MIPS (Microprocessor without Interlocked Pipeline Stages - Microprocesador sin fases de segmentación interbloqueadas)

Microprocesadores con arquitectura RISC (Reducted Instruction Set Computer - Cada instrucción tiene una sola función).

* Historia

Un equipo liderado por Hennessy comenzó a trabajar en el primer procesador MIPS, el cual comenzó como una idea para mejorar el rendimiento mediante el uso de la segmentación, el cual es un esquema de manejo de memoria donde la ejecución de una instrucción se divide en varias etapas.; comenzando la “etapa 1” de una instrucción antes de haber finalizado la ejecución de la instrucción; mientras, que los diseños tradicionales esperaban la finalización por completo de una instrucción antes de pasar a la siguiente.

El primer diseño exitoso de MIPS fue el R3000, luego, el R3000A, una versión aceleradas hasta los 40 MHz, fue utilizado en la Sony PlayStation.

MIPS presentó el R4000, su primer procesador de 64 bits, el cual tuvo dificultades en su lanzamiento al mercado.

SGI compró la compañía de Hennesy, para evitar perder el diseño de MIPS, y la compañía pasó a llamarse MIPS Technologies. Luego comenzó a otorgar licencias de los diseños a terceros.

Las últimas versiones fueron llamadas R16000 y R16000A.

* Microprocesadores RISC/CISC
* RISC

Alpha, MIPS, ARM, AVR, PowerPC.

* CISC (Complex Instruction Set Architecture-Cada instrucción tiene muchas funciones)

Motorola 68000 e Intel 80x86.

* Microprocesadores basados en MIPS

Ali, NEC VR4300 (Nintendo64), SGI Indigo.

* Principios de diseño de MIPS
* Simplicidad
* La simplicidad favorece la regularidad
* Instrucciones sencillas y limitadas.
* Registros limitados (32 registro)
* Modos de direccionamiento limitados
* La simplicidad favorece la regularidad
* Mejora el rendimiento a menor coste
* Principio de regularidad
* Todas las instrucciones son de tamaño fijo (4 bytes - 32 bits)
* Facilita la implementación
* El código de operación es siempre los 6 primeros bits.
* Cuanto más pequeño más rápido
* Un número alto de registros incrementa el tiempo de ciclo del reloj
* Un buen diseño necesita buenas soluciones de compromiso
* Guardar todas las instrucciones con la misma longitud, por eso se requieren diferentes clases de formatos para diferentes clases de instrucciones
* Hacer rápido los casos más comunes
* Como las operaciones aritméticas.
* Aplicaciones de MIPS
* En algunas consolas de videojuegos (PlayStation2 y PlayStation portable), Tablet, smartphones dispositivos Windows CE.
* La tecnología empleada en MIPS ha sido ofrecida como diseños de “IP-cores” (Intellectual Property- Módulos de propiedad intelectual- Bloques lógicos o datos de construcción) para sistemas embebidos (sistemas de computación diseñados para realizar una o algunas pocas funciones dedicadas, ​​ frecuentemente en un sistema de computación en tiempo real).
* Microchip Technology aprovecha los procesadores MIPS para sus microcontroladores de 32 bits (PIC32). Mobileye, con sus EyeQ2 y EyeQ3 se basan en núcleos con licencia de MIPS.
* Arquitecturas basadas en MIPS
* Cavium Networks (Fabricante de procesadores)
* Broadcom ( Procesadores para redes de datos y comunicaciones)
* IDT (Procesadores para comunicaciones de datos)
* PMC-Sierra (Especialista en redes)
* Loongson (Procesadores desarrollados en el ICT)

MIPS32

* Características
* Ancho de palabra y tamaño de los buses: 32 bits
* Tamaño de los datos en las instrucciones:
* Byte: 8 bits, B
* Halfword: 16 bits, H
* Word: 32 bits, W
* Doubleword: 64 bits, D
* Arquitectura de carga/almacenamiento:
* Todo dato debe ser cargado previamente en un registro de propósito general, antes de ser utilizado en una instrucción aritmética.
* Instrucciones aritméticas con 3 operandos (a = b + c) de 32 bits de registros.
* Esquema de bus único para memoria y E/S

MIPS R2000

Principalmente utilizan registros y números a la hora de operar con los datos. A veces es necesario acceder a la memoria para encontrar los operando.

* Procesador MIPS = CPU + coprocesadores auxiliares
* CPU (Registros - $0,$1,…$31): Unidad Aritmética y Multiplicar-Dividir (HI, LO)
* Coprocesador 0 = (Soporta) excepciones, interrupciones y sistema de memoria virtual
* Coprocesador de control de sistema
* Maneja los cambios en el modo de ejecución (usuario, núcleo, supervisor)
* Traduce direcciones virtuales en físicas
* Registros: BadVAdress, Cause, Status, FPC
* Controla el subsistema de memoria caché
* Coprocesador 1 = FPU(Unidad de Punto Flotante)
* Registros ($0,$1,…,$31): Unidad aritmética

Registros

Es una memoria integrada en la CPU. Su capacidad de 4 bytes (32bits) y de acceso muy rápido.

Se usan para guardar valores temporales y resultados de operaciones aritméticas.

El compilador asocia las variables con los registros.

Las variables se guardan en la memoria principal

La mayoría de las CPUs modernas mueven las variables de la memoria principal a los registros, operan sobre ellos y regresa el resultado a la memoria (Arquitectura load/store), MIPS emplea esta arquitectura.

La Arquitectura carga/almacenamiento divide las instrucciones en dos categorías: acceso a la [memoria](https://en.wikipedia.org/wiki/Memory_(computing)) ( [carga y almacenamiento](https://en.wikipedia.org/wiki/Load_and_store) entre la memoria y los [registros](https://en.wikipedia.org/wiki/Processor_register) ) y operaciones (que solo ocurren entre los registros). Lo contrario a una Arquitectura registro/memoria, donde uno de los operandos de la suma puede estar en la memoria y el otro operando en el registro, como en la arquitectura CISC.

* 32 registros de propósito general de 32 bits:
* r0 = $zero
* r2-r3 = $v0-$v1 (valores devueltos en funciones, valores de resultado)
* r4-r7 = $a0-$a3 (argumentos en funciones)
* r8-r15 = $t0-$t7 (registros (valores) temporales, pueden ser sobrescritos por el destinatario)
* r16-r23 = $s0-$s7 (registros salvados o guardadas por el destinatario)
* r28 = $gp (puntero global para datos estáticos)
* r29 = $sp (puntero de pila)
* r30 = $fp (puntero de frame(marco))
* r31 = $ra (dirección de retorno)
* Registros con propósito específico:
* HI: En producto, almacena los 32 bits más significativos del resultado. En divisiones almacena los 32 bits del resto.
* LO: En producto, almacena los 32 bits menos significativos del resultado. En divisiones almacena los 32 bits del cociente.
* PC: Contador de programa. Al inicio de cada ciclo, almacena la dirección de memoria que contiene la instrucción que se va a ejecutar.
* Aritmética MIPS

Todas las instrucciones aritméticas tienen 3 operandos: add $s0, $s1, $s2 ($s0 = $s1 + $s2)

Los operandos de las instrucciones aritméticas tienen que ser registros

Organización de la memoria

Se usan 32 bits para direccionar la memoria.

