

ARQUITECTURA
SPARC:
CONCEPTOS
GENERALES

De Diego Varona, Rubén
Romay López, Oscar Manuel
Vega Martínez, Jorge

INTRODUCCIÓN

SPARC (del inglés Scalable Processor Architecture). Un procesador con arquitectura escalable es aquel en el que se pueden definir versiones posteriores de la misma con mayor cantidad de características, siempre salvaguardando la compatibilidad con los programas de versiones anteriores. Tiene una arquitectura basada en un juego de instrucciones RISC (Reduced instruction set computer) desarrollado por la empresa Sun Microsystems y lanzado a mediados de 1987.

Fue originalmente diseñada por Sun Microsystems y dirigido por el ingeniero Kaa en 1985, se basa en los diseños RISC I y II de la Universidad de California en Berkeley que fueron definidos entre los años 1980 y 1982.

La empresa Sun Microsystems diseñó esta arquitectura y la licenció a otros fabricantes como Texas Instruments, Cypress Semiconductor, Fujitsu, LSI Logic entre otros.

SPARC es la primera arquitectura RISC abierta y como tal, las especificaciones de diseño están publicadas, así otros fabricantes de microprocesadores pueden desarrollar su propio diseño.

Una de las ideas innovadoras de esta arquitectura es la ventana de registros que permite hacer fácilmente compiladores de alto rendimiento y una significativa reducción de memoria en las instrucciones load/store en relación con otras arquitecturas RISC. Las ventajas se aprecian sobre todo en programas grandes.

La cpu SPARC está compuesta de una unidad entera, UI (*Integer Unit*) que procesa la ejecución básica y una FPU (*Floating-Point Unit*) que ejecuta las operaciones y cálculos de reales. La IU y la FPU pueden o no estar integradas en el mismo chip.

Aunque no es una parte formal de la arquitectura, las computadoras basadas en sistemas SPARC de Sun Microsystems tienen una unidad de manejo de memoria (MMU) y un gran caché de direcciones virtuales (para instrucciones y datos) que están dispuestos periféricamente sobre un bus de datos y direcciones de 32 bits.

HISTORIA

Sparc es una marca registrada de SPARC International Inc, una organización establecida en 1989 para promover la arquitectura Sparc , y proporcionar tests de estabilidad. La implementación de la arquitectura original de 32 bits de Sparc fue inicialmente diseñada en estaciones de trabajo Sun-4 (son una serie de estaciones de trabajo y servidores Unix, lanzados en 1987), reemplazando los sistemas anteriores (Sun-3) basados en la familia de procesadores Motorola 68000. Posteriormente, los procesadores Sparc fueron utilizados en servidores SMP (Symmetric multiprocessing donde dos procesadores de características idénticas están conectados a una única memoria central compartida) desarrollados entre otros por Sun y Fujitsu, y diseñados para operaciones de 64 bits.

Ha habido tres grandes versiones de la arquitectura Sparc. La primera de ellas (publicada en 1986) fue la versión V7 (32 bits). Sparc v8 (V8) (una versión superior) fue lanzada en 1990. La principal diferencia entre estas dos versiones fue la introducción de operaciones de producto y cociente de números enteros. La versión V8 deSPARC ha servido como base para Estándar IEEE 1754-1994, un estándar IEEE para una arquitectura de microprocesadores de 32 bits.

La versión V9 de Sparc fue lanzada en 1993, esta ya poseía una arquitectura de 64 bits.

En el año 2002, fujitsu y Sun liberaron JpS1 (*Joint Programming Specification 1*) . Describiendo las funciones de procesador que de modo idéntico fueron puestas en práctica en variasCPU´s de ambas empresas. Las primeras CPU´s en las que se implementa JPS1 fueron el UltraSPARC III por Sun y el SPARC64 por Fujitsu

A principios del año 2006, Sun liberó UltraSPARC (Arquitectura 2005) Esta incluía no sólo las características de SPARC V9, sino también todas las extensiones arquitectónicas (como CMT, hiperprivileged, VIS 1, y VIS 2) el presente en los procesadores UltraSPARC de Sun los cuales se ponen en práctica con UltraSPARC T1. UltraSPARC (Arquitectura del 2005) incluye las extensiones estándar de Sun..

