

picoJava™

Introducción

El principal objetivo de Sun al introducir Java era poder intercambiar programas ejecutables Java entre computadoras de Internet y ejecutarlos sin modificación. Para poder transportar programas binarios entre máquinas distintas, Sun definió una arquitectura de máquina virtual llamada Java Virtual Machine. Esta máquina tiene una memoria formada por palabras de 32 bits y puede ejecutar 226 instrucciones.

El problema que acarrea este problema es que la interpretación de programas para JVM es lenta, por lo que SUN y otras compañías diseñaron chips de JVM.

Aunque picoJava II no es un chip concreto, es la base de varios chips como la CPU microJava 701 de Sun y muchos otros. Por simplicidad nos referiremos al picoJava II como un chip concreto.

Qué es?

El picoJava es un procesador diseñado para ejecutar nativamente Java *bytecodes* (instrucciones de Java virtual machine).

PicoJava II es una mejora de picoJava que tiene un pipeline RISC de seis etapas con instrucción *folding previo* (un proceso de cargar y ejecutar una instrucción en un solo ciclo). Esto resulta en un factor precio / prestaciones competitivo cuando se compara con procesadores RISC de altas prestaciones operando a la misma frecuencia de reloj.

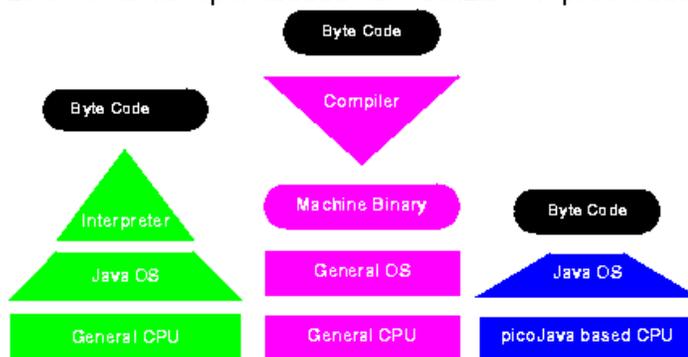
Características

Hasta ahora, un bytecode necesitaba un interprete o era dinámicamente compilado usando un compilador *just in time* (JIT) para cada procesador. Ahora el procesador picoJava elimina la necesidad de los interpretes comúnmente usados, los compiladores JIT y su sobrecarga correspondiente. Esto crea la oportunidad de utilizar el procesador picoJava en aplicaciones cuyos requerimientos de memoria y de energía son limitados.

El picoJava acelera el ambiente de ejecución del Java virtual machine, además se encarga de sincronización de hebras y una variedad de métodos de *garbage collection*. También implementa la invocación de métodos y el ocultamiento de

carga desde variables locales, entonces facilita la programación orientada a objetos. El picoJava puede ser optimizado para energía, espacio(die size), o velocidad. Esta flexibilidad permite una relación precio / prestaciones necesaria para el rango de aplicaciones para las cuales fue diseñado.

El procesador picoJava ejecuta en *hardware* las instrucciones más comúnmente usadas. Las instrucciones complejas con hechas en micro código, y las más complejas son atrapadas y emuladas en *software*. Basado en un *pipeline* de seis etapas con un *stack*



(pila) en caché de 64 entradas, las operaciones realizadas en el tope de la pila son aceleradas utilizando *instruction folding*. Además de los *bytecodes* estándar del Java virtual machine, el picoJava implementa un conjunto extendido de *bytecodes*, los cuales pueden ser ejecutados solamente por el sistema operativo y no están disponibles para el usuario. Estos *bytecodes* extendidos soportan arbitrariamente la carga / almacenamiento, el manejo de caché, y el acceso a registros internos. La interface de entrada / salida y la memoria es manejada por la **Unidad de Interfaz de Bus** (*bus interface unit*). Los datos de punto flotante son permitidos con una precisión simple y doble por la **Unidad de Punto Flotante** (Floating Point Unit) si es que la aplicación lo requiere.

Ventajas para *software*

- Optimizado para Java Virtual Machine
- Para compiladores de C/C++, herramientas de debugger y linckadores en Solaris , Windows NT , and Windows 95
- Provee una **Interfaz para Programas de Aplicación** *Application Programming Interface* (API) para:
 - Ambiente de aplicaciones personales de Java
 - Java TV
 - Java Phone
 - Auto Java

Aplicaciones

El picoJava está principalmente enfocado en el consumidor y para productos que se beneficien con sus prestaciones tales como:

- Digital set-top Boxes
- TV por Internet
- Teléfonos con pantallas que permiten ver quien es el interlocutor
- Dispositivos para comunicación con los automóviles (GIS)

Aspectos técnicos

Es una CPU de un solo chip con dos interfaces de bus, una para el bus de memoria que tiene una anchura de 64 bits y una para el bus PCI que tiene una anchura de 32 bits,. El picoJava tiene una caché de nivel 1 opcional dividida en el chip, con hasta 16 KB para instrucciones y hasta 16 KB para datos. No tiene caché de nivel 2 porque uno de sus aspectos de diseño fue el bajo costo para ser usado en sistemas embutidos. El chip microJava 701 utiliza el bus PCI (a 33 MHz o a 66 MHz). La ventaja de usar este bus es que es estándar y este es un punto positivo en el caso de televisores de Web y otros aparatos más grandes.