La memoria tienebytes con direcciones desde 0 hasta .

La memoria tiene palabras con direcciones desde 0 hasta .

MIPS usa byte adressing (se refiere a arquitecturas de hardware que admiten el acceso a [bytes](https://en.wikipedia.org/wiki/Byte) individuales de datos en lugar de solo a unidades más grandes como words), el índice apunta a un byte de memoria.

0 - 8 bits de memoria

1 - 8 bits de memoria

2 - 8 bits de memoria

3 - 8 bits de memoria

4 - 8 bits de memoria

5 - 8 bits de memoria

…

* La memoria está alineada. Las palabras (Word) deben comenzar en direcciones que son múltiplo de 4:

0 - datos 32 bits

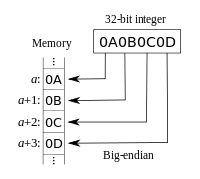
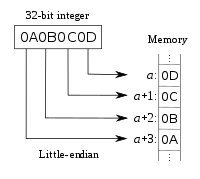
4 - datos de 32 bits

8 - datos de 32 bits

12 - datos de 32 bits

…

* MIPS se puede configurar por hardware como big-endian (byte más significativo del número se guarda en la dirección baja de la memoria) o little–endian. Se usan para referirse a las dos formas en que se pueden guardar los números que ocupan más de un byte.



* Instrucciones de referencia a memoria
* Instrucción lw (load word – carga palabra(32bits-4 byte)), el destino va primero
* lw rd, num(rs)  → carga 4 bytes de dir = num + rs a rd
* lw $t0, 32($s3) → $t0=Memoria[$s3+32]
* Instrucción sw (store word – almacena palabra), el destino va segundo
* sw rt, num(rs) → almacena 4 bytes de rt en dir = num + rs
* sw $t0, 32($s3) → Memoria[$s3+32]=$t0
* Operandos de memoria

lw $t0, 32($s3) → $t0=Memoria[$s3+32]

El registro ($t0) se llama registro base y la constante (32) se llama offset.

MIPS carga palabras pero direcciona bytes.

Lenguaje de máquina

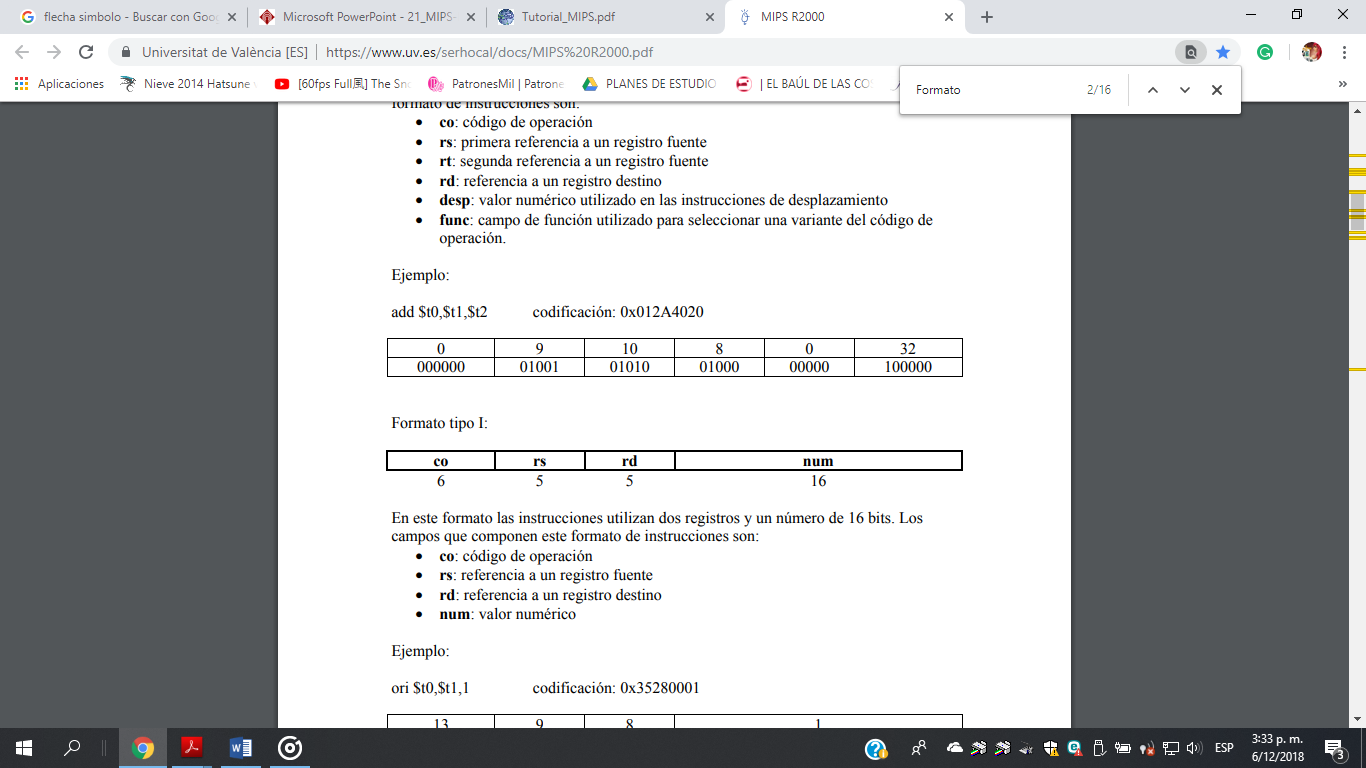
Es lenguaje binario cuyo hardware solo entiende bits. El ensamblador traduce cada instrucción de lenguaje ensamblador a lenguaje de máquina.

