

Asignatura de Organización de Computadoras

ARQUITECTURA

MIPS

PRIMER TRABAJO



Alberto Hernández Cerezo

Cristian Tejedor García

Rodrigo Alonso Iglesias

Universidad de Valladolid

Segundo de Ingeniería Técnica Informática de Sistemas

15 de octubre de 2010



ETSII

1.-Introducción a la arquitectura MIPS

Las siglas MIPS (*Microprocessor without Interlocked Pipelines Stages*, o lo que es lo mismo, microprocesador sin enclavamiento de estados de tuberías) hacen referencia a la gama de microprocesadores desarrollados por MIPS Technologies, de arquitectura RISC y registros tipo propósito general de clasificación registro-registro, en los que la mayoría de las instrucciones no acceden a memoria (salvo las instrucciones de carga/descarga) y las instrucciones de los procesadores presentan dos operandos, el fuente y el resultado.

Los diseños de arquitecturas MIPS son utilizados en la línea de productos SGI (siglas de la compañía Silicon Graphics International), numerosos sistemas embebidos, dispositivos para Windows CE (sistema operativo de Microsoft incrustado modular de tiempo real para dispositivos móviles de 32-bits inteligentes y conectados. Nacido en 1994 y combina la compatibilidad y los ping a servicios de aplicación avanzados de Windows con soporte para múltiples arquitecturas de CPU y opciones incluidas de comunicación y redes para proporcionar una fundación abierta para crear una variedad de productos), routers CISCO (*Cisco Systems* es una empresa multinacional con sede en San Jose (California, Estados Unidos), principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones), y videoconsolas (como la N64 o la PSP entre otras).

En un primer momento bastantes fabricantes basaron el diseño de sus Workstation en procesadores MIPS (como *ACER*, *NEC*, *Siemens-Nixdorf*, etc), sin embargo su uso ha decaído, enfocándose su empleo en sistemas embebidos, gracias a sus características de implementación (un bajo consumo energético y gran disponibilidad de herramientas de desarrollo y expertos concedores de la arquitectura), y desarrollo de iteraciones de alto rendimiento de esta arquitectura a favor de procesadores basados en la tecnología Intel IA64.

El diseño de la familia de CPU's MIPS destaca por presentar un conjunto de instrucciones claras. Su diseño ha influido de manera importante en otras arquitecturas de tipo RISC posteriores como los DEC Alpha.

2.-Historia del MIPS

Hablando un poco de su historia, el origen de la arquitectura MIPS se remonta al año 1981. En la universidad de Stanford, un equipo liderado por John L. Hennessy (fundador de MIPS Technologies) comienza a trabajar en lo que sería posteriormente el primer procesador MIPS. Su idea era mejorar a gran escala el rendimiento de la máquina a través del uso de la segmentación (que como ya sabemos, es un tipo de proceso paralelo consistente en el procesado simultáneo de instrucciones en diferente fase), tecnología por entonces de sobra conocida pero difícil de implementar. Su funcionamiento consistió en que la ejecución de una instrucción era dividida en varias etapas, comenzando por la “etapa 1” de una instrucción antes de que haya finalizado la ejecución de la instrucción anterior, en comparación a los diseños tradicionales, donde grandes áreas de la CPU se mantenían inactivas hasta la finalización completa de una instrucción. Además, la frecuencia del reloj de toda la CPU venía dictada por la latencia del ciclo completo, en lugar del camino crítico, es decir, la latencia de la etapa de segmentación que más tardaba en completarse.

Otro de los principales objetivos marcados por esta arquitectura fue el de que todas las subfases de las instrucciones (incluyendo el acceso a memoria) tardasen un único ciclo en completarse, dejando así de ser necesarios los bloqueos (anteriormente necesarios para asegurar el que las instrucciones que necesitaban varios ciclos de reloj para completarse dejaran de cargar datos desde los registros de segmentación, suponiendo una importante limitación en cuanto a la velocidad de trabajo), permitiendo un rendimiento de un solo ciclo.