En el año 2007, Sun publica una nueva actualización; la arquitectura UltraSPARC 2007.

Entre tantas implementaciones de Sparc, las mas populares fueron la SuperSPARC y la UltraSPARC-I, estas fueron usados como sistemas de referencia para los benchmarks. El UltraSPARC 296 Mhz es el sistema de referencia para el Benchmark de SPEC CPU2006.

PRINCIPALES CARACTERÍSTICAS

- Su característica distintiva es utilizar ventanas de registros.(De la cual hablaremos a continuación)
- 32 registros de "enteros" de 32 bits.
- 16 registros de punto flotante de 64 bits (para el caso de doble precisión) que se pueden utilizar como 32 registros de 32 bits (para precisión simple).
- Modos de direccionamiento:
 - Inmediato, (constantes de 13 bits).
 - Directo, (offset de 13 bits).
 - Indirecto, (registro + offset de 13 bits o registro + registro).
- Utiliza instrucciones retardadas (saltos, load y store).
- Manejo de memoria:
 - Espacio virtual de 4 Gigabytes.
 - Unidad de manejo de memoria (MMU) que trabaja con páginas de tamaño configurable.

VENTANAS DE REGISTROS

Los procesadores SPARC realizan un “sistema de ventanas”. Cada ventana son 24 registros (a los cuales se le suman 8 más que son globales, o sea, accesibles por todas las ventanas; la suma de 32 registros es visible en todo momento) que pertenecen a un procedimiento o subprograma. A medida que se van creando nuevas ventanas los registros de salida de la

ventana actual se solapan sobre los de entrada de la ventana siguiente. Los CPU registros locales y de salida de la ventana actual son diferentes a los registros locales y de salida de la ventana siguiente. El siguiente gráfico nos lo muestra esquemáticamente:

En la figura, CWP es el índice de la ventana actual. El comando SAVE y las TRAPS decrementan dicho índice, y RESTORE y RETT lo incrementan.

Para verificar si un número de ventana es válido se utiliza el registro WIN.

Invalid Mask) de 32 bits. Si el bit i-ésimo está en 1 indica que esa ventana inválida y si se intenta acceder esta, se genera un error (TRAP “window_overflow”).

El registro G0 está fijo por medio del hardware con el valor 0. En él, los almacenamientos no alteran su valor. Los registros G1 a G7 son globales y pueden contener variables enteras, apuntadores a tablas u otros datos importantes. Los registros de salida son de O0 a O7 y son usados por los procedimientos para pasar parámetros a los procedimientos que llaman. El primer parámetro va al registro O0, el siguiente al O1 y así sucesivamente. El registro O6 (AE) se utiliza como apuntados a la pila de memoria. La pila se usa para los parámetros que exceden los registros, espacio de asignación dinámica de la pila, ventanas apiladas en memoria, así como para salvar registros de punto flotante, apuntadores a buffers en donde los procedimientos llamados pueden enviar estructuras y arreglos de datos, etc. La instrucción CALL (llamada) deposita la dirección de retorno en el registro O7. Los 8 registros para variables locales L0 a L7, se pueden usar en la forma que el programador o el compilador considere más conveniente. Las 8 variables de entrada I0 a I7 son los parámetros que son pasados al procedimiento actual por el que le llama: Los registros no utilizados pueden contener variables locales adicionales. I6 (FP) es el apuntador de marco que se usa para direccionar variables en el marco de pila. A diferencia de SP que solo puede cambiar mientras el procedimiento se ejecuta, FP apunta la misma palabra en memoria durante toda la ejecución del procedimiento, siendo más apropiado para la indexación que el SP. La dirección de retorno del procedimiento se encuentra contenida en el registro I7.

Por lo tanto y en general, existen 4 grupos de registros diferentes:

Registros Globales : G0-G1

Registros de Salida: O0-O7

Registros Locales: L0-L7

Registros de Entrada: I0-I7

CATEGORÍA DE INSTRUCCIONES

La arquitectura SPARC tiene cerca de 50 instrucciones enteras. Estas instrucciones se pueden clasificar en cinco grandes categorías.

- LOAD y STORE (La única manera de acceder a la memoria). Estas instrucciones usan dos registros o un registro y una constante para calcular la dirección de memoria a direccionar.
- Instrucciones Aritméticas/Lógicas/Shift. Ejecutan operaciones aritméticas, lógicas y de desplazamiento de bits. Estas instrucciones calculan el resultado si es una función de 2 operandos y guardan el resultado en un registro.
- Operaciones del Coprocesador. La IU (unidad de instrucciones) extrae las operaciones de punto flotante desde las instrucciones del bus de datos y los coloca en la cola para la FPU (Unidad de punto flotante). La FPU ejecuta los cálculos de punto flotante con un número fijo en unidad aritmética de punto flotante, (el número es dependiente de la aplicación). Las operaciones de punto flotante son ejecutadas concurrentemente con las instrucciones de la IU y con otras operaciones de punto flotante cuando es necesario. La arquitectura SPARC también especifica una interfaz para la conexión de un coprocesador adicional.
- Instrucciones de Control de Transferencia. Estas incluyen saltos, llamadas, etc... El control de transferencia tiene retardo hasta después de la ejecución de la próxima instrucción, así el pipeline no es vaciado porque ocurre un control de tiempo. De este modo, los compiladores pueden ser optimizados por ramas retardadas.
- Instrucciones de control de registros Lectura/Escritura. Estas instrucciones se incluyen para leer y grabar el contenido de varios

registros de control. Generalmente la fuente o destino está implícito en la instrucción.

En las siguientes dispositivos podemos ver cuales son algunas de estas instrucciones:

DIRECCIONAMIENTO DE MEMORIA

A diferencia de los procesadores 80x86 el espacio de direcciones de los procesadores SPARC es lineal y no segmentado. Por otra parte el bus de direcciones es de 32 bits lo que permite direccionar hasta 2^{32} bytes = 4 Gbytes de memoria virtual. Es decir que para especificar una dirección de memoria son necesarios 32 bits.

Como ya hemos hablado antes otra característica importante de los procesadores SPARC es que son procesadores RISC. Para que un procesador sea considerado RISC debe cumplir cierto conjunto de propiedades entre las cuales hay una que nos interesa resaltar:

la codificación de todas las instrucciones ocupan exactamente 32 bits.

De lo anterior queda claro que no es posible, en los procesadores SPARC, codificar una dirección de memoria absoluta de 32 bits junto con la instrucción que la utilizará ya que no queda espacio para colocar el valor de la dirección mas los bits necesarios para codificar la instrucción.

Para realizar esto el procesador SPARC provee una instrucción especial llamada **SETHI** que permite colocar un valor de hasta 22 bits en los 22 bits más significativos de algún registro. De esta forma luego utilizando adecuadamente la instrucción **OR** es posible colocar los 10 bits menos significativos de la constante en los 10 bits menos significativos del registro.

Veamos como seria el código para realizar "r4=FFFF0000h" :

La forma de direccionar memoria utilizada por los procesadores SPARC es a través de registros y tiene la siguiente sintaxis:

Aquí *desp13* es un número de 13 bits con signo codificado en complemento a 2, es decir que pertenece al rango [-4096,4095] y registro es cualquiera de los 32 registros del procesador.

La semántica de la expresión (1), es decir, a que dirección representa esta dada por:

$(registro) + desp13$

y la semántica de la expresión (2) es

$(registro1) + (registro2)$

(registro) es el valor almacenado en el registro

Ejemplos:

#57(%4) ; contenido de %4 + 57

#-10(%15) ; contenido de %15 - 10

(%4)(%15) ; contenido de %4 + contenido de %15

ALGUNOS EJEMPLOS

- *Sun SPARCstation IPX*
Sun SPARCv7, 40 MHz, Weitek WTL8701
- Ultra Enterprise 3000
Superscalar SPARC Versión 9, UltraSPARC
- Servidores SPARC Enterprise M
SPARC64 VII