* add destino, fuente1, fuente2 → 000000fffffgggggdddddxxxxx100000
* 000000 es el código del add.
* fffff es el número del registro fuente 1
* ggggg es el número del registro fuente 2
* ddddd es el número del registro destino
* xxxxx don’t care
* 1000000 es el código de la suma de registros

Formato de Instrucción

* Formato R
* Solo utilizan registros.
* add $t0, $t1, $t2

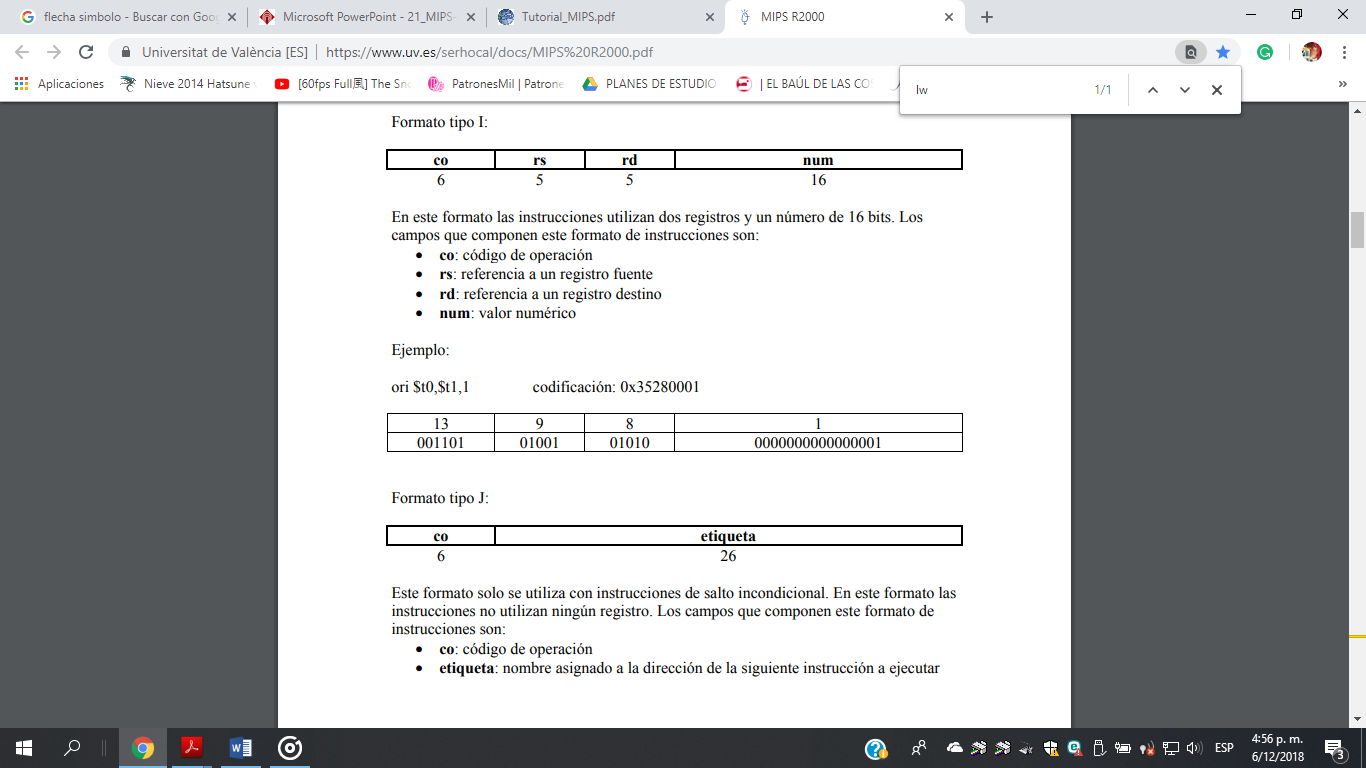
Codificación: 0x012A020

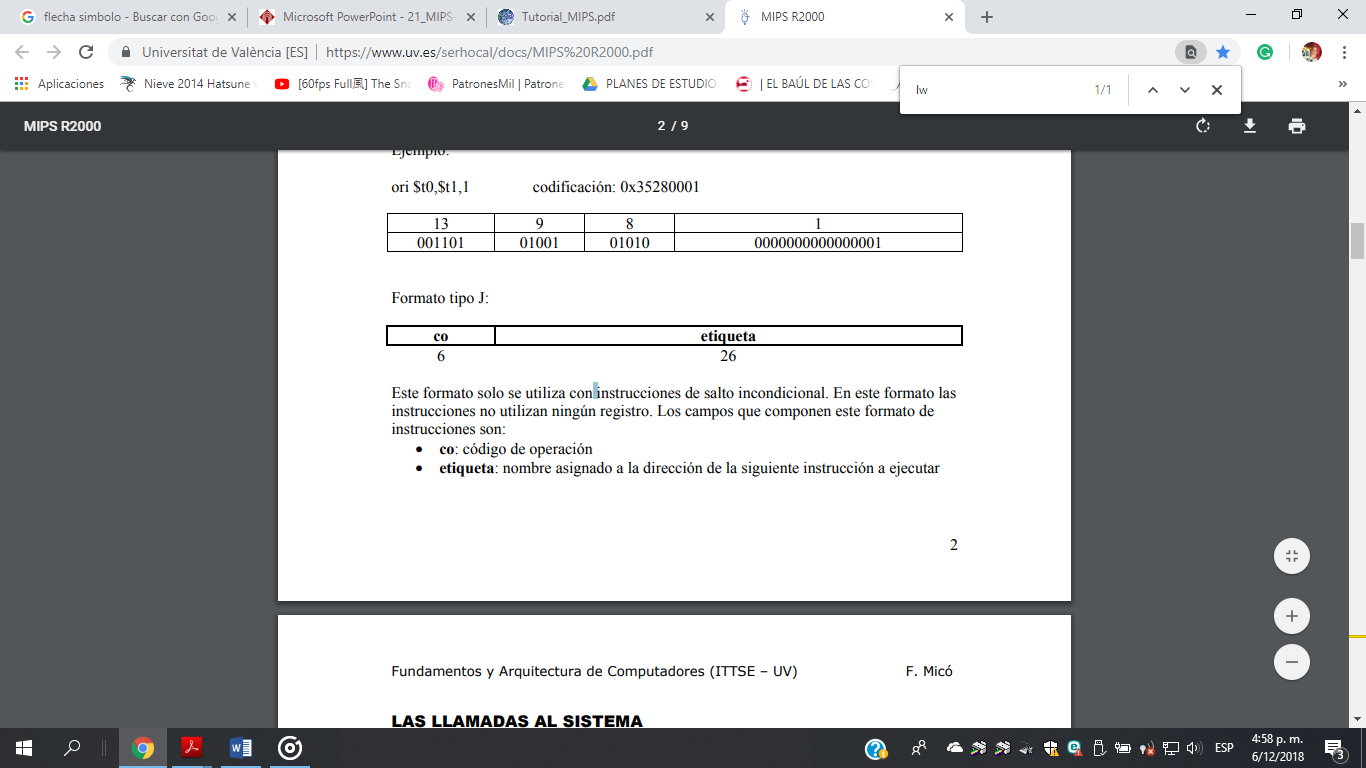


* Formato I
* Se usan dos registros y un número de 16 bits (operando inmediato)
* Antes de ser utilizado en una instrucción aritmética, todo dato debe ser cargado previamente en un registro de propósito general