Como resultado de esta idea de diseño se eliminaron ciertas instrucciones de gran utilidad, como cabe destacar el hecho de que las operaciones de multiplicación y división requerirían de varias instrucciones, pero se dio lugar a un conjunto que podría trabajar a un rendimiento muy superior al poder funcionar los chips a frecuencias mucho mayores, un logro bastante complicado debido a la intervención de bloqueos, cuyo tiempo necesario está determinado en función del tamaño del chip y la frecuencia del reloj (y la incorporación del hardware necesario para evitarlos reduciría notablemente la velocidad del sistema).

Como detalle curioso cabe destacar la fuerte crítica recibida por parte de observadores al diseño, derivadas sobre todo por dicha supresión de instrucciones, que afirmaban que las expectativas marcadas por la arquitectura MIPS eran un muro imposible de superar (y si no cómo explicar que si el sistema sustituía la operación de multiplicación por una serie de operaciones produciría un incremento de velocidad). El fallo de estas críticas sin fundamento residía en que se ignoraba que la velocidad del diseño residía en la segmentación, no en las instrucciones.

Con todo esto, en 1984 Hennessy decide dejar Stanford y fundar MIPS Computer Systems. Su primer diseño fue el R2000, presentado en 1985, optimizado posteriormente para dar lugar al R3000, lanzado al mercado en 1988. Se trata de CPUs de 32 bits que supusieron la base de la compañía en la década de los 80, y su uso se destinó a la integración en algunas series de Workstation de la compañía SGI, con diseños que diferían de los del ámbito académico en la implementación de la mayoría de los bloqueos con hardware y el proporcionamiento de instrucciones completas para la multiplicación y la división entre otros.

Las dificultades económicas atravesadas por la compañía durante el lanzamiento del R4000, primer procesador de 64 bits, y la necesidad de sus diseños por parte de la compañía SGI, provoca la compra de la compañía por parte de SGI en 1992, fecha aproximada a partir de la cual se comienzan a distribuir licencias de sus diseños a terceros, probando con éxito la simplicidad del núcleo, que permitía ser empleado en aplicaciones que hacían uso de diseños CISC menos capaces y de precio similares.

El sistema de licencias de MIPS se consolida en 1999 con el lanzamiento del MIPS32 y el MIPS64 (de 32 y 64 bits respectivamente), lo que resultó ser un éxito de ventas que llega hasta la actualidad, donde los núcleos MIPS son unos de los más importantes dentro del mercado de dispositivos como ordenadores de mano, decodificadores o sintonizadores de televisión.

Cerrando ya en cuanto a lo referente a la historia, hacer un pequeño inciso acerca de la familia de CPUs MIPS, que abarca desde el R2000, primer modelo comercial ,anunciado en 1985, que incluía instrucciones multiciclo para la multiplicación y división en una unidad independiente integrada en el procesador, con registros de 32 bits (sin registro de estado), soporte de hasta 4 coprocesadores, hasta nuevos modelos basados en el R10000 lanzado en 1995 (con un chip único de mayor velocidad de reloj que su antecesor, el R8000, mayores cachés primarios de 32 Kb para instrucciones y datos, implementado con una FPU más simple), podemos mencionar las siguientes arquitecturas de la familia:

