

Índice

1. Introducción

[Historia Antigua, camino a los 64 bits](#)

[Motivaciones para una arquitectura de 64 bits](#)

2. Sun UltraSPARC

[Motivaciones para el diseño de SPARC v9](#)

[Logros debido a cambios en la arquitectura de v8 a v9](#)

[UltraSPARC III - La punta de lanza de la arquitectura SPARC](#)

[El futuro de la arquitectura SPARC](#)

3. PowerPC

[La arquitectura POWER, precursora de la arquitectura PowerPC](#)

[Nacimiento y evolución de PowerPC](#)

[Ideas finales sobre PowerPC](#)

[Gráfico - Evolución de los procesadores PowerPC](#)

[Gráfico - Planes futuros de la PowerPC](#)

4. IA-64 de Intel

[Características de la arquitectura](#)

[Contraste de IA-64 con las arquitecturas actuales de 64 bits](#)

[Características del procesador Itanium](#)

[Ideas finales sobre IA-64](#)

[Gráficos - Benchmarks del Itanium](#)

5. x86-64 de AMD

[Introducción a la arquitectura x86-64 nombre código "Hammer"](#)

[Modos de operación](#)

[Progreso del x86-64 en el tiempo](#)

[AMD Roadmap](#)

[Problemas a enfrentar](#)

[Ideas finales sobre x86-64](#)

6. Conclusiones

7. Bibliografía

[Apéndice: Benchmarks SPEC comparativos de varias arquitecturas](#)

1. Introducción [\(Volver\)](#)

Por más de 15 años la arquitectura de procesadores dominante en el mundo fue la de los procesadores de 32 bit. Si se compara esto con los 2 años que duró la tecnología de 8 bit, 3 de la de 16 bits, nos lleva a cuestionarnos el por qué de este increíble éxito.

Aún más resulta misterioso este hecho si consideramos que aunque varias empresas empezaron a investigar y a realizar las primeras implementaciones de procesadores de 64 bits en los finales de los 80s y principios de los 90s, esta tecnología quedó relegada a un sector pequeño del mercado constituido por los servidores de alta performance (PowerPC, SPARC) y a sistemas incrustados (MIPS) y procesadores de propósito específico (aceleradoras de vídeo, etc).

Con el presente trabajo analizaremos a los sucesores de las arquitecturas de 32 bits con la intención de averiguar el por qué de su éxito, el por qué de la poca penetración en el mercado de las arquitecturas de 64 bits y en general cuál es la tendencia evolutiva del mercado de diseño y construcción de procesadores.

Nos centraremos sobre todo en las arquitecturas de 64 bit más importantes que ya existen como PowerPC de IBM, SPARC del SPARC Consortium, Itanium de Intel y que se encuentran por salir al mercado como la familia Hammer de AMD.

Historia antigua, el camino a los 64 bit [\(Volver\)](#)

- 4 bit: el 4004 designado para un fabricante de calculadoras
- 8 bit: el 4004 engendró al 8008, este al 8080 y este al 8085 y al Z80 de Zilog
- 16 bit: cuando surgió la necesidad de direccionar más de 64k de memoria se creó el 8086. El 8086 engendró al 80286 el cual implementaba “modo protegido”.
- 32 bit: a partir del 286 surgió el 386 el cual fué el primero en tener 32 bit flat addressing que permitía direccionar teóricamente hasta 4Gb de memoria virtual. Es a partir de este procesador que AMD empieza a lanzar sus propios procesadores. El 386 engendró al 486 el cual introducía el co-procesador matemático. A partir de estos se engendraron el Pentium, Pentium Pro, Pentium II, Pentium III que introducían MMX, SSE y otra mejoras. Paralelamente AMD y Cyrix hacían sus propios procesadores x86.

Motivación para una arquitectura de 64-bit [\(Volver\)](#)

La necesidad de procesadores de 64 bits es clara en aplicaciones que necesitan direccionar grandes cantidades de memoria ya sea esta virtual o física. Como ejemplos de dichas aplicaciones podemos citar:

- Bases de Datos: actualmente existen muchas bases de datos con más de 4Gb usados en registros y otra información, esta cantidad constituye el direccionamiento directo máximo posible con 32 bits. Aunque es posible y de hecho a menudo puesto en práctica, el direccionamiento de bases de datos muy grandes con procesadores de 32 bits recurriendo a distintas técnicas, es más fácil y más eficiente hacerlo si el procesador soporta operaciones y direcciones de 64 bits.
- Creación de contenido Digital: el DVD (Digital Versatile Disk) y el HDTV (High Definition TV) cada vez se vuelvan más comunes y el contenido de estos, por los grandes volúmenes de información que requieren, es el principal candidato para la edición en PCs avanzadas. TIVO (dispositivos semi inteligentes de grabación digital) y otros dispositivos similares podrían eventualmente volver obsoleto a las videograbadoras convencionales. En general toda la creación de contenido digital será beneficiada por el direccionamiento y procesamiento de 64 bits.

- CAD & Simulaciones: estos a menudo tratan con modelos enormes(edificios, aviones, explosiones nucleares, etc.) y como es de imaginar dichos modelos pueden fácilmente exceder los 4Gb de tamaño. Como ya se explicó antes cuando el rango direccionable natural del procesador es superado, se deben usar varios trucos para direccionar mas datos, esto implica a menudo una sobrecarga significativa al procesador.
- Criptografía: esta involucra grandes cantidades de cálculos con grandes enteros, y se beneficiaría inmensamente gracias a la disponibilidad de registros de 64 bit lo que le permitiría realizar cálculos mucho más rápidamente.
- Otros: las aplicaciones computacionalmente intensas también se verán favorecidas, por ejemplo:
 - Automatización de diseño electrónico
 - Servidores de alto rendimiento
 - Efectos gráficos como el Transform and Lighting

Todas estas aplicaciones se beneficiarán tanto de las direcciones de 64 bit como de un número mayor de registros.

2. Sun UltraSPARC [\(Volver\)](#)

UltraSPARC es la familia de microprocesadores utilizados para los servidores de Sun Microsystems. Estos microprocesadores, como su nombre lo indica, implementan la arquitectura SPARC (Scalable Processor ARCHitecture).

A diferencia de las arquitecturas IA-64 y x86-64 de Intel y AMD respectivamente, SPARC es una arquitectura abierta, con varios fabricantes que la implementan, y con el UltraSPARC de Sun alcanza su versión 9, desde que fuera lanzada en su primera versión en 1987.

SPARC v9 extiende la arquitectura SPARC a los 64 bits. Sin embargo, el consorcio SPARC ha mantenido la compatibilidad de sus procesadores con versiones anteriores hasta la primera versión, lo que da una reputación de compromiso y buen diseño a esta arquitectura.

Motivaciones para el diseño de SPARC v9 [\(Volver\)](#)

El diseño de los procesadores RISC comenzó en la década de 1980. Los primeros RISC se caracterizaban por arquitecturas tipo load-store, de una instrucción por ciclo y direccionamiento de 32 bits. La ISA (Instruction Set Architecture) de estos primeros chips RISC representaba el estado del arte en materia de poder de cómputo para el mercado de consumidores de los 80's y primera mitad de los 90's.

Sin embargo la demanda por mayor poder de cómputo ha crecido increíblemente en la década pasada, y las aplicaciones del presente y del futuro requieren cada vez más interactividad y distribución a través de múltiples procesadores. Asimismo, esto requiere de mayor espacio de direccionamiento e interfaces de Sistemas Operativos más sofisticadas.

Las arquitecturas del futuro deben proveer mejor soporte para el multiprocesamiento, para lightweight threading (multihilado de baja carga) y programación orientada a objetos. Los sistemas modernos además deben desempeñarse con mayor confiabilidad.

SPARC afirma que en la evolución de las arquitecturas RISC, algunos microprocesadores no han podido proveer ya sea espacios de direccionamiento mayores o mejoras en las funcionalidades. Otros han proveído direccionamiento de 64 bits, pero no han cambiado mucho desde la década de los 80's.

Logros debido a cambios de la arquitectura de v8 a v9 [\(Volver\)](#)

- direccionamiento virtual de 64 bits y datos enteros de 64 bits.
- Mejoras significativas en el rendimiento
- Elaboración de compiladores optimadores avanzados
- Implementaciones superescalares
- Sistemas Operativos avanzados
- Tolerancia ante fallas
- Cambios de contexto y manejo de interrupciones extremadamente veloces
- Soporte para órdenes big-endian y little-endian

Direccionamiento y datos de 64 bits

SPARC v9 soporta directamente direccionamiento virtual de 64 bits y datos enteros de 64 bits. Todos los registros de SPARC v8 se han extendido para manejar la nueva longitud. Naturalmente se han añadido instrucciones para manejar valores de 64 bits. Todo esto se ha hecho manteniendo la compatibilidad con el código ejecutado en v8.

Mejoras significativas en el rendimiento

El rendimiento es una de las mayores preocupaciones de los usuarios y los fabricantes de computadoras. Teniendo en cuenta esto se han agregado 16 nuevos registros de punto flotante de doble precisión, para dar un total de 32. Estos registros adicionales reducen el tráfico de memoria, permitiendo a los programas correr a mayores velocidades. Los nuevos registros de punto flotante también pueden ser operados como 8 registros de cuádruple precisión. El soporte de SPARC v9 para registros de 128 bits es una característica única.