Ori $t0, $t1, 1

Codificación 0x35280001



* Formato J
* Solo se utiliza con instrucciones de salto incondicional
* Solo existen 3 instrucciones de salto incondicional:
* j C → salta a la dirección C
* jr $r → salta a la dirección guardada en $r
* jal C → llama al procedimiento que comienza en C, la dirección de regreso de guarda en $31, es decir, regresa con jr $31}

Instrucciones de control

Instrucciones de toma de decisión que alteran el flujo de control (cambian la siguiente instrucción que será ejecutada).

beq $s, $t, C → salta a la dirección C si $s == $t

bne $s, $t, C → salta a la dirección C si $s != $t

slt $t0, $t1, $t2 → $t0 vale 1 si $t1 < $t2, caso contrario vale 0

Constantes

Están limitadas a 16 bits con signo.

Se pueden tener constantes de 32 bits solo si se almacenan en un registro.

addi $x, $zero, C

o

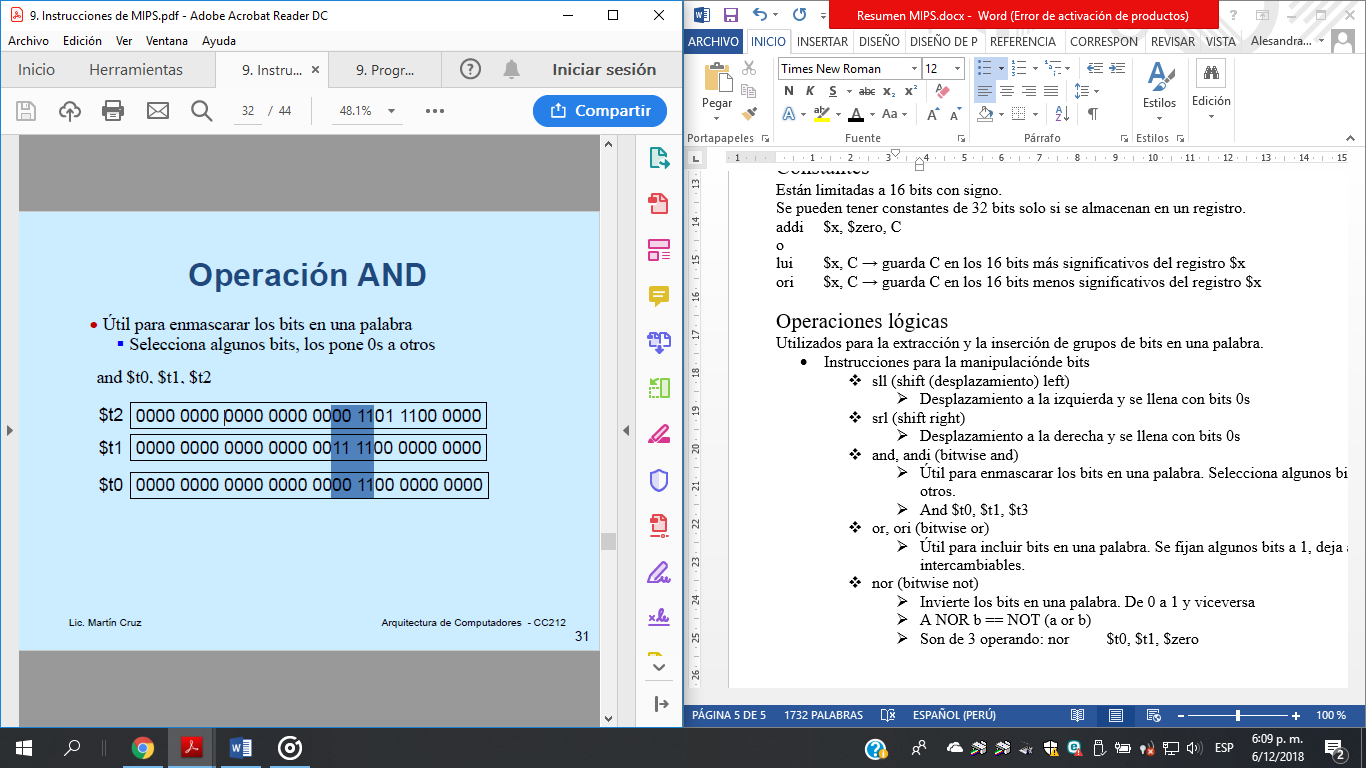
lui $x, C → guarda C en los 16 bits más significativos del registro $x