- R3000, lanzado en 1988, añade una cache de 32 Kb posteriormente aumentada a 64 Kb para instrucciones de datos, junto con el soporte de “coherencia cache” para el uso del multiprocesador y una unidad de manejo de memoria (MMU). De esta rama derivan otros MIPS como el R3400, R3500 o R3900.
- R4000, presentado en 1991, extendía el juego de instrucciones para construir una arquitectura de 64 bits y movió la FPU a ese mismo circuito para crear un sistema chip único, operando a una velocidad de reloj muy superior, lo que implicó un recorte de la caché a 8 Kb y la necesidad de 3 ciclos de reloj para acceder a ellas. Derivados de él son el R4400 de 1993, con caches de 16 Kb, y los diseños de bajo coste R4200 y R4300 entre otros.
- El R5000 se encuadra junto con el R4700 y el R4650 desarrollados por la compañía Quantum Effects Devices, con diseños de grandes caches solo accesibles en 2 ciclos de reloj el uso eficiente del silicio y una FPU de rendimiento más eficiente y flexible que en anteriores versiones.
- El R7000 y R9000, también de QED, destinados a sistemas embebidos tipos redes e impresoras láser.
- R8000, primer diseño MIPS superescalar (que ejecutaba varias líneas de instrucciones simultaneas), capaz de procesar dos operaciones ALU y otras dos de memoria en cada ciclo de reloj, con un diseño basado en 6 chips (dos caches de 16 KB, una unidad de punto flotante, tres RAM de caché secundaria personalizable y un controlador de caché ASIC). Además se incluían dos unidades segmentadas de suma-multiplicación en doble precisión.

3.-Arquitectura del MIPS

Conocida ya algo de su historia y objetivos, pasamos a tratar de forma más profunda los componentes de la arquitectura MIPS (En nuestro caso vamos a centrar la descripción en el modelo R2000):

- **Procesador:** presenta un diseño interno compuesto por los siguientes elementos:
 - **(ALU) Unidad Aritmética y Lógica:** Unidad Aritmética entera, operaciones de multiplicación y división.
 - **(FPU) Unidad punto flotante:** Coprocesador dedicado al manejo de memoria caché y virtual.

El procesador del R2000 se basa en una arquitectura tipo Harvard (término con el que se alude a las arquitecturas de computadoras que utilizaban dispositivos de almacenamiento físicamente separados para las instrucciones y los datos, en oposición clara a la arquitectura de Von Neumann) con un banco de registros (apodado BR), constituido por:

- 32 Registros de 32 bits de propósito general para operaciones con enteros (GPRs (r0-r31), r0 tiene siempre valor 0, intentar cambiarlo no da error pero no su valor nunca cambia). Éstos se identifican por el carácter especial \$ seguido de un número de 0 a 31. Así por ejemplo el registro 1 se reconocerá cuando se vea la representación \$1.
- Coprocesador 0-CP0: Incorporado en el chip, provee funciones para apoyar al sistema operativo tales como manejo de excepciones, planificación de direccionamiento de memoria y control de recursos críticos.
 - Registro de estado (CP0reg12) estado del procesador y control.
 - Registro de causa (CP0reg13) causa de la interrupción más reciente.
 - Registro EPC(CP0reg14) contador de programa en la última interrupción.
 - Registro BadVAddr (CP0reg08) la dirección de la excepción de direccionamiento más reciente.
- 32 Registros de 32 bits de propósito general, que soportan el formato IEEE754(FPRs(f0-f31)) (estándar de la IEEE para la aritmética en punto flotante que define formatos para la representación de números flotante, incluyendo el 0, y valores desnormalizados, así como valores especiales como infinito y NaN, con un conjunto de operaciones en punto flotante que trabajan sobre esos valores) para operaciones en punto flotante de precisión simple y 16 registros de 64 bits para operaciones en punto flotante de doble precisión.

- 2 Registros especiales de 32 bits llamados respectivamente HI y LO, que sirven para almacenar los resultados de la multiplicación/división. Además también se pueden utilizar para operaciones de transferencia de datos.
- La Unidad Aritmético-Lógica, de 32 bits. Tratando algo más acerca de ésta entidad y su funcionamiento podemos indicar una descripción de las conexiones para una aritmética lógica:

En la primera fase se busca la instrucción, colocando el PC en la entrada de direcciones de la memoria de instrucciones (MI) la instrucción a ejecutar. El MI cual presenta una señal de lectura siempre activa y su salida será la instrucción. En esta implementación se incrementa simultáneamente el contenido del PC, aunque no se actualiza.