La nueva versión soporta 4 registros de condición para operaciones de punto flotante, a diferencia de la versión 8, que soporta sólo uno. Los procesadores SPARC v9 pueden lograr un mayor grado de paralelismo en máquinas superescalares lanzando varias instrucciones al mismo tiempo. Con un único registro de condición, las instrucciones tendrían una dependencia en serie al esperar la actualización de este registro. La versión 9 permite iniciar hasta 4 comparaciones en simultáneo.

Otras mejoras se lograron mediante:

la aparición de instrucciones para la multiplicación y división con operandos de 64 bits

instrucciones de load y store para operandos de punto flotante de cuádruple precisión

predicción de saltos (branch prediction) configurable en software, lo que da al hardware gran probabilidad de mantener las líneas de procesamiento llenas.

Instrucciones 'move' condicionales, que evitan la realización de muchos saltos.

Elaboración de compiladores optimadores avanzados

Se han incluido muchas características en la versión 9 que permiten al software controlar explícitamente la carga anticipada de datos e instrucciones, reduciendo el retardo de memoria para evitar esperas en el microprocesador, y reduciendo las fallas de paginación del caché.

SPARC v9 tiene soporte para 'cargas a prueba de fallos' (non-faulting loads), que permiten a los compiladores mover las instrucciones de carga fuera de estructuras condicionales de control que verifican su correcto uso. La semántica de los non-faulting loads es la misma que para otros loads, siempre y cuando no ocurra un error del tipo out-of-range. En este caso, dichos errores se ignoran, y el hardware y el software cooperan para hacer aparecer la carga como correcta, devolviendo un valor 0 de retorno.

Esta optimización es particularmente útil a la hora de optimizar procesadores superescalares. Considere este ejemplo:

```
if (p!=NULL) x=*p + y;
```

Con non-faulting loads, la carga de *p se puede mover antes del chequeo (p!=NULL), permitiendo la ejecución solapada. Un load normal hubiera causado que se interrumpiera la ejecución del programa en caso de ser p==NULL. El efecto es equivalente a la siguiente transformación:

```
temp_register=*p;
```

```
if (p!=NULL) x=temp_register + y;
```

Imagine un procesador superescalar que puede ejecutar 4 instrucciones por ciclo, pero solamente una de ellas puede ser un load o store. En un bucle de 8 instrucciones con 2 loads, puede suceder que sin esta transformación no sea posible adelantar alguno de los loads al primer grupo de 4 instrucciones. En este caso un tercer o cuarto ciclos de reloj podrían ser necesarios en lugar de los dos ciclos en el caso óptimo.

La detección de alias es un problema muy difícil de solucionar para los compiladores. Si un compilador no puede determinar si dos punteros apuntan al mismo lugar de memoria o no, entonces no hay libertad para adelantar la carga más allá de cierto punto. Esto puede crear un cuello de botella para el procesador. SPARC v9 tiene instrucciones específicas que ayudan al compilador mediante la detección por hardware de los alias. Dos pointers pueden compararse y su resultado se almacena en un registro entero. La instrucción FMOVZ, por ejemplo, realiza moves condicionales basados en el resultado de esta comparación.

Implementaciones superescalares

SPARC v9 se pensó para soportar implementaciones superescalares avanzadas. Los diseñadores de CPUs aprenden cómo ejecutar más instrucciones por ciclo cada año. Dos o tres instrucciones por ciclo ya son algo común. SPARC se ha diseñado para eventualmente soportar 8, 16 y más instrucciones en simultáneo.

Muchos de los cambios efectuados para lograr esto se deben a la experiencia ganada con la implementación del SuperSPARC de Texas Instruments y del HyperSPARC de Ross Technologies. Las instrucciones simples de decodificar y de longitud fija de SPARC, y unidades de aritmética entera y de punto flotante separadas permitieron la implementación de tecnología superescalar.

Además, v9 provee más registros de punto flotante, soporte para carga a prueba de fallos, códigos de múltiple condición y branch prediction. Todas estas características permiten un mayor nivel de paralelismo en el procesador. Para el sistema de memoria, se cuenta con una instrucción sofisticada de barrera, que permite a los programadores especificar un nivel mínimo de sincronización necesaria para asegurar la correcta operación.

Sistemas Operativos avanzados

La interfaz con el Sistema Operativo ha sido completamente rediseñada. Hay nuevos registros privilegiados y nuevas estructuras para dichos registros que hacen más simple el acceso a información de control.

Varios cambios se han realizado para proveer soporte a sistemas operativos de estilo microkernel. Los niveles de trap (trampa, captura) anidados proporcionan modularidad estructural al código, y son más eficientes. El soporte para hilos de baja carga (lightweight threads) y para los cambios de contexto ha mejorado, proveyendo mayor velocidad.

Las implementaciones SPARC anteriores soportan multiprocesadores. Ahora se ha agregado soporte para multiprocesadores a muy gran escala, incluyendo una instrucción de barrera de memoria, y un nuevo modelo de memoria denominado RMO (Relaxed Memory Order). Estas características permiten ordenar las operaciones con la memoria para lograr alta performance, al mismo tiempo que se realiza la sincronización y operaciones de bloqueo necesarias para multiprocesamiento con memoria compartida.

Tolerancia ante fallas

La mayoría de las arquitecturas de microprocesadores no provee soporte explícito para fallas. Aún así se pueden construir máquinas tolerantes ante fallas, pero son más costosas y difíciles de lograr.

SPARC v9 tiene un número de características para lidiar con este asunto. Primero, cuenta con una instrucción de comparar e intercambiar. Esta función tiene conocidas propiedades que la hacen deseable para sistemas tolerantes ante fallas y para entornos de multiprocesadores.

Cambios de contexto y manejo de interrupciones extremadamente veloces

La entrada a interrupciones para transferir el control a los manejadores de interrupciones ha sido rediseñada para una rápida respuesta. Se han añadido 8 nuevos registros llamados “globales alternas” para que el manejador de interrupción cuente con un conjunto no utilizado de registros para su uso inmediato. El software no necesita salvar registros antes de entrar al manejador de interrupción.

Se ha añadido soporte para interrupciones anidadas, a diferencia de SPARC v8, donde una interrupción no podía generar otra interrupción. Esto simplifica enormemente el código de muchos manejadores de interrupciones y simplifica el diseño de sistemas operativos.

Soporte para órdenes big-endian y little-endian

Finalmente, se ha proveído soporte para datos creados en máquinas little-endian como las x86. La arquitectura permite que el código en modo user y el código en modo supervisor explícitamente accedan a datos en orden little-endian. Es posible incluso cambiar el orden por defecto para el modo user, para el modo supervisor o ambos.

UltraSPARC III - La punta de lanza de la arquitectura SPARC [\(Volver\)](#)



El UltraSPARC III

Características generales

- Diseño de núcleo renovado
- Compatibilidad SPARC hasta las primeras implementaciones
- Arquitectura altamente escalable
- Diseño de circuitería altamente escalable
- Confiabilidad, Disponibilidad y Mantenibilidad mejoradas
- Mejor soporte para Solaris OS
- Nuevo bus de sistema de alta performance

Performance de procesador único

- 650 MHz, 750 MHz disponibles (900 MHz próximamente)
- Arquitectura preparada para 1.5 GHz y más
- Proceso de fabricación a 0.18 micrones

- Interconexiones en aluminio

Performance de varios procesadores

- 4 GB de subsistema de memoria para cada procesador
- El ancho de banda de memoria de procesador se incrementa al incrementarse el número de procesadores
- Soporta hasta 8MB de caché externo.
- Potencial de escalado hasta cientos de procesadores

Características de ejecución de instrucciones

- superescalar de 4 vías
- 6 líneas de ejecución (2 enteras, 2 FP/VIS, 1 para load/store, 1 de direccionamiento)

Componentes integrados al UltraSPARC III

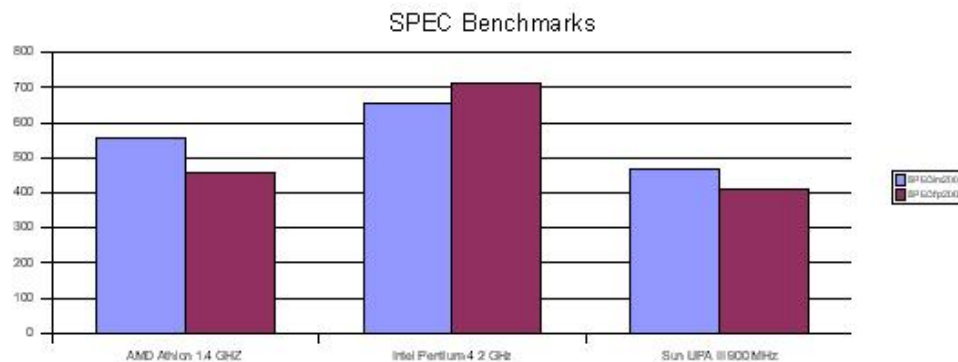
- Caché L1: 64 KB de datos, 32 KB de instrucciones
- L2: circuitería para soportar 1, 2 u 8 MB de memoria externa
- Bus de datos dedicado de 256 bits para el caché L2
- Controlador de memoria SDRAM integrado