ori $x, C → guarda C en los 16 bits menos significativos del registro $x

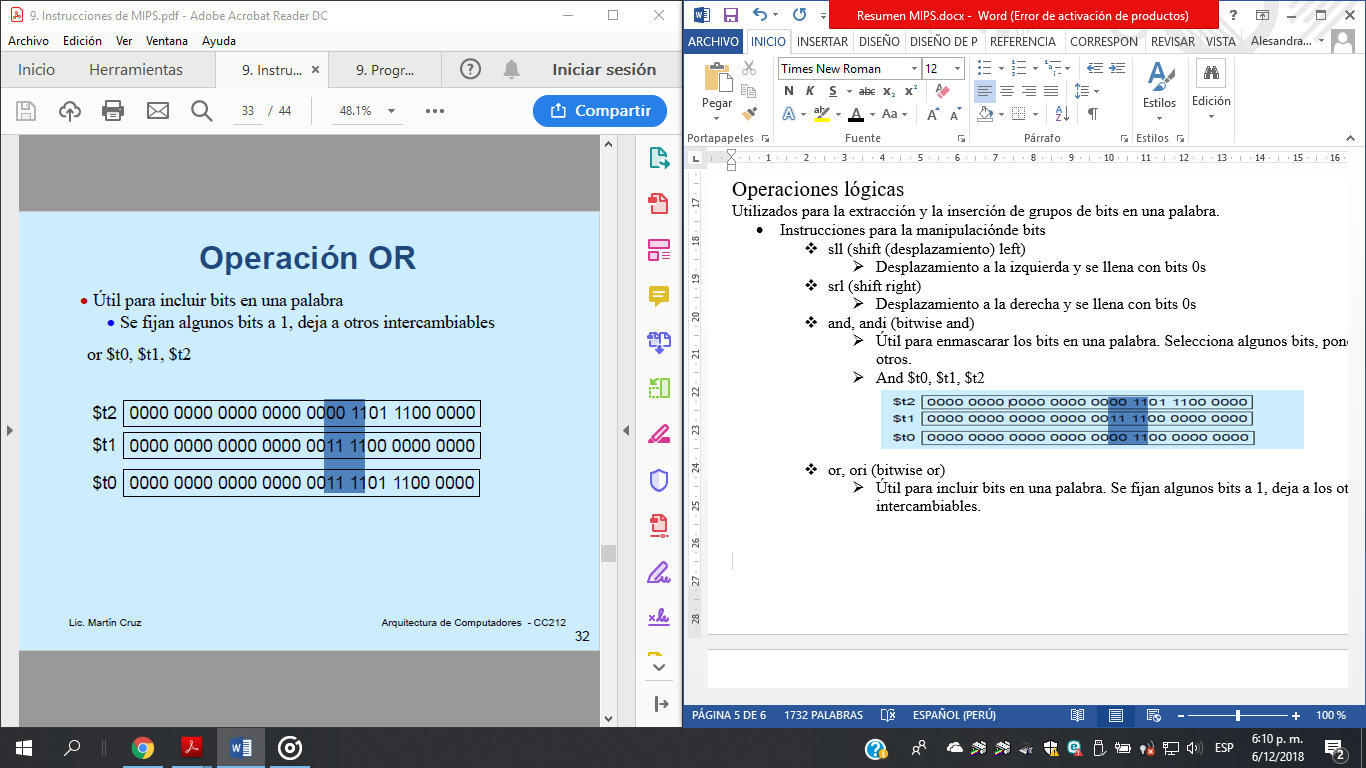
Operaciones lógicas

Utilizados para la extracción y la inserción de grupos de bits en una palabra.

* Instrucciones para la manipulaciónde bits
* sll (shift (desplazamiento) left)
* Desplazamiento a la izquierda y se llena con bits 0s
* srl (shift right)
* Desplazamiento a la derecha y se llena con bits 0s
* and, andi (bitwise and)
* Útil para enmascarar los bits en una palabra. Selecciona algunos bits, pone 0s a los otros.
* And $t0, $t1, $t2



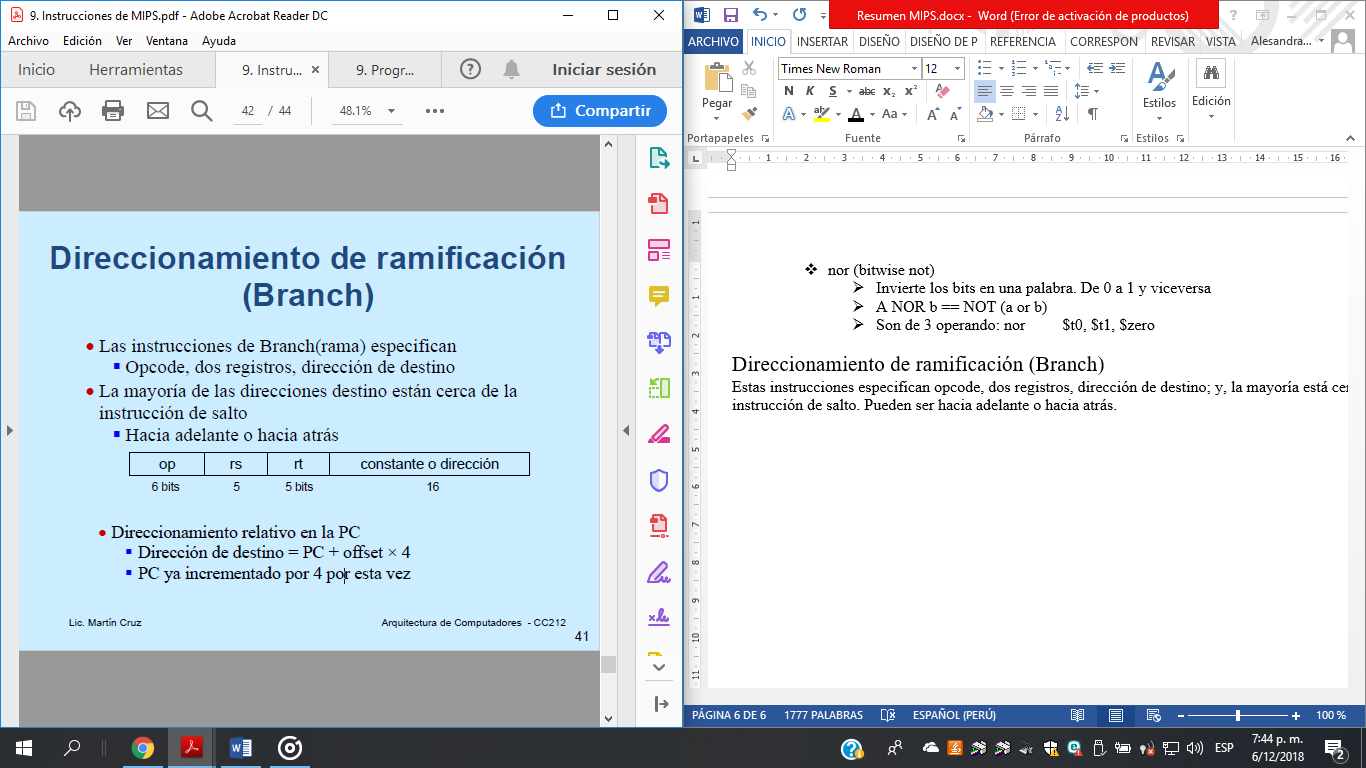
* or, ori (bitwise or)
* Útil para incluir bits en una palabra. Se fijan algunos bits a 1, deja a los otros intercambiables.



* nor (bitwise not)
* Invierte los bits en una palabra. De 0 a 1 y viceversa
* A NOR b == NOT (a or b)
* Son de 3 operando: nor $t0, $t1, $zero

Direccionamiento de ramificación (Branch)

Estas instrucciones especifican opcode, dos registros, dirección de destino; y, la mayoría está cerca de la instrucción de salto. Pueden ser hacia adelante o hacia atrás.



Uso de la memoria

Solo las instrucciones de carga y almacenamiento pueden acceder a la memoria.

Los sistemas basados en MIPS dividen la memoria en 3 segmentos:

* Segmento de pila (stack segment)
* Es donde se guardan los stack frames ( marcos de pila)
* Segmento de datos (data segmente)
* Datos dinámicos: Variables que se crean durante el programa
* Datos estáticos: Variables de tamaño fijo y que necesitan ser accesados durante todo el programa como las variables globales.
* Segmento de texto (text segmente)
* Es donde se guarda el código del programa

Llamadas a sistema

Los simuladores ofrecen los servicios (como leer el teclado, escribir a la consola y manejar archivos del sistema) de tipo sistema operativo a través de la instrucción syscall.

Para pedir un servicio, el programa carga el número de servicio en $v0 y los argumentos en $a0-$a3, si el servicio regresa un valor, lo hace en $v0.

Procesador MIPS

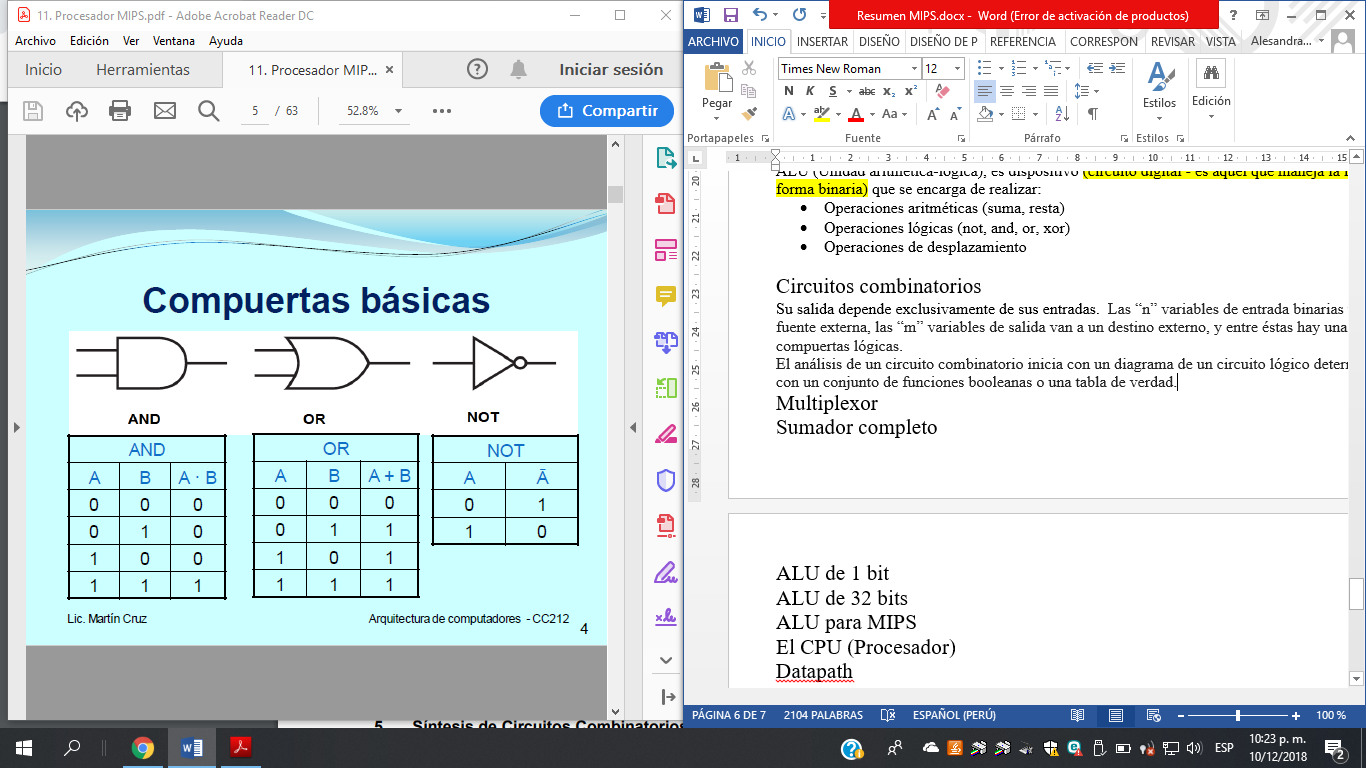
ALU (Unidad aritmética-lógica), es dispositivo (circuito digital - es aquel que maneja la información en forma [binaria](https://es.wikipedia.org/wiki/Sistema_binario)) que se encarga de realizar:

* Operaciones aritméticas (suma, resta)
* Operaciones lógicas (not, and, or, xor)
* Operaciones de desplazamiento

Circuitos combinatorios

Su salida depende exclusivamente de sus entradas.  Las “n” variables de entrada binarias vienen de una fuente externa, las “m” variables de salida van a un destino externo, y entre éstas hay una interconexión de compuertas lógicas.

El análisis de un circuito combinatorio inicia con un diagrama de un circuito lógico determinado y culmina con un conjunto de funciones booleanas o una tabla de verdad.



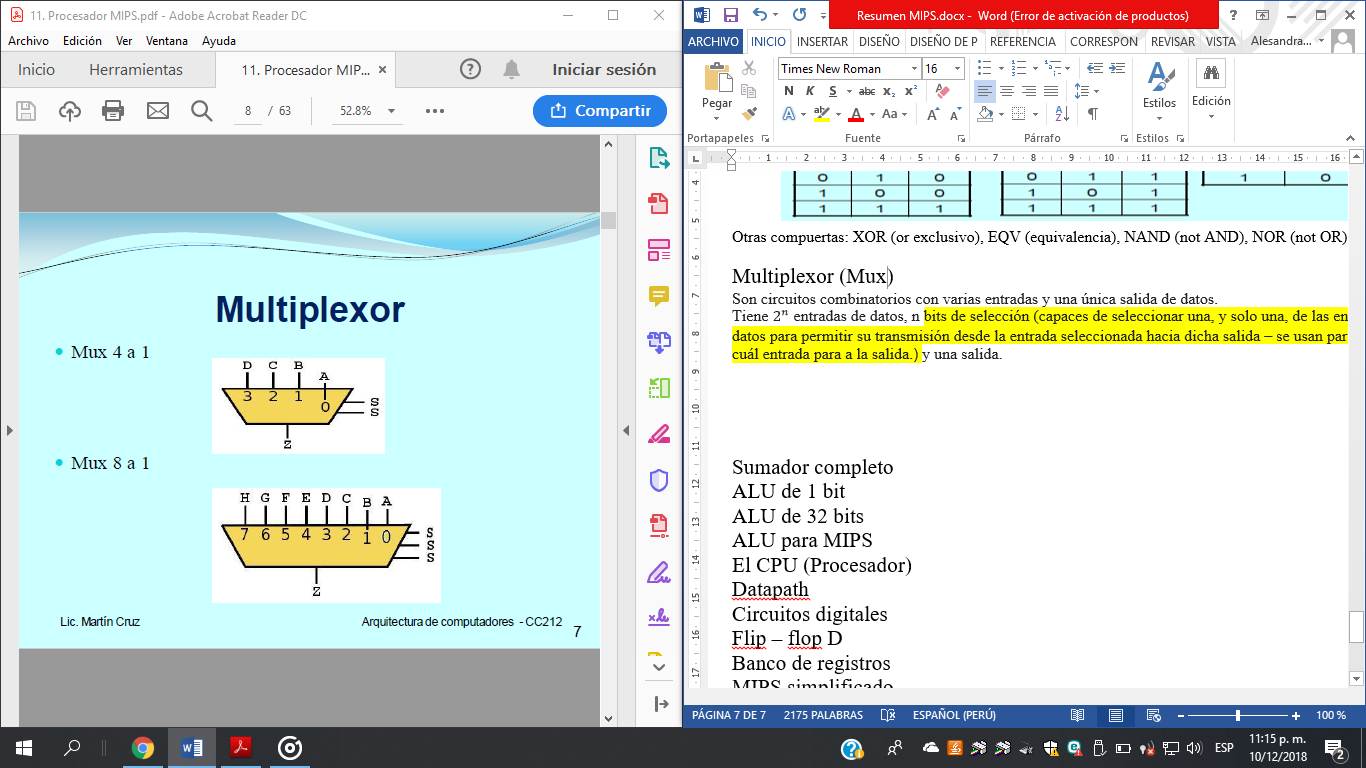
Otras compuertas: XOR (or exclusivo), EQV (equivalencia), NAND (not AND), NOR (not OR).