En la segunda fase podemos distinguir tres pasos que se realizan de forma simultánea: la decodificación de la instrucción, en la que se envían los bits correspondientes al código de operación a un circuito especial denominado unidad de control, encargada de enviar las señales apropiadas de lectura, escritura y selección a los elementos que lo precisen (el código de operación se emplea para poder distinguir entre los distintos tipos de instrucciones).

En una instrucción aritmética o lógica se requieren dos operandos. Es también en esta fase cuando de manera simultánea se buscan los operandos en el banco de registros.

Finalmente, como es una instrucción aritmética o lógica, además del código de la operación se precisa de un circuito de control especial denominado control de la ALU, para poder identificar la operación que la ALU debe realizar.

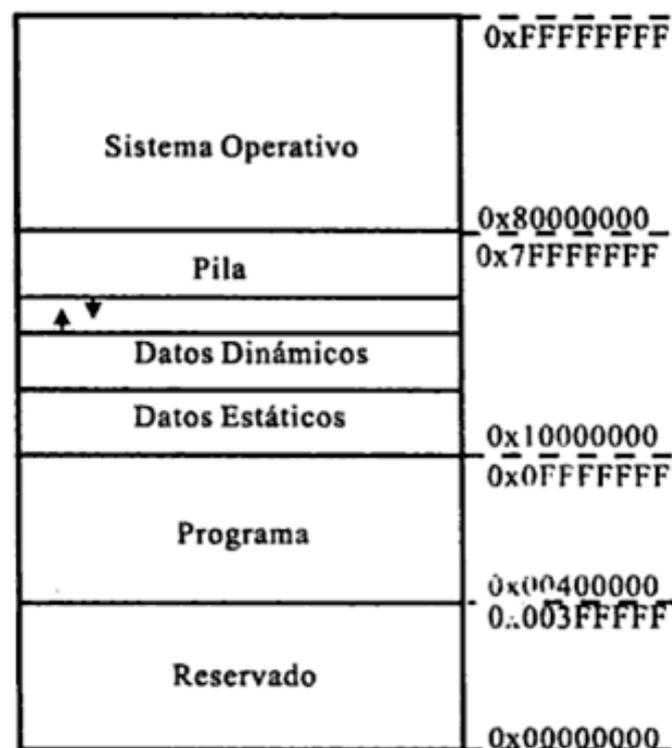
- Nombre de los registros, número, uso y convenciones de llamada:

Nombre Registro	Número	Uso
zero	0	Constante 0
at	1	Reservado para el assembler
v0	2	Para evaluación de expresiones y retorno de resultados de una función
v1	3	
a0	4	Argumento 1
a1	5	Argumento 2
a2	6	Argumento 3
a3	7	Argumento 4
t0	8	Temporal (no se preserva a través de los llamados)
t1	9	Temporal (no se preserva a través de los llamados)
t2	10	Temporal (no se preserva a través de los llamados)
t3	11	Temporal (no se preserva a través de los llamados)
t4	12	Temporal (no se preserva a través de los llamados)
t5	13	Temporal (no se preserva a través de los llamados)
t6	14	Temporal (no se preserva a través de los llamados)
t7	15	Temporal (no se preserva a través de los llamados)

Nombre Registro	Número	Uso
s0	16	Temporal que debe preservarse entre llamados a funciones
s1	17	Temporal que debe preservarse entre llamados a funciones
s2	18	Temporal que debe preservarse entre llamados a funciones
s3	19	Temporal que debe preservarse entre llamados a funciones
s4	20	Temporal que debe preservarse entre llamados a funciones
s5	21	Temporal que debe preservarse entre llamados a funciones
s6	22	Temporal que debe preservarse entre llamados a funciones
s7	23	Temporal que debe preservarse entre llamados a funciones
t8	24	Temporal (no se preserva a través de los llamados)
t9	25	Temporal (no se preserva a través de los llamados)
k0	26	Reservado para el núcleo del Sist. Operativo
k1	27	Reservado para el núcleo del Sist. Operativo
gp	28	Puntero al área global de datos
sp	29	Puntero al tope de la pila. Stack pointer
fp	30	Puntero a zona de variables en la pila. Frame pointer
ra	31	Dirección de retorno (usado en invocaciones a funciones)

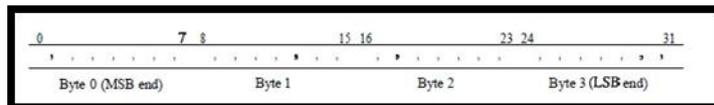
- Los registros preservados durante una llamada son aquellos que (por convenio) no serán modificados por una llamada de sistema o a un procedimiento o función. Por ejemplo, los registros \$s deben ser almacenados en la pila por el procedimiento que los necesita, siendo siempre incrementados en constantes \$sp y \$fp, para ser después decrementados una vez finalizado el procedimiento (se marca como disponible la memoria reservada). Por el contrario, \$ra es modificado automáticamente tras una llamada a una función normal (cualquiera que utilice la instrucción jal), y los registros \$t deben ser salvados por el programa antes de llamar a cualquier función (por si el programa necesita los valores contenidos en dichos registros tras la ejecución de la subrutina).
- **La memoria:** la memoria del MIPS presenta una serie de particularidades que enumeraremos a continuación:
 - Se denomina palabra (Word) al contenido de una celda de memoria y en el MIPS corresponde a 8 bits (diferente a la palabra de procesador que en este caso es de 32 bits). La arquitectura MIPS posee palabras de 32 bits, pero también otros tamaños de datos:
 - Bit: 0,1
 - 4 bits (nibble, que para los que no se acuerden, es el término con el que apelamos al conjunto de cuatro dígitos, y cuyo interés reside en que cada cifra en hexadecimal puede ser representada como un cuarteto, pues dieciséis es la potencia cuarta de 2, y que el cuarteto es la base del sistema de codificación BCD).
 - 8 bits – byte es utilizado cuando los datos que se desean manejar son del tamaño de 8 bits, caso de caracteres del código ASCII o bien en casos en los que el programador intenta acceder a los registros de 8 bits de algunas tarjetas de entrada/salida del computador.
 - 16 bits -- half-word. En el procesador cabe la posibilidad de operar con este tipo de datos de dos formas distintas con dos tipos de instrucciones diferentes: unas manejan datos enteros sin signo y las otras tienen en cuenta los signos de los operandos. En este último caso los datos son representados en complemento a dos.
 - 32 bits – Word se trata de un tipo que requiere una palabra de 32 bits para su representación, y que al igual que el de tipo Half, el procesador dispone de instrucciones para tratar dichos datos con o sin signo.
 - 64 bits – double-word

- Otra forma de organizar los diferentes simbolos del MIPS es mediante el tipo:
 - Character: ASCII Código de 7 bits por símbolo.
 - Decimal (BCD): Dígitos de 0-9 codificados desde 0000 hasta 1001.
 - Enteros: 32 bits (sin signo o complemento a 2).
 - Reales: 32 bits (precisión simple) o 64 bits (doble precisión) .
- Las palabras de 32-bits, bytes y medias-palabras pueden ser cargados en GPRs (general porpouse register, que se caracterizan por poder almacenar tanto datos como direcciones), después de ser cargadas, bytes y medias palabras rellenan con 0 o con el bit de signo los 32 bits. Solo las unidades de 32 bits pueden ser cargadas en FPRs (floating point register, que como indica su nombre, son empleados para guardar datos en formato “punto flotante”) y los números reales de 32 bits (precisión simple) se guardan en los FPRs pares. Los reales de precisión doble (64-bit) se guardan 2 FPRs consecutivos empezando por número par.
- **Organización de la memoria:** La dirección física de la memoria de instrucciones, en modo usuario, empieza en la dirección 0x00400000 y termina en la 0x0FFFFFFF. Mientras que la dirección de la memoria de datos, en modo usuario, empieza en la 0x10000000 y termina en la 0x7FFFFFFF.



- La memoria accesible por el usuario se encuentra dentro del rango [0x00400000, 0x7FFFFFFF]. Esta memoria es del tipo RAM (Random Access Memory) sobre la cual se pueden realizar operaciones de lectura escritura. La parte de memoria del programa sirve para contener el código de las aplicaciones que se ejecutan. Los segmentos de datos estáticos, datos dinámicos y pila contienen los datos que las aplicaciones utilizan en su ejecución. Los datos estáticos son aquellos que se definen y que permanecen a lo largo de la ejecución del programa; por lo que este fragmento de memoria puede crecer y decrecer. El segmento de pila se utiliza para transferirse los datos el programa principal y los procedimientos en las llamadas a subrutinas.
- **Alineación:** MIPS tiene restricciones de acceso a memoria porque utiliza el byte como mínima unidad con dirección: las palabras(32-bit) tienen que empezar en una dirección múltiplo de 4; así una dirección de una palabra $4n$ incluye 4bytes con direcciones $4n$ $4n+1$, $4n+2$ y $4n+3$. Las medias-palabras(16-bit) tienen que empezar en una dirección múltiplo de 2; así una dirección de una palabra $2n$ incluye 2bytes con direcciones $2n$ y $2n+1$. Las direcciones son dadas como un entero de 32 bits sin signo
- Dos formas de numerar los bytes contenidos en una palabra:

- **Big endian (IBM, Motorola, MIPS):** la dirección del byte más significativo termina en 00 (en binario), sólo si la palabra está alineada.



- **Little endian (Intel, Dec):** la dirección del byte menos significativo termina en 00 (en binario), sólo si la palabra está alineada.

4.-Instrucciones del MIPS:

Brevemente vamos a comentar un poco acerca de las instrucciones de MIPS R2000:

El conjunto de instrucciones permite realizar instrucciones de carga y almacenamiento desde y hacia memoria, tiene capacidad de desarrollar programas que resuelven problemas aritméticos y lógicos, y ofrece la posibilidad de controlar el flujo de la ejecución del programa mediante instrucciones de comparación y salto. En este último caso dispone de instrucciones de salto tanto condicional como incondicional. Una breve clasificación sería:

- Instrucciones aritméticas.
- Instrucciones lógicas.
- Instrucciones de salto incondicional.

Todas las instrucciones del MIPS R2000 tienen el mismo tamaño (32 bits). Se pueden clasificar en función de los elementos que utilizan (banco de registros, memoria de datos, ALU). Cada uno de los componentes que utiliza la instrucción se debe especificar en una serie de bits. Los distintos tipos de instrucciones constan de diferentes tamaños para los espacios reservados para esos bits (campos) es decir, utilizan diferentes formatos para codificar sus campos.

En el caso del MIPS R2000 se distinguen tres tipos de formatos de instrucción:

- Formato R o de tipo registro.
- Formato I o de tipo inmediato.
- Formato J o de salto incondicional (del inglés, jump).

5.-Bibliografía y referencias orientativas:

- http://gorrion.die.uchile.cl/~escverano/materialdocente/2_Disenio_Implementacion_Procesador_MIPS.pdf
- http://profesores.elo.utfsm.cl/~tarredondo/info/comp-architecture/paralelo2/C03_MIPS.pdf
- http://www.cse.ohio-state.edu/~teodores/download/teaching/cse675.au08/CSE675.02_MIPS-ISA_part1.pdf
- http://es.wikipedia.org/wiki/MIPS_%28procesador%29
- <ftp://db.stanford.edu/pub/cstr/reports/csl/tr/86/289/CSL-TR-86-289.pdf>

6.- Miembros del grupo

- *Alberto Hernández Cerezo*
- *Rodrigo Alonso Iglesias*
- *Cristian Tejedor García*



ETSII