El bus de sistema

- 150 MHz de frecuencia de reloj
- Ancho de banda: 2.4 GB/s para CPU a memoria, 4.8 GB/s para bus a memoria
- Arbitraje de bus distribuido

Características físicas

- Cantidad de transistores: 11 millones para lógica, 12 millones para RAM, 6 millones para otros usos
- 1.5V para I/O y 1.7V en núcleo
- Máxima disipación de poder: 70 W a 750 MHz
- Proceso de fabricación: 0.18 micrones (0.15 proyectado)

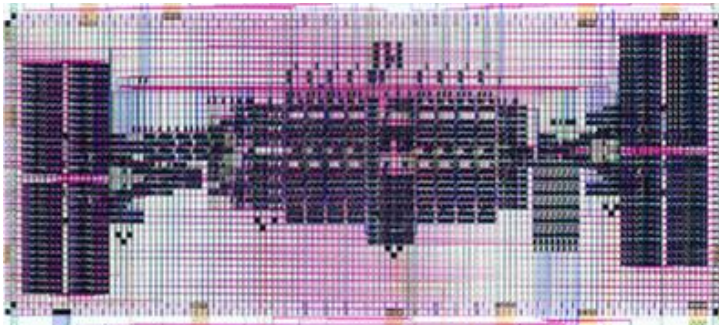


El futuro de la arquitectura SPARC [\(Volver\)](#)

Sun Microsystems, el miembro más sobresaliente del consorcio SPARC, continúa llevando esta arquitectura a nuevas alturas. Su producto estrella en sus servidores Sun Blade 1000, el UltraSPARC III 800 MHz está establecido en el mercado de los servidores para tareas científicas y para e-business.

Sun planea introducir nuevos modelos del UltraSPARC III corriendo a 900 MHz, para hacer frente a los PowerPC de IBM y a la entrada del Itanium al mercado de procesadores de 64 bits. Este nuevo modelo sigue el proceso de fabricación en cobre (copper interconnect technology) a 0.15 micrones, en colaboración con TI (Texas Instruments).

La siguiente generación de UltraSPARCs, el UltraSPARC IV, se planea para finales del 2003, a 0.13 micrones. Luego vendrá el UltraSPARC V, en 0.10 micrones, a finales del 2004.



Núcleo del chip FLEETzero (tamaño real 4 mm x 1.8 mm)

Por otro lado, Sun está investigando arquitecturas asíncronas. En la Conferencia ASYNC 2001, Sun describió el prototipo FLEETzero, con velocidades brutas de alrededor del doble de los chips actuales, pero que funciona sin un reloj maestro. La ventaja principal de este modo de funcionamiento es que el consumo de energía se reduce drásticamente, porque sólo entran en funcionamiento aquellas partes del procesador que son necesarias. El resto permanece inactivo.

3. PowerPC [\(Volver\)](#)

La arquitectura POWER, precursora del PowerPC [\(Volver\)](#)

En febrero de 1990 IBM comenzó la entrega de los RS/6000. IBM desarrolló estos productos para responder a la necesidad de los consumidores por workstations y sistemas de desempeño mediano con sistemas operativos UNIX. Los procesadores en estos productos eran implementaciones de la arquitectura POWER, una arquitectura RISC (Reduced Instruction Set Computer) de segunda generación.

Características de la arquitectura Power

La arquitectura POWER incorporaba características comunes a la mayoría de las otras arquitecturas RISC, las instrucciones eran de tamaño fijo (4 bytes), se proveían GPR (General Purpose Registers) para los cálculos de enteros y FPRs para los de punto flotante. Por otro lado la arquitectura POWER era única entre las otras arquitecturas RISC ya que se encontraba funcionalmente particionada, separando claramente funciones de flujo de programa, calculos de enteros, calculos de punto flotante. Esto facilitaba diseños superescalares en los cuales multiples unidades funcionales (procesadores) ejecutaban concurrentemente instrucciones independientes.

Una segundo diseño de procesador POWER fue hecho posible gracias al avances de la tecnología de silicio. Este diseño RSC (RISC Single Chip) integró varias unidades de procesamiento en un mismo chip, lo que lo hizo apropiado para sistemas desktop de bajo costo.

Nacimiento de PowerPC [\(Volver\)](#)

Hacia 1991, el consumidor empezó a darse cuenta que los sistemas multiprocesadores suministraban valor extra, porque ofrecían una mejor relación precio-performance y mejor performance en general que sistemas con un sólo procesador. Debido a esto Apple, Motorola, e IBM formaron una asociación cuya base era el uso de una arquitectura común derivada de la arquitectura POWER, ya que era imposible empezar con una arquitectura nueva y satisfacer a tiempo las demandas de los clientes.

El la nueva arquitectura debía:

- Permitir un amplio rango de implementaciones desde controladores de bajo costo hasta procesadores de alta performance.
- Ser lo suficientemente simple como para permitir que el ciclo de diseño de nuevos procesadores sea corto
- Minimizar efectos que obstruyan el diseño de implementaciones superescalares.
- Debe soportar múltiples procesadores.
- Definir una arquitectura de 64 bits que incluya a la arquitectura de 32 bits, para así proveer compatibilidad para las aplicaciones de 32 bits permitiendo que los usuarios de sistemas POWER ejecuten sus aplicaciones en el nuevo sistema y obtengan en la mayoría de los casos beneficios de la performance del nuevo procesador.

Finalmente se diseñó la arquitectura PowerPC de 64 bits. Esta arquitectura extiende el direccionamiento y el cálculo de enteros a 64 bit y soporta cambio dinámico entre los modos de 64-bits y de 32-bits donde se permite la ejecución de aplicaciones POWER.

Para llegar a la arquitectura PowerPC y cumplir los objetivos fijados los asociados tuvieron que modificar la arquitectura POWER profundamente, algunos de estos cambios son:

- Se excluyeron algunas instrucciones POWER que eran de escaso uso y que podían ser implementadas mediante instrucciones más básicas.
- Se relajaron requerimientos no críticos
- Se agregaron nuevas instrucciones

Todos estos cambios se hicieron con el objetivo simplificar substancialmente los diseños de los procesadores futuros.

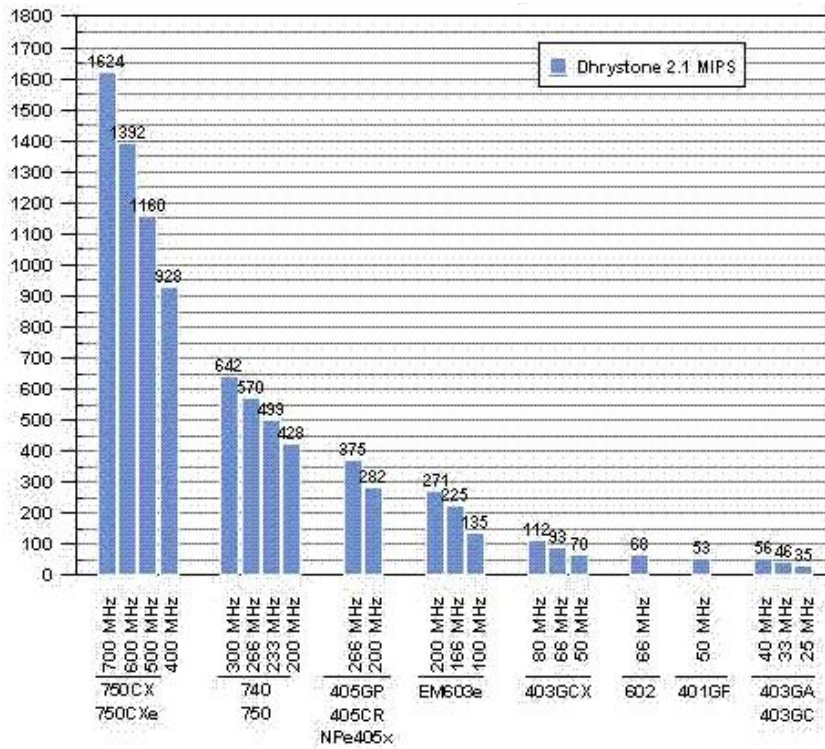
Finalmente ciertas características se agregaron a la arquitectura PowerPC para completarla:

- Memoria: en la arquitectura POWER el procesador no mantenía los datos de memoria consistentes ni con I/O ni con los pedidos de las instrucciones. Era misión del software el manejo de la consistencia de la memoria para estas áreas. El modelo de memoria de la PowerPC por otro lado, provee gran flexibilidad, atributos asociados con cada página de la memoria virtual permiten al software controlar como el sistema realiza los accesos. Este modelo permite el acceso a cualquier página a no ser que tenga un atributo indicando que contiene I/O o exhibe otras características volátiles. Entonces resulta posible mapear I/O en el espacio de la memoria principal. La definición del sistema de memoria también incluía consistencia de los datos de memoria controlada por el procesador, lo que liberaba al software de esta responsabilidad con respecto a las operaciones I/O, pero por otro lado exige al software la manutención de la consistencia de la memoria de instrucciones con la memoria de datos.
- Word aligned: si las instrucciones intentan acceder a una dirección que no se encuentra word-aligned entonces el resultado de esta operación no se encuentra definido. Esto permite implementaciones más simples, evitando la necesidad del control de hardware para evitar un error de programación fácilmente evitable.
- Soporte para Big y Little Endian: aunque el modelo POWER era big endian, una de las adiciones finales al modelo de memoria de la PowerPC fue el soporte para ambas semánticas de memoria, mediante la especificación de un modo controlado por software.