Multiplexor (Mux)

Son circuitos combinatorios con varias entradas y una única salida de datos.

Tiene entradas de datos, n bits de selección (capaces de seleccionar una, y solo una, de las entradas de datos para permitir su transmisión desde la entrada seleccionada hacia dicha salida – se usan para decidir cuál entrada para a la salida.) y una salida.

Mux 4 a 1:



Sumador completo

Es un circuito combinatorio que realiza la suma de dos bit y un acarreo de entrada, obteniendo en su salida la suma y un carry de salida.

* Entrada: 2 números de 1 bit y un bit de carry de entrada.
* Salida: La suma de 1 bit y un bit de carry de salida.



ALU de 1 bit

Realiza 2 operaciones: AND y OR; y, posee un bit para seleccionar entre estas dos operaciones.



* Agregando un sumador completo: 3 operaciones (AND, OR, suma) con 2 bits para seleccionar operación.



Operación 00: Resultado (a.b), CarryOut (X)

Operación 01: Resultado (a+b), CarryOut (X)

Operación 10: Resultado (a ⊕ b ⊕ CarryIn), CarryOut ((a+b).CarryIn+ (a.b))

Operación 11: Resultado (X), CarryOut (X)

El CPU (Unidad central de procesamiento - Procesador)

Es el hardware dentro de un ordenador que interpreta las instrucciones de un programa informático mediante

Consta de 2 componentes:

* Datapath: Es una colección de unidades funcionales que procesan datos y direcciones (lw y sw) dentro del CPU: Registros, ALUs, multiplexores, memorias, etc.
* Control: Extrae instrucciones de la [memoria](https://es.wikipedia.org/wiki/Memoria_(inform%C3%A1tica)), las decodifica y las ejecuta, llamando a la ALU cuando sea necesario. Ordena el datapath, memoria y dispositivos de entrada/salida, lo que hay que hacer de acuerdo al programa.

Datapath

Realiza operaciones aritméticas y lógicas (el datapath utiliza banco de registros para leer operandos y guardar resultados y ALU para realizar operaciones aritméticas y lógicas)

Elementos:

* ALU
* Contador de programa (PC): Para guardar la dirección de la instrucción actual. Indica la posición donde está el procesador en su secuencia de instrucciones.
* Memoria de instrucciones: Para guardar y leer instrucciones.
* Memoria de datos: Almacena los datos (lw – lee de la memoria de datos y sw-escribe en la memoria de datos).
* Banco de registros: Para leer y escribir registros.
* Sumadores: Para incrementar la PC

Primera parte:

* Ciclo Fetch: Lee una instrucción e incrementa el PC.

Segunda parte:

* Implementación de las instrucciones aritméticas y lógicas
* Implementación de carga y almacenaje (load/store)

Memoria de datos

* 4 entradas:
* Dirección del registro
* Dato a escribir (cuando es escritura)
* Señal de modo de lectura
* Señal de modo de escritura
* 1 salida:
* Dato leído (cuando es lectura)

Circuitos digitales

Es aquel que maneja la información en forma binaria, es decir, con valores de "1" y "0".

* Circuitos combinatorios: La salida depende solo de las entradas.
* Circuitos secuenciales: La salida depende de las entradas y de la salida actual. Como los Latches y flip-flops. Pueden almacenar 1 bit.

Flip – flop (biestables – maestro-esclavo de datos)

Son dispositivos de dos estados, que sirven como memoria básica para las operaciones de lógica

Secuenciales.

Son usados para el almacenamiento y transferencia de datos digitales (“1” y “0”). Se usan normalmente para

los registros, para almacenar datos numéricos binarios.

La salida se actualiza durante el flanco de reloj.

* Flip - flop D: El término "D", significa dato; este "flip-flop" almacena el valor que está en la línea de datos. Se puede considerar como una celda básica de memoria

Banco de registros

Es un conjunto de registros para guardar y leer datos. Con dos puertos de lectura y uno de escritura.

Se puede leer y escribir el mismo registro durante el mismo ciclo de reloj.

* Para leer un registro (se lee lo que se escribió un ciclo anterior):
* Entrada: Se indica el número de registro
* Salida: Dato contenido en el registro
* Para escribir un registro (se hace durante los flancos del reloj):
* Entrada: Se indica el número de registro y dato que se va a escribir; y una señal de reloj para controlar la escritura.
* Lo escrito está disponible en el siguiente ciclo.

MIPS simplificado

Las instrucciones se hacen en un ciclo de reloj, comenzando a ejecutarse en un flanco de reloj y terminando en el siguiente flanco.

Tres tipos de instrucciones:

* De referencia de memoria (carga/almacenamiento): Load Word (lw- copia información desde memoria principal a los registros.) y Store Word (sw - copia información de un registro a memoria principal).
* Aritmético
* Lógicas

Implementación

La implementación de las distintas instrucciones tiene varias acciones en común. Los primeros dos pasos son iguales:

1. Enviar el PC (contador de programa) a la memoria y sacar la siguiente instrucción (ciclo de fetch).
2. Leer uno o dos registros.

Lo siguiente depende de la clase de instrucción.

Todas las instrucciones, excepto el salto incondicional (instrucción j), usan ALU.

Después de usar ALU:

* Las instrucciones de referencia a memoria acceden a la memoria para cargar o guardar un dato.
* Las instrucciones aritmético-lógicas guardan el dato de la ALU en un registro
* Los saltos, dependiendo de la condición, cambian el contador de programa (PC) o lo incrementan en 4.