Un sistema microJava 701 normalmente incluye una PROM flash. Esto facilita el caso de los aparatos en los cuales una gran parte del programa, sino todo, tiene que estar incluido en el dispositivo.

El microJava 701 tiene 16 líneas de E/S programable que pueden conectarse a botones, interruptores y lámparas del aparato. Por ejemplo los botones de un teléfono celular o un horno microondas.

El chip también tiene tres temporizadores programables incluidos, lo cual también es útil porque los aparatos a menudo operan en tiempo real.

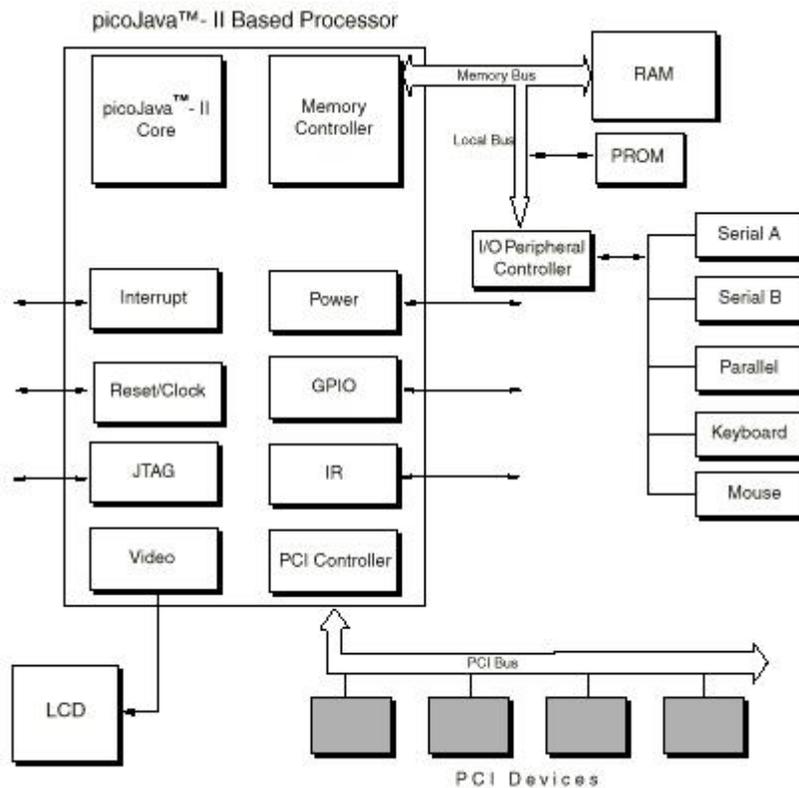


Figure 2. Example of a picoJava™-II Processor Based System

Comparación del Pentium, UltraSPARC y picoJava

A pesar de ser procesadores muy diferentes estos tres procesadores tienen mucho en común.

Las tres máquinas tienen unidades de ejecución similares. Todas las unidades de ejecución aceptan micro operaciones que contienen un código de operación, dos registros fuente y un registro destino. Todas ellas pueden ejecutar una micro operación en un ciclo; todas ellas tienen filas de pre-procesamiento profundas y predicción de ramificaciones y todas ellas tienen cachés I y D divididos, con 16 KB por caché.

La diferencia primordial entre el Pentium II, el UltraSPARC II, Y EL picoJava II es la forma como llegan de su conjunto de instrucciones de ISA a la de ejecución. El Pentium II tiene que descomponer sus instrucciones CISC para ponerlas en el formato de tres registros que la unidad de ejecución necesita. El picoJava II tiene el problema opuesto: cómo combinar varias instrucciones para obtener microoperaciones simples y claras. El UltraSPARC II no tiene nada que hacer porque sus instrucciones nativas ya son microoperaciones.

PicoJava Virtual Machine

El pico Virtual Machine es un java (*software*) adaptado para las necesidades de pequeños sistemas basados en micro controladores de 8/16/32 bit, tales sistemas podrían ser encontrados en beepers, teléfonos celulares, juguetes, y otros.

Cuando se utiliza con estos dispositivos, el pVM brinda todo el poder y la conveniencia de Java.

Beneficios

- ⑩ Campos configurables y dispositivos programables.
- ⑩ Simplifican el desarrollo de la aplicación
- ⑩ Es un camino rápido para expandir la funcionalidad del producto.