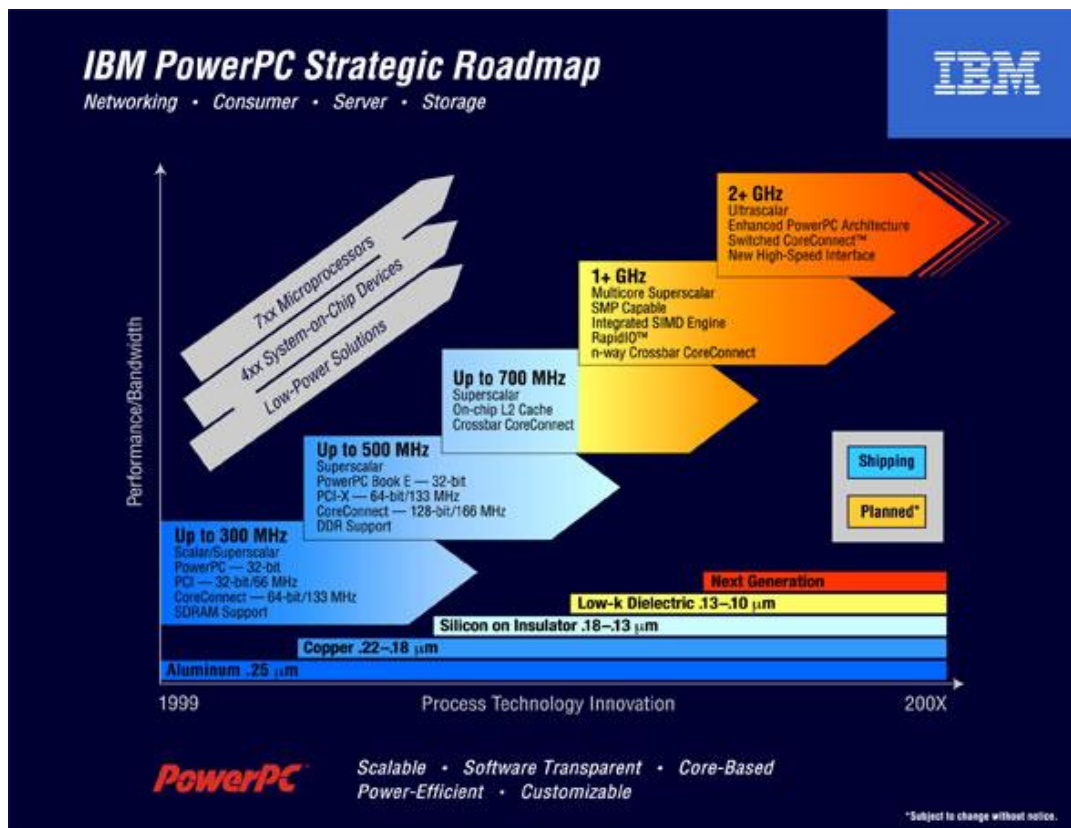
Ideas finales sobre la PowerPC de IBM [\(Volver\)](#)

PowerPC fue desarrollada en 1991 por un grupo de empresas deseando una arquitectura que expandiera e implementara mejor las funcionalidades de la arquitectura POWER. El resultado de este esfuerzo permite un gran rango de implementaciones desde desktops de bajo precio, y notebooks hasta procesadores de alta performance que permiten el uso de múltiples procesadores.

Evolución de los procesadores PowerPC [\(Volver\)](#)



Planes Futuros de la Power PC [\(Volver\)](#)



4. Intel IA-64 [\(Volver\)](#)

IA-64 (Itanium) es la familia de procesadores con la que Intel se introduce al mercado de los procesadores de 64 bits. El primer producto de esta familia de procesadores es, precisamente, el Itanium.

Se ha diseñado para ser utilizado en servidores y workstations de alta performance. Intel afirma que su diseño va más allá de conceptos como CISC o RISC mediante la incorporación de una cantidad masiva de recursos de procesamiento con compiladores inteligentes que permiten la producción de código objeto que hace explícito el paralelismo.

Ya terminó su fase final de producción y se comercializa en pequeñas cantidades. Cuenta con Sistemas Operativos como Windows Advanced Server, Windows XP 64-bit Edition, Linux Trillian, HP-UX y otros. Pero para realizar el cambio con mayor comodidad, el Itanium ejecuta código IA-32 en hardware.

Características de la arquitectura [\(Volver\)](#)

- EPIC - Explicitly Parallel Instruction-Set Computing
- Mejora el ILP (Instruction Level Parallelism) maximizando la complementación de soluciones hardware-software. Provee mecanismos tales como branch-hints, cache-hints (pistas de salto, de caché) para el compilador, para que este indique al procesador de tales eventos y permita que maneje el hardware más eficientemente. Estos mecanismos de aviso minimizan el costo computacional de los saltos y reducen fallas de consulta a la memoria caché. Además de eso, el conjunto de instrucciones en sí cuenta con opcodes que explícitamente permiten su ejecución en paralelo (codeblock-hint).
- Otros conceptos que se manejan con EPIC son:
 - Speculation (Especulación): mejora la performance permitiendo que el compilador adelante las instrucciones de carga debidas a saltos reduciendo el retardo debido a la memoria (memory latency).
 - Predication: elimina muchos saltos y las penalidades debidas a las fallas en la predicción de los saltos (branch prediction).
 - Register Stacking: reduce el overhead introducido por las secuencias de llamada y retorno mediante un modelo eficiente de registros enteros administrados por el RSE (Register Stack Engine).
 - Register Rotation: renombra registros en hardware de manera automática para mejorar la performance de los loops, sin el overhead debido a los métodos tradicionales (loop unrolling).
 - Instrucciones SIMD: mejora la performance de aplicaciones multimedia al operar en varios datos enteros o de coma flotante en una sola instrucción.
 - Cantidad masiva de registros: 128 registros de enteros, 128 registros de coma flotante, 8 registros de salto, 64 de instrucciones.
 - Escalabilidad para 32 y más CPUs en paralelo.
 - Uso optimizado de memoria:
 - tamaños de página de hasta 256 MB
 - tres niveles de caché
 - Compatibilidad con IA-32 en hardware

Contraste de IA-64 con las arquitecturas de 64 bits actuales [\(Volver\)](#)

La predicación (predication) mejora el paralelismo

Los saltos son un problema importante de las arquitecturas RISC que ejecutan código fuera de orden. Estas arquitecturas tradicionales usan una técnica llamada branch prediction (predicción de saltos) para predecir el camino correcto.

Fallas en la predicción son de ocurrencia común, alrededor del 5-10% de las veces, resultando en grandes penalidades en la performance, de hasta 30-40%. IA-64 usa una técnica llamada Predication para ejecutar ambos caminos en paralelo y evitar las limitaciones de las arquitecturas tradicionales.

64 registros de 1 bit se utilizan para poner en práctica esta técnica, 'apagando' (descartando) aquellas ramas en la ejecución que no son parte del camino correcto. IA-64 también provee soporte eficiente para saltos múltiples y comparaciones paralelas, ambas características permiten realizar el salto múltiple en 1 ciclo de máquina, acelerando la ejecución significativamente.

En todos los casos, el registro de predicación indica si una rama en particular y sus datos asociados están 'encendidos' o 'apagados'. Las ramas 'encendidas' continúan su ejecución y ni bien se 'apagan' su ejecución cesa.

Los predicados, que son parte de toda condicional de la IA-64, proveen un mecanismo muy eficiente para terminar la ejecución de ramas que no deben seguirse, para establecer bucles y terminarlos y para administrar la predicción dinámica y realizar comparaciones de n caminos.

Los predicados ayudan a manejar la compleja tarea del flujo de control, compleja debido a un agresivo paralelismo a nivel de instrucciones (ILP – Instruction Level Parallelism) que se realiza en tiempo de compilación.

De esta manera, la predicación reduce las fallas de predicción en los saltos, incrementando la performance. Es particularmente útil en aplicaciones donde es difícil predecir los saltos como en grandes bases de datos, data mining, warehousing, etc.

Como no cuentan con predicación, las arquitecturas RISC tradicionales han tenido poco éxito con ILP.

La especulación (speculation) minimiza el efecto del retardo de la memoria

En el contexto actual, las CPUs van incrementando su velocidad a una tasa mucho mayor que la memoria. Se espera que esta tendencia continúe en el futuro, y posiblemente se acentúe. Entonces estamos ante un gran cuello de botella que aqueja a las arquitecturas de hoy.

Para reducir el impacto de esta limitación, las arquitecturas tradicionales permiten que el compilador y el procesador realicen la carga de datos tiempo antes de que estos sean necesarios, pero los saltos hacen el papel de barreras, de límites para el entorno en el que se aplica esta técnica.