Fase Fetch

Es el período que tarda el CPU en ejecutar una instrucción de lenguaje máquina. Comprende una secuencia de acciones determinada que debe llevar a cabo la CPU para ejecutar cada instrucción en un programa. Cada instrucción del [juego de instrucciones](https://es.wikipedia.org/wiki/Conjunto_de_instrucciones) de una CPU, puede requerir diferente número de ciclos de instrucción para su ejecución. Un ciclo de instrucción está formado por uno o más [ciclos máquina](https://es.wikipedia.org/w/index.php?title=Ciclo_m%C3%A1quina&action=edit&redlink=1).

Instrucciones de Formato R

Lee dos registros que son operandos en la instrucción, ejecuta la operación aritmética/lógica y escribe el resultado en el registro.

Política de reloj

* Se suponen circuitos secuenciales disparados por el flanco de reloj.
* Las partes combinatorias reciben entradas de alguna parte secuencias y escriben sus salidas en alguna otra parte secuencial.
* Los elementos secuenciales son los únicos que pueden guardar valores.
* Las señales deben propagarse de los elementos secuenciales 1 a 2 en un ciclo de reloj.
* El tiempo de propagación determina el tamaño del pulso de reloj.

Instrucciones load/store

Se necesita la unidad de extensión de signo y memoria de datos (almacena los datos - .data)

El datapath para estas instrucciones usa:

* Banco de registros
* ALU
* Unidad de extensión de signo para mover constantes de 16 a 32 bits
* Memoria de datos

Locality

* Locality
* Principio que establece que en un instante dado un programa accesa un espacio pequeño de su espacio de direcciones
* Locality en tiempo
* Si se accede a una dirección de memoria, es posible que vuelva a ser usado pronto.
* Locality en espacio
* Si se accede a una dirección de memoria, es posible que las direcciones de memoria que se encuentras próximas a ella sean usadas pronto.

Bloques

Unidad mínima de información que puede o no estas presente en una jerarquía de 2 niveles.

Incrementar el tamaño del bloque generalmente reduce la tasa de fallos.

La tasa de fallos aumenta si el bloque es demasiado grande comparado con el tamaño del caché.

Memoria caché

Es el nivel de memoria situada entre el procesador y la memoria principal.

* Cachés de datos
* Guardan los últimos datos referenciados
* Cachés de instrucciones
* Guardan las últimas instrucciones ejecutadas
* Cachés de trazas
* Guardan secuencias de instrucciones para ejecutar que no son necesariamente adyacentes.

Estrategias que se usan

* Caché de mapeo directo
* A cada dato se le asigna un lugar en el cache de acuerdo a su dirección en la memoria principal.
* Cache asociativo total
* Caché asociativo por conjunto

Etiquetas y bit válido

Cada bloque en el caché tiene una etiqueta, la cual tiene la información necesaria para identificar si el dato en el caché es el dato buscado.

En el mapeo directo, la etiqueta tiene los bits altos de la dirección del dato.

Cada bloque tiene un bit llamado bit válido.

Mapeo de escrituras

Una escritura (write) es el resultado de un store. El store debe escribir en el caché y en la memoria, de otro modo sería inconsistentes.

Dos estrategias que se usan para el manejo de escrituras:

* Write-through
* Escribir cada vez en el caché y en la memoria.
* Escribir en la memoria es lento
* Necesita un ciclo de reloj para escribir
* Write-back:
* Escribir solo en el caché y copiar el dato a la memoria cuando la entrada en el caché va a ser reemplazado
* El bloque del caché se escribe en la memoria solo si es necesario
* El bloque tiene un bit llamado bit inconsistente, si este está pagado, el bloque no fue modificado y puede ser reemplazado; si está prendido, antes de ser reemplazado, debe escribirse en la memoria.
* Necesita dos ciclos de reloj para escribir

Memoria virtual

Actúa como un caché de la memoria secundaria (disco duro).

Posee mecanismos para asegurar que múltiples procesos que compartan la CPU, memoria o dispositivos de E/S no interfieran uno con otro.

La memoria virtual funciona por el principio de localidad (un programa es probable que acceda a una parte relativamente pequeña de su espacio de direcciones durante cualquier periodo de tiempo)

Bloque = página

Fallo = falta o fallo de página

La CPU produce una dirección virtual, que es traducida por hardware y software como una dirección física que usa para acceder a la memoria principal.

Fallo de página

Ocurre cuando el bit válido es falso.

El sistema operativo toma el control, interrumpiendo el proceso y guardando su estado; luego, busca en la página en el área de swap del disco duro y la carga en la memoria principal, si la memoria está llena, se necesita reemplazar una página.

Área de swap

Área del disco reservada para todo el espacio de memoria virtual de un proceso.

Protección

El sistema operativo debe asegurar que un proceso no escriba en el espacio de direcciones de otro proceso.

Un proceso no debe accesar los datos de otro.

Un proceso de usuario no puede cambiar su propia tabla de páginas, solo lo puede hacer el sistema operativo.

Reducir la tasa de fallos

* Usar páginas grandes para tomar ventaja de la locality espacial.
* Que el sistema operativo usa técnica como LRU.

Características del sistema de E/S

* Fiabilidad
* Especialmente para dispositivos de almacenamiento
* Medidas de desempeño
* Tiempo de respuesta
* Rendimiento del ancho de banda
* Principalmente interesados en el tiempo de respuesta y la diversidad de los dispositivos
* Servidores
* Interesados en el rendimiento y la capacidad de expansión de los dispositivos

Algunos dispositivos de almacenamiento

* Discos
* No volátil
* Almacenamiento flash
* Semiconductor no volátil
* Más pequeño y de bajo consumo
* Más costo entre más GB tenga.
* Tipos:
* Flash NOR
* Celdas como una compuerta NOR
* Utilizado para memoria de instrucciones
* Flash NAND
* Celdas como una compuerta NAND
* Más barato por GB

Interconectando componente

Se necesitan interconexiones entre el CPU, la memoria y los controladores de E/S.

Se usan buses, aunque son limitados por factores físicos como la longitud del cable y el número de conexiones.

Buses (canal de comunicación compartido)

Conjunto de cables que permiten el intercambio de datos entre la CPU y el resto de unidades.

Pueden ser Firewire, USB, PCI Express, SATA, SAS.

* Buses procesador-memoria
* De pequeña longitud y de alta frecuencia
* Diseñado para maximizar el ancho de banda procesador-memoria
* Buses de E/S
* Son más largas, por lo cual permite múltiple conexiones
* Especificado por normas para interoperabilidad
* Se conectar al bus procesador-memoria a través de un puente
* Buses gráficos
* Características diferentes y funciones especiales

Señales y sincronización en el Bus

* Líneas de datos
* Lleva dirección y datos
* Multiplexado o separado
* Líneas de control
* Indica tipos de datos
* Sincroniza transacciones
* Síncrono
* