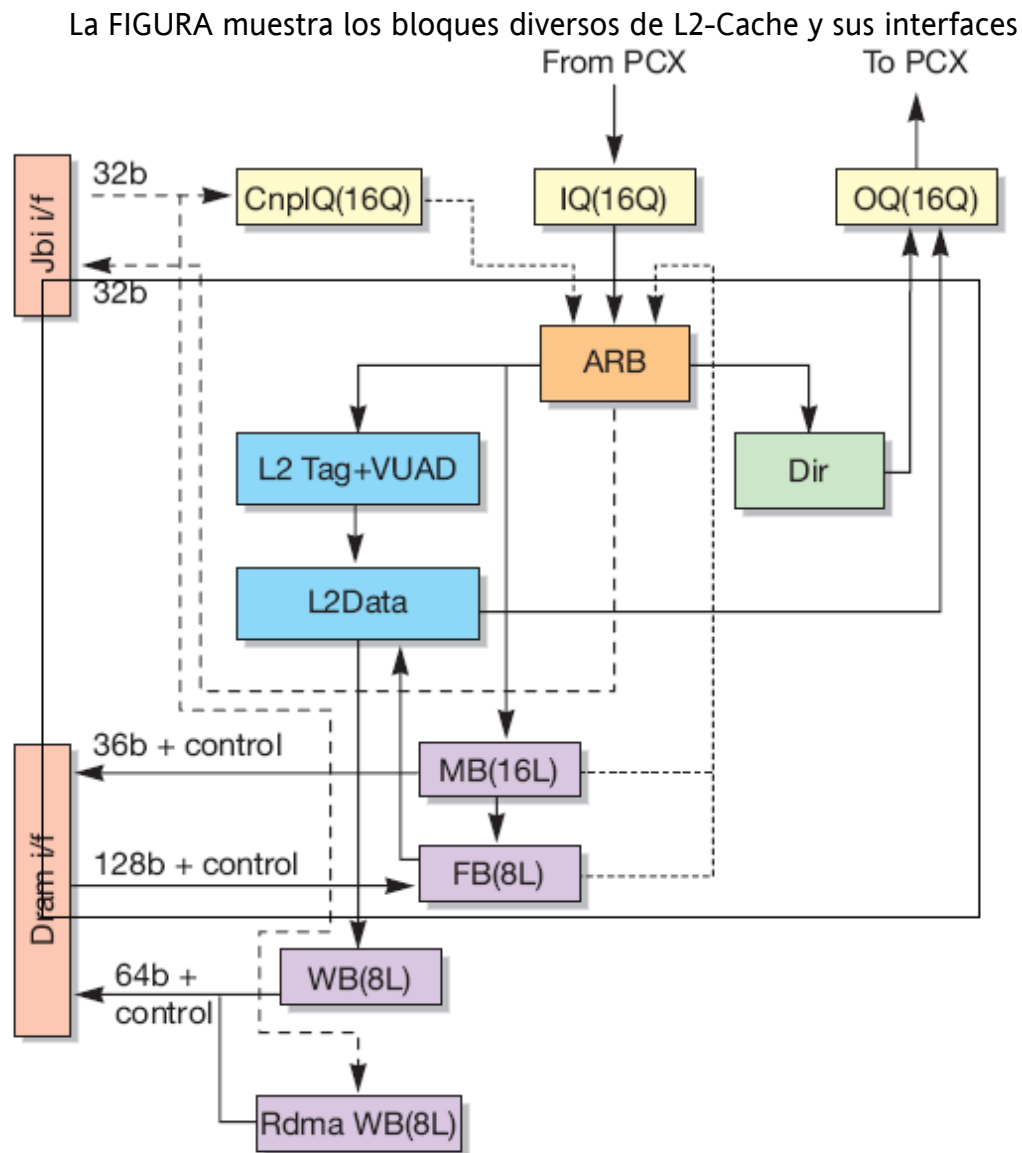


FIGURE 4-1 Flow Diagram and Interfaces for an L2-Cache Bank

3.2.1.1 Arbitro (ARB)

El árbitro controla los accesos a la cache L2 desde distintos componentes que solicitan el acceso.

3.2.1.2 L2 Tag (etiqueta)

El bloque de "etiqueta" L2 contiene el array sctag y la lógica de control asociada. Cada etiqueta de 22 bits está protegida por 6 bits de SEC ECC (la etiqueta L2 no da soporte a la detección de errores de bit- double).



El estado de cada línea es mantenido usando bits válidos (V), usados (U), asignados (UNO), y maliciosos (D). Estos bits son almacenados en el array L2 VUAD.

Las cuatro condiciones son - válido (V), usado (U), ubicado (UNO), y sucio (D). El bit usado no está protegido porque un error usado no causará funcionamiento incorrecta. Los bits VAD son paridad protegida porque un error será fatal. El array L2 VUAD tiene dos puertos de lectura y dos de escritura.

3.2.1.3 L2 Data (sdata)

El banco de datos L2 (array *sdata*) es una sencilla estructura exportada SRAM. Cada banco de L2-Cache es 768 KBytes de tamaño, con cada línea lógica de 64 bytes. El banco permite acceso de lectura de 16 bytes y 64 bytes, y cada línea de cache tiene 16 bytes disponibles para escribir (cada 4 bytes). Sin embargo, se actualización los 64 bytes a la vez.

Cada banco del *sdata* array está además subdividido en cuatro columnas. Cada columna consta de seis sub-arrays 32 KByte.

Cualquier acceso de datos a la L2-Cache lleva dos ciclos para completar el acceso, así que ninguna de las columnas puede ser accedida en los ciclos consecutivos. . El *sdata* array tiene un rendimiento de un acceso por ciclo.

Cada palabra de 32 bits está protegida por siete bits de SEC/DED ECC. (Cada línea es $32 \times 32 + 7 \text{ ECC} = 1248$ bits). Todos los accesos de subpalabra requieren una lectura y una operación de escritura, y son llamados como *parcial stores*.

3.2.1.4 La cola de entrada (IQ)

La cola de entrada (IQ) es de 16 entradas (FIFO) que encola paquetes entrantes en el PCX cuando no pueden ser aceptados en la cache L2. Cada entrada en el IQ es 130 bits de ancho.

3.2.1.5 La cola de salida (OQ)

La cola de salida (OQ) es de 16 entradas (FIFO) que encola operaciones que esperan acceder a la CPX. Cada entrada en el OQ es de 146 bits de ancho.



3.2.1.6 Búfer de fallo (MB)

El búfer de fallo (MB) de 16 entradas almacena instrucciones que no pueden ser tramitadas como un acierto de cache. Estas instrucciones incluyen fallos verdaderos de cache L2, instrucciones que tienen la misma dirección de la línea de cache como un fallo previo o una entrada en el búfer del write-back, instrucciones requieren atravesar la "tubería" de L2-Cache, los fallos no asignados de L2-Cache, y accesos que causan errores "tag ECC".

3.2.1.7 Búfer de llenado (FB)

El búfer de llenado es de 8 entradas (FB) de la anchura de una línea de la cache para almacenar temporalmente datos que llegan de la DRAM antes de que lleguen a la cache. Las direcciones se guardan también para satisfacer condiciones de coherencia.

3.2.1.8 Write-back Buffer

El búfer del write-back (WBB) es un búfer de 8 entradas usado para almacenar la línea de datos "sucia" desalojada, de 64 bytes, de la cache L2. El algoritmo de reemplazo escoge una línea para desalojar en un fallo. Una instrucción cuya dirección de cache es igual a la dirección de una entrada en el WBB es introducida en el búfer de fallos (FB). Esta instrucción debe esperar para que la entrada en el WBB escriba a la DRAM antes de introducirse en la cache L2.

3.2.1.9 Remote DMA Write Buffer

Es un búfer de cuatro entradas dedicado a las transacciones DMA.

3.2.1.10 Directorio L2-Cache (DIR)

Cada directorio de L2-Cache tiene 2048 entradas, con una entrada por etiqueta L1 que asocia a un banco particular de la cache L2. La mitad de las entradas corresponde a la cache de instrucción L1 (icache) y la otra mitad de las entradas es propia de la cache de datos L1 (dcache). El directorio L2 participa de gestión de coherencia del L2-Cache.

El directorio de L2-Cache también asegura que la misma línea no sea



residente en ambos el icache y el dcache (a través de todos CPUs). El directorio de L2-Cache es dividido en un directorio del icache (icdir) y un directorio del dcache (dcdir), cuyos tamaños y funcionalidades son similares.

El directorio de L2-Cache es escrito sólo cuando una carga es realizada. En ciertos accesos a datos (cargas, stores y desalojos), el directorio es "cammed" para determinar si los datos son residente en las caches L1.

3.2.1.11 Transacciones de la cache L2

El L2-Cache procesa tres tipos principales de instrucciones:

- Las peticiones de una CPU por medio del PCX (instrucciones como load, store, streaming load, lfetch, prefetch, store, atomics, interrupt...).
- Peticiones de la / O I por medio del JBI (lectura del bloque (RD64), write invalidate (WRI), y partial line write (WR8)).
- Peticiones desde el IOB (I/O Buffer) por medio del PCX.

3.3 Level 1 Instruction Cache

La cache de instrucción es comúnmente llamada la cache 1 de instrucción nivelada (L1I). El L1I está físicamente indexada y etiquetada y es conjunto de 4 vías asociativas con 16 KBytes de datos. El tamaño de una línea de la cache es 32 bytes. El array de datos L1I tiene un solo puerto, y el tamaño de l-cache es 16 bytes por acceso. Las características de los datos de la cache incluyen - las instrucciones de 32 bits, 1 bit de la paridad, y 1 bit de predecodificado. La etiqueta array también tiene un solo puerto.

Hay un array separado para comprobar el bit válido (V-Bite). Este array V-Bite guarda el estado de la línea de la memoria cache, que puede ser inválido o válido. El array dispone un puerto de lectura y uno de escritura (1R1W). La invalidación de la línea de la cache sólo gana acceso al array V-Bite, y la política de reemplazo de la línea de cache es pseudo-aleatoria.

El acceso de lectura para la l-cache tiene una prioridad superior sobre el acceso de escritura. Los accesos de lectura y de escritura (ASI) a la l-cache son fijados para reducir las prioridades. La terminación de los accesos ASI es oportunista, y hay mecanismo de equidad incorporado para prevenir la inanición del servicio para los accesos ASI.

El período máximo de espera para un acceso de escritura para el l-cache



es de 25 ciclos de reloj del núcleo SPARC. Una espera de más de 25 ciclos de reloj parará el núcleo SPARC, que dará permiso a la I-cache para la terminación del acceso de escritura.



4 Conclusión

Con este trabajo hemos comprendido de forma más precisa y completa el tratamiento que los procesadores dan a los datos. Hemos visto la importancia que tiene la memoria cache en el rendimiento de la máquina y la gran evolución que han tenido con la introducción del nivel L2 on-chip en los últimos 10 años aproximadamente.

Además hemos descubierto todo el hardware complementario a la cache que está integrado en el propio núcleo y que interactúan en las operaciones directamente con la cache (LSU, Branch Prediction, búfers, etc.)

Nos vemos en la necesidad de mencionar las dificultades que hemos tenido para encontrar información sobre los procesadores y más concretamente sobre las caches del mercado actual. Igualmente la otra gran adversidad ha sido disponer únicamente de información en inglés técnico, con las dificultades que ello conlleva para una correcta interpretación.

A pesar de todo ha sido una experiencia positiva y de gran dedicación que nos ha ayudado a aprender mucho más sobre las memorias cache.



5 Bibliografía

- **Transparencias de clase**
- www.amd.com
 - AMD Athlon™ Processor x86 Code Optimization Guide
 - AMD Athlon™ Processor and AMD Duron™ Processor with Full-Speed On-Die L2 Cache
 - Software Optimization Guide for AMD64 Processors
 - IOMMU Architectural Specification
 - AMD Athlon Processor, Technical Brief
- www.sun.com
 - OpenSPARC™ T1 Microarchitecture Specification
 - UltraSPARC T1™ Supplement to the UltraSPARC Architecture 2005
- **Reliability Tradeoffs in Design of Cache Memories**
Hossein Asadi, Vilas Sridharan, Mehdi B. Tahoori, David Kaeli, Northeastern University, Dept. of ECE, Boston MA 02115
- **Pc ACTUAL n° 156**
- **Wikipedia**