IA-64 emplea una técnica llamada especulación para iniciar la carga desde memoria en forma anticipada, inclusive anticipando al uso en porciones de código más allá de un salto.

La especulación es una característica muy utilizada en IA-64. Las arquitecturas RISC tradicionales pueden utilizar una técnica llamada 'carga a prueba de fallas' (non-faulting load) para evitar el costoso manejo de errores cuando la carga anticipada podría no ser válida.

IA-64 evita este problema ofreciendo una solución en la arquitectura para manejar la carga anticipada. La idea consiste en que cada dato va asociado a un bit que dice si un error se ha producido o no cuando se cargó dicho dato. Este bit de error (llamado “Nat”) acompaña al dato a través de todas las operaciones aritméticas, moves y condicionales hasta que se realiza sobre el dato una instrucción de check, que detecta si hubo un error, y dispara la rutina manejadora del error. Sin embargo, un 99% de las veces una carga anticipada que produce un error resulta pertenecer a una rama que nunca se debió ejecutar, por lo que simplemente se abandona su ejecución, sin saltar a la rutina manejadora de errores.

Esto significa que con IA-64, el compilador puede realizar carga anticipada en forma agresiva (speculation) sin prestar atención a penalidades en caso de errores.

Una forma adicional de especulación se denomina “Data Speculation”. Esto permite al compilador realizar la carga aún antes de un store que podría escribir sobre el mismo registro para el cual se realiza la carga anticipada. IA-64 mantiene un registro de todas las cargas anticipadas en una tabla llamada ALAT (Advanced Load Address Table). Cuando se encuentra una instrucción store que causa conflictos con una carga anticipada contenida en la tabla, la entrada correspondiente se elimina, y cuando la carga se chequea en el ALAT, la ausencia de su entrada indica que el valor debe ser recuperado nuevamente y se obtiene el resultado correcto.

Loop unrolling and rotation (desdoblamiento y rotación de bucles)

El manejo eficiente de bucles es uno de los factores clave para determinar la performance de todas las arquitecturas, y IA-64 provee características especiales para ello.

Una técnica común utilizada es el desdoblamiento. Consiste en evitar los saltos al principio del bucle duplicando el código del bucle dos, tres o más veces para reducir el número de saltos. Esta técnica está disponible en IA-64 y en las arquitecturas RISC.

El problema de esta técnica es que conlleva un aumento del código. IA-64 sin embargo provee características especiales para explotar esta técnica. Cuenta con la rotación de registros que permite la ejecución de bucles sin el aumento de código.

En IA-64, los registros y los predicados pueden rotar bajo el control del compilador. Esto significa que con cada paso en la rotación, los valores que estaban en, digamos, el registro 40, avanzarían al registro 41. El valor en el último registro se mueve al primer registro, o sea, los valores de los registros rotan. Lo mismo sucede con los registros que contienen predicados.

Con esta rotación, el compilador puede correr una copia del código dentro del bucle y los valores se asignan a distintos registros automáticamente a medida que el código itera. Cuando el bucle finaliza, con la ayuda de los registros “loop count” y “epilog count” se vuelve al estado anterior a la rotación y el bucle termina.

De esta manera se administran los bucles sin la necesidad de prólogos o epílogos al principio y al final de cada bucle para manejar la inicialización y la restauración, y sin usar branch-prediction que podría llevar a penalidades en la performance en caso de fallar.

Recursos masivos

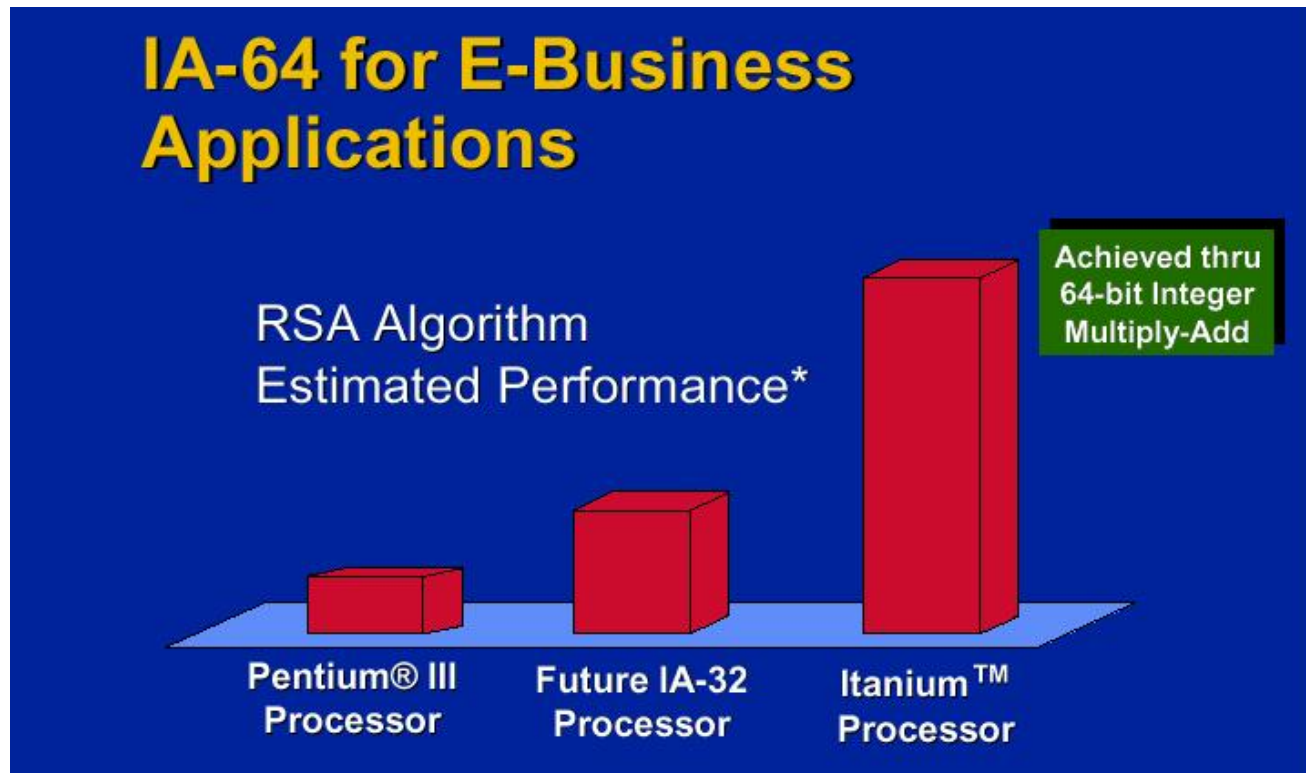
Las arquitecturas tradicionales, aparte de tener muchas restricciones para llevar a cabo ILP, también cuentan con pocos recursos en hardware para permitir la ejecución en paralelo. En contraste, los procesadores basados en IA-64 cuentan con 128 registros enteros de propósito general, 128 registros de punto flotante, 64 registros de predicación y muchas unidades de ejecución para asegurar el buen funcionamiento con altas cargas de trabajo. La arquitectura se diseñó para ser escalable, permitiendo la fácil adición de unidades de ejecución.

EPIC como una nueva forma de concebir el código

La implementación de la filosofía de diseño EPIC permite alcanzar nuevos niveles de paralelismo y rompe el paradigma de ejecución secuencial que existe en las arquitecturas tradicionales. El uso de mecanismos como la predicación y la especulación combinadas con el paralelismo explícito permiten que IA-64 logre ganancias de performance donde otras arquitecturas no han podido hacerlo (retardo de memoria, fallas en la predicción de ramas).

Intel y HP acentúan las bondades de la predicación y la especulación más allá de simples características que mejoran la performance, y las sitúan al nivel de adelantos revolucionarios que permiten diseñar todo un nuevo paradigma para lograr verdaderamente la ejecución paralela más allá de su simple aplicación a casos especiales.

Características del procesador Itanium: [\(Volver\)](#)



Performance estimada del procesador Itanium para encriptación con algoritmo RSA

- Frecuencias de reloj de 733 y 800 MHz
- Caché unificado L3 de 2 ó 4 MB, L2 de 96 KB y L1 de 32 KB
- Dos unidades de enteros y dos de memoria, permitiendo ejecutar 4 instrucciones ALU por pulso de reloj.
- Unidad de punto flotante que contiene 2 unidades FMAC (Floating-point Multiply ACcumulate) que operan con operandos de 82 bits. Cada FMAC puede ejecutar 2 operaciones de coma flotante por pulso de reloj con precisión simple, doble o doble extendida.
- Dos unidades de coma flotante para multimedia adicionales, capaces de ejecutar dos operaciones de coma flotante de precisión simple cada una, lo que combinado con las FMACs significa un total de 8 operaciones de coma flotante por pulso de reloj, resultando en un pico de 6.4 GFLOPS.

- Bus de datos de 266 MHz
- Detección avanzada de errores (MCA - Machine Check Architecture), y ECC (Error Correcting Code) en caché y bus de sistema.
- 25 millones de transistores en CPU, 300 millones en cache
- Tasa de transferencia del bus de datos del sistema de hasta 2.1 GB/seg.

Ideas finales sobre IA-64 [\(Volver\)](#)

La fase de producción de IA-64 ha finalizado, y la cantidad de unidades Itanium en el mercado va aumentando. Se espera que con Itanium Intel ingrese en forma agresiva al mercado de los microprocesadores de 64 bits.

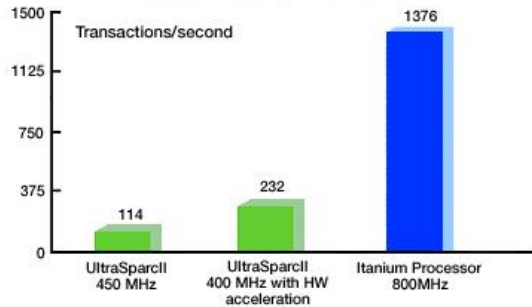
Su arquitectura tipo EPIC incorpora las técnicas de superescalarización más innovadoras del momento, y el liderazgo de Intel y HP en las tecnologías de los procesos de fabricación aseguran frecuencias de reloj mayores modelo tras modelo.

Los detractores de esta arquitectura señalan su excesiva complejidad y la gran cantidad de cambios que se tienen que realizar para lograr aprovechar al máximo sus prestaciones, como por ejemplo tecnologías de compiladores ultra-inteligentes que permitan detectar condiciones de optimización que puedan ser aprovechadas por EPIC. También se critica su aparente mediocridad a la hora de ejecutar código IA-32, lo que conllevará a todo un cambio de software para lograr un aprovechamiento razonable de IA-64.

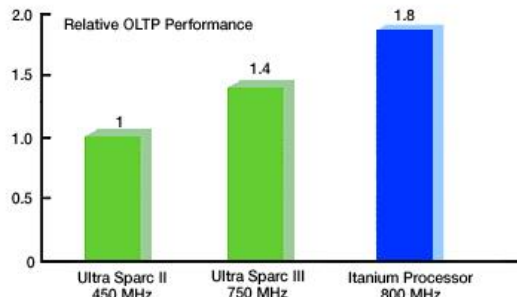
Sin embargo, el impulso que la gigantesca Intel dará a esta arquitectura no puede dejar de tenerse en cuenta, y si se suma a ello el pronto apoyo de Microsoft, el rol de IA-64 en el mercado de las CPUs de 64 bits probablemente será muy importante.

Benchmarks del Itanium [\(Volver\)](#)

Security Performance
Coradiant* Secure Transactions
The Itanium™ processor provides world-class security performance for the most demanding e-Commerce environments

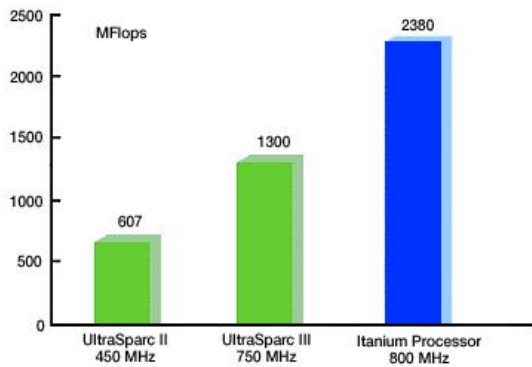


Transaction Processing
Best of class 4-way transaction performance



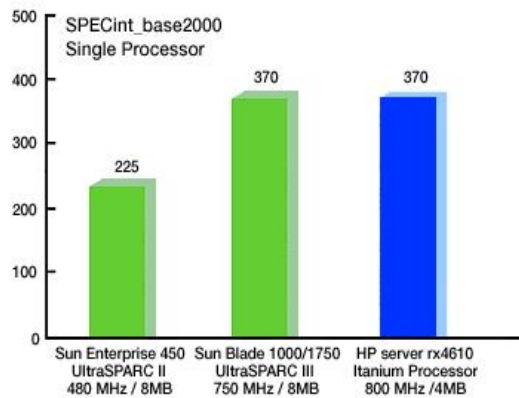
High Performance / Technical Computing
Linpack*-1000 Benchmark

The Itanium™ processor provides world-class floating-point performance in the most rigorous technical computing environments

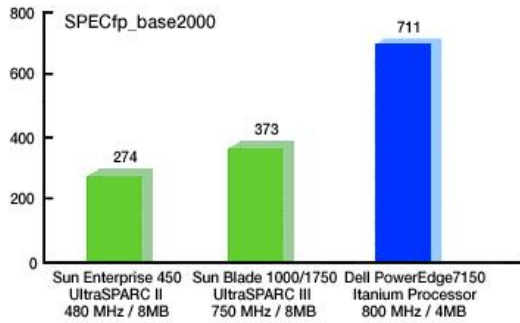


SPEC CPU2000 Performance – SPECint2000

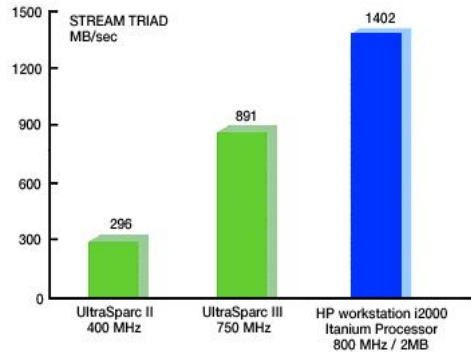
Itanium™ Processor delivers best of class floating point performance and competitive integer performance



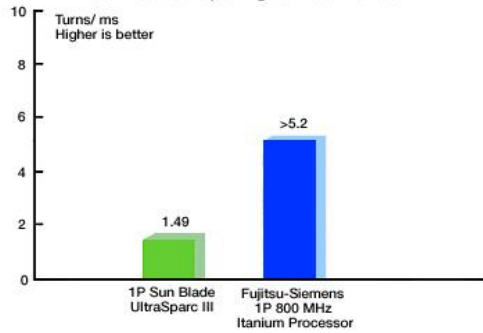
SPEC CPU2000 Performance – SPECfp2000
Itanium™ processor delivers best-of-class floating point performance and competitive integer performance



System Memory Bandwidth Performance
STREAM* Benchmark
The Itanium™ processor with the Intel® 460GX chipset provides leading system memory bandwidth performance



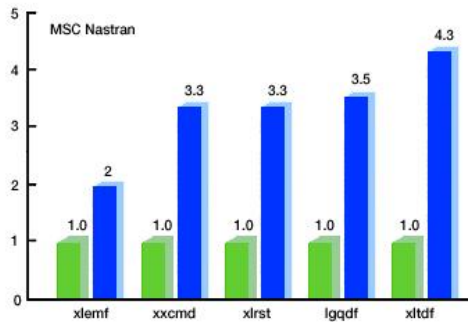
Scientific Computing Performance
Sixtrack* Benchmark @ CERN
The Itanium™ processor provides world-class floating-point performance in the most rigorous technical computing environments



Mechanical Computer-Aided Engineering
MSC.Nastran – Nastran 70.7.2
Itanium™ processor-based systems provide performance for the most demanding MCAE applications

Itanium™ processor-based systems provide performance for the most demanding MCAE applications

■ 1P UltraSparc II 400 MHz
 ■ 1P Itanium-based system 800 MHz



5. x86-64 de AMD [\(Volver\)](#)

Introducción a la arquitectura x86-64 nombre código “Hammer”

La arquitectura x86-64 es una extensión lógica simple del set de instrucciones de la arquitectura x86-32 que se usa actualmente en todas las PCs basadas en Windows. Es por ello que diferencia de la IA64 de Intel la plataforma de 64 bits de AMD es capaz de correr casi sin ninguna penalidad de rendimiento sistemas operativos y aplicaciones de 16 y 32 bits y no exige la creación de compiladores y herramientas radicalmente diferentes a las existentes hasta el momento. Por otro lado esta basada en tecnología con más de 20 años de antigüedad y por esto debe enfrentarse a varios problemas propios de esta.

La versatilidad de los procesadores Hammer proviene de su capacidad de operar en distintos modos, esto será explicado con mayor detalle en el siguiente apartado.

Modos de Operación [\(Volver\)](#)

Table 1. Operating Modes

Mode		Operating System Required	Application Recompile Required	Defaults ¹			
				Address Size (bits)	Operand Size (bits)	Register Extensions ²	GPR Width (bits)
Long Mode ³	64-Bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	
		16					
Legacy Mode ⁴		Legacy 32-bit or 16-bit OS	no	32	32	no	32
				16	16		

1. Defaults can be overridden in most modes using an instruction prefix or system control bit.
 2. Register extensions includes eight new GPRs and eight new XMM registers (also called SSE registers).
 3. Long mode supports only x86 protected mode. It does not support x86 real mode or virtual-8086 mode. Also, it does not support task switching.
 4. Legacy mode supports x86 real mode, virtual-8086 mode, and protected mode.

- Legacy Mode: este modo es completamente compatible con las implementaciones de 32 bits existentes de la arquitectura x86, lo que incluye tanto sistemas operativos como aplicaciones de 32 bits. Se hicieron cambios a las estructuras de paginación del x86 de 32 bits para que puedan usar direcciones físicas más grandes. Específicamente un software ejecutando en legacy mode puede tomar ventaja de las extensiones a la dirección física PAE (Physical Address Extension) y extensión de tamaño de página PSE (Page Size Extension).
- Long Mode: este modo opera en 64 bits.

Una de las características principales de este modo es que implementa 64-bit flat virtual addressing. Los procesadores actuales de 32-bits se encuentran limitados a solo unos pocos gigabytes de memoria física, lo cual todavía es suficiente para varios equipos desktop. El X86-64 long mode soporta hasta 64 bits de espacio de direccionamiento virtual (2^{64} bytes o 18.000.000.000.000.000 bytes o bien 18 exabytes) con un máximo de 2^{52} bytes (4.500.000.000.000.000 bytes o 4,5 petabytes o bien 4.500 terabytes) de

espacio de direccionamiento físico. Esto constituye mucha memoria más que cualquier computadora o super computadora existente. Sin embargo las primeras implementaciones de la familia Hammer solo tendrán 48 bits de espacio de direccionamiento virtual (280.000.000.000.000 o 280 terabytes) con 40 bits de espacio de direccionamiento físico (más o menos 1 terabyte). Lo que todavía se encuentra más allá de las computadoras de 32 bits actuales.

El Long Mode está compuesto por 2 sub modos entre los cuales puede moverse a nivel de segmentos de código y que ofrecen distintas funcionalidades:

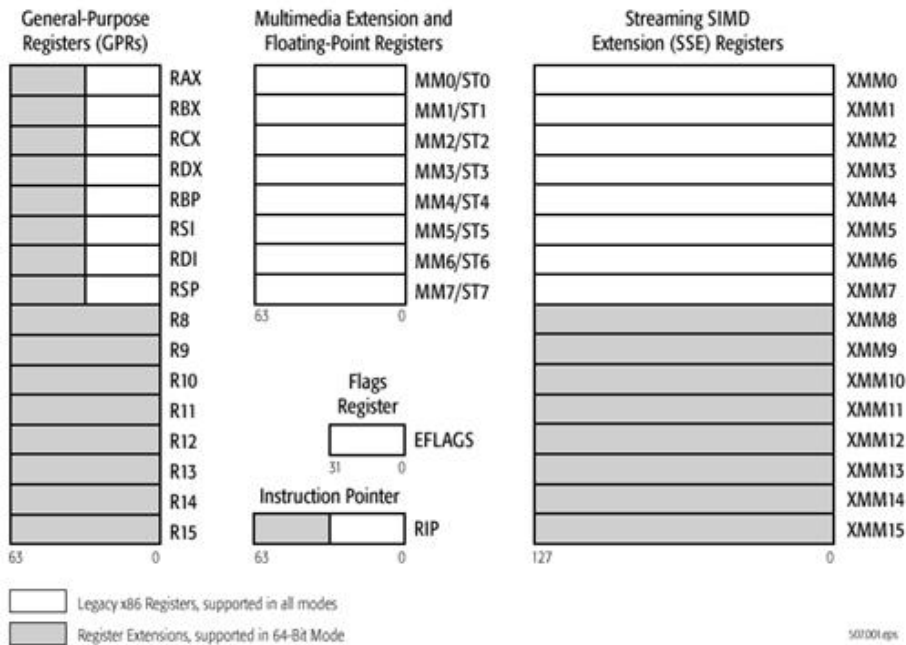
- **Compatibility Mode:** este modo permite que los sistemas operativos de 64 bits implementen una compatibilidad binaria con las aplicaciones de 16 bits y 32 bits x86, permitiendo que estas aplicaciones se ejecuten sin que haga falta recompilarlas. En el compatibility mode las aplicaciones sólo pueden acceder a los primeros 4GBytes del espacio de memoria virtual. El modo de compatibilidad puede ser activado por el sistema operativo a nivel de un segmento de código. Mediante esto se logra que desde el punto de vista de la aplicación el compatibility mode sea parecido a un ambiente de modo protegido en un x86 de 32 bits y desde el punto de vista del sistema operativo de 64 bits la traducción de direcciones, interrupciones, manejo de excepciones, y estructuras de datos del sistema usen los 64 bits del long mode.

Al igual que en el legacy mode se pueden usar direcciones físicas de memoria más grandes, pero además de esto existe un mecanismo de traducción de página de cuatro niveles que se encuentra disponible para el long mode y que es usado por el compatibility mode.

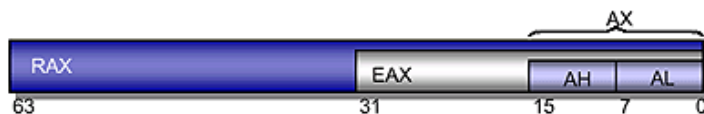
Finalmente aunque en este modo no se utilicen las extensiones de 64 bits las aplicaciones corridas en este modo se beneficiarán de los avances de velocidad del procesador tanto como las aplicaciones de 64 bits.

- **64-Bit Mode:** en este modo se ejecutan las aplicaciones de 64 bits y es en este modo que se implementan y se usan las extensiones de la arquitectura X86-64 de las cuales citaremos ahora las más importantes:
- **8 nuevos registros de propósito general o GPRs (General-Purpose Registers):** los registros son lugares de memoria extremadamente rápidos usados dentro de la CPU para cálculos y otras operaciones. X86-64 dobla el número actual de GPRs introduciendo 8 nuevos GPRs de 64 bit, estos GPR extra tienen la capacidad de acelerar grandemente las aplicaciones que son computacionalmente intensas.
- **8 nuevos registros para SSE (Streaming SIMD Extensions):** esto brinda soporte para las extensiones SIMD (Single Instruction Multiple Data). Esto es de importancia ya que varios benchmarks indican que el Athlon ofrece menor rendimiento en varias aplicaciones debido a la superioridad del SSE sobre el análogo pero más viejo 3D Now! de AMD. Debido a esto el x86-64 tiene un total de 16 registros de punto flotante de 128 bits.

El siguiente gráfico muestra la configuración de registros final del x86-64 coloreando de distintos colores los registros disponibles en el legacy mode (y para todos los otros modos) y los registros disponibles sólo para el 64 bit mode.



- Extensión de los GPRs actuales: el x86-64 soporta enteros de 64 bits, lo que significa que números tremendamente grandes pueden ser manejados precisa y rápidamente, entonces además de los 8 nuevos registros de propósito general el x86-64 extiende los registros existentes en arquitecturas x86 de 32 bits a 64 bits, de manera similar a como el i386 extendió en su tiempo los registros de 16 bits a 32 bits. La extensión es lograda mediante el uso del prefijo REX, por ejemplo el registro RAX es la versión de 64 bits del registro EAX, lo que permitiría a una función ADD (con algún prefijo identificador) sumar números de 64 bits.



- Acceso relativo a datos o RIP (Relative Data Addressing): en este modo también se soporta direccionamiento relativo al apuntador de instrucción de 64 bits. Esto mejora la eficiencia del código independiente de la posición y del código que direcciona a datos globales.

Otras Características

- Branches (saltos): la arquitectura del long mode posee dos mecanismos de branching para acomodar saltos en un espacio virtual completo de 64 bits:
- Semántica de las instrucciones de salto cercano (Near-branch) fueron redefinidas en el modo de 64 bits: en este modo el tamaño de operando para todos los saltos cercanos (CALL, RET, JCC, JCXZ, JMP, and LOOP) es forzado a 64 bits. Por lo que estas instrucciones actualizan por completo el RIP de 64 bits sin la necesidad del prefijo REX de tamaño del operando.
- Tanto en el modo 64 bits como en el compatibility mode se define un descriptor call-gate para llamadas lejanas: los mecanismos call gate proveen puntos de entrada públicos al sistema operativo, aunque en arquitecturas legacy x86 se usan mecanismos call-gate para permitir que el software salte de un nivel de

privilegio a otro, también pueden ser usados para saltos que no involucren un cambio de privilegio. Por esto el descriptor call-gate del long mode permite que los saltos lejanos referencien a cualquier lugar del espacio en la memoria virtual soportada, los call gates del long mode también contienen el selector de código destino (CS) permitiendo cambios de privilegio y de tamaños por defecto.

- Extensión por ceros de direcciones de 16 bits y 32 bits: todas los cálculos de direcciones de 16 bits y 32 bits son extendidos con ceros en el long mode para así formar direcciones de 64 bits. Entonces a la hora de realizar los cálculos de direcciones se truncan primero las direcciones a su tamaño efectivo según el prefijo que indica el tamaño de la dirección. Es por esto que las aplicaciones 16 bits y 32 bits ejecutadas en el compatibility mode pueden acceder sólo a los 4GBytes inferiores del espacio de direcciones virtuales del long mode.
- Page Map Level 4 (PML4): la arquitectura del modo long agrega una tabla nueva llamada PML4, la cual se encuentra por encima del Page Directory Pointer (PDP) en la jerarquía de traducción de página. La PML4 contiene 512 entradas de 8 bytes y cada una de estas apunta a una tabla PDP. Esto se usa para extender los esquemas tradicionales de paginación a 64 bits.
- Stack Pointer: mientras que en el modo 64 bits el tamaño del stack pointer es siempre 64 bits, en los otros modos se usa un descriptor SS (Stack Size) para determinarlo.

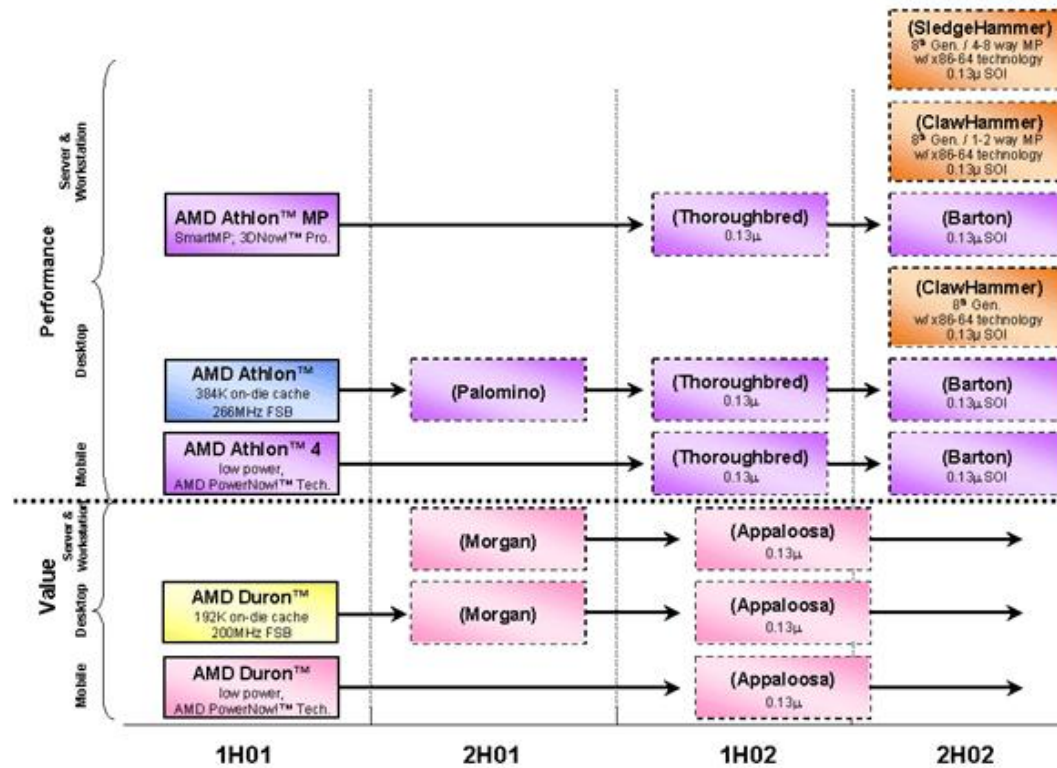
Progreso del X86-64 en el tiempo [\(Volver\)](#)

El 16 de Enero 2001 AMD y Virtutech anuncian que están trabajando en una herramienta de alto desempeño llamada "VirtuHammer" que permitirá que los desarrolladores escribir y probar programas de 64 bits en computadoras existentes.

- El 31 de Enero del 2001 se realiza la primera demostración pública del VirtuHammer en la feria LinuxWorld en Nueva York, este se muestra corriendo en una Athlon de 1.1.
- El 10 de Agosto del 2001 AMD empieza a distribuir públicamente "The AMD x86-64TM Architecture Programmers Overview", con esto AMD espera motivar a los programadores para que empiecen el desarrollo de software para su línea de procesadores de 64 bits. La información en este documento es esencial para que los arquitectos de compiladores y sistemas operativos puedan tomar ventaja de las características ofrecidas por la nueva arquitectura 64 bits.
- Sólo 5 días después de la publicación de las características del x86-64, una alianza de desarrolladores y distribuidores Linux anunciaron su apoyo la línea de procesadores Hammer de AMD. La lista incluye a Red Hat, SuSE, CodeSourcery y Ada Core Technologies.

AMD Roadmap [\(Volver\)](#)

Se incluye a continuación los planes futuros (roadmap) de AMD con respecto a sus microprocesadores. Tanto el ClawHammer como el SledgeHammer pertenecen a la familia Hammer que usa la arquitectura x86-64, la diferencia entre estos es que el ClawHammer constituye una versión disminuida (y por ende más económica) del SledgeHammer.



Problemas a enfrentar [\(Volver\)](#)

Hasta ahora el único asunto que todavía no ha sido manejado por AMD es la falta de chipsets para la familia Hammer. VIA, la mayor proveedora de chipsets está orientada actualmente a la producción de chipsets de bajo precio, por lo que se espera que no se interese en los procesadores de la familia Hammer que son de alto rendimiento y las otras dos compañías en la industria de los chipsets que apoyan a AMD, Ali y SiS, no tienen el tamaño de VIA en el mercado lo que les haría difícil proporcionar chipsets para la familia Hammer. Esto deja a AMD sola para enfrentar la tarea de crear el chipset como en el pasado hizo con el chipset AMD760 para sus procesadores Athlon y Duron.

Ideas finales sobre x86-64 [\(Volver\)](#)

Son realmente pocos los que ven problemas con la implementación como con el suficiente apoyo de compiladores para el x86-64 de AMD. Esto se debe a que su diseño se encuentra basado en principios bien entendidos y difundidos lo que permite la ejecución de legacy software sin penalidades en la performance. El Itanium de Intel, su principal competidor, por otro lado está optimizado para VLIW (Very Long Instruction Word) de 64 bits lo cual implementa tecnologías radicalmente nuevas, requiere la creación de complicados compiladores y hace que la performance del sistema en aplicaciones de 32 bits sea menor a la de las Pentiums IV, Athlons, Pentiums III y hasta posiblemente CPUs del tipo Duron.

Por otro lado existe una desventaja a una arquitectura de 64 bits original. AMD se basa en una arquitectura con más de 2 décadas de antigüedad y el hecho de conservar esta arquitectura implica traer todos los problemas

viejos de esta al nuevo diseño. AMD piensa por su lado que la performance de un procesador tiene más que ver con la forma en que el set de instrucciones se encuentra implementada, más que con el set de instrucciones mismo.

6. Conclusiones [\(Volver\)](#)

Probablemente el mayor motivo del suceso de las plataformas de 32 bits se debe a que la memoria máxima que pueden direccionar sus procesadores (4Gbytes) es hasta el momento suficiente para la mayoría de las aplicaciones que interesan al usuario común de computadoras, aunque otras causas secundarias que podrían citarse serían:

- Intel, el mayor productor de procesadores en el mundo recién ahora empieza a producir microprocesadores de 64 bits.
- La descentralización de las tareas, con la introducción de las nuevas placas de video, sonido, etc., permiten al procesador realizar cada vez menos tareas de las que antes se le requerían.

Aún con la introducción de Intel en mercado y la creciente cantidad de aplicaciones que requerirían los beneficios de los procesadores de 64-bits, según analistas pasarán probablemente otros 10 años más antes que esta arquitectura se difunda entre el público en general, de misma manera que se encuentran difundidos los procesadores de 32 bits actualmente.

Sin embargo, el mundo de los procesadores de 64 bits está atravesando por grandes cambios. Los chips Alpha, considerados como la arquitectura de más poder de cómputo para el mercado de consumo, de DEC en un principio, y más tarde de Compaq, dejarán de producirse al ser la misma Compaq adquirida por HP, que codesarrolla con Intel la arquitectura IA-64. Es de esperarse que gran cantidad de conocimientos tecnológicos que se aplicaban a las Alpha sean utilizados para mejorar el diseño de las Itanium de Intel-HP.

MIPS, otro desarrollador de arquitecturas de microprocesadores, ha dejado el negocio de los servidores de alta performance (los servers y workstations de Silicon Graphics) y se dedica al desarrollo de sistemas incrustados (embedded).

Esto deja como competidores establecidos al consorcio SPARC, liderado por Sun Microsystems, y a las PowerPC de IBM, y como nuevos competidores a Intel y HP con IA-64 y a AMD con su x86-64.

7. Bibliografía [\(Volver\)](#)

- INTEL, “Intel Itanium Processor - Hardware Developer's Manual”, Agosto de 2001, Intel Corp. (PDF).
- INTEL, “EPIC - Rising Over RISC”, 2001, Intel Corp. (PDF).
- HUCK, J., ZAHIR, R. “The IA-64 System Architecture: Tutorial for Hardware, OS, & Application Developers”, Febrero de 2001, HP & Intel Corp. (PDF).
- DOSHI, Gautam. “Understanding the IA-64 Architecture”, Setiembre de 1999, Intel Corp. (PDF).
- <http://www.intel.com/products/eBusiness/ia64/faqs/>, “Intel Corp. Website - IA64 FAQ”
- <http://www.tomshardware.com/>, “Tom's Hardware Review”
- <http://www.chips.ibm.com/products/powerpc/>, “IBM Microelectronics Website - Products - PowerPC”
- <http://www.spec.org/>, “SPEC Forum”
- SPARC, “The SPARC Architecture Manual - Version 9”, 2000, Prentice-Hall (PDF).
- <http://www.sun.com/microelectronics/UltraSPARC-III/>, “Sun Microsystems - UltraSPARC III Processors”

Apéndice: Benchmarks SPEC comparativos de varias arquitecturas

[\(Volver\)](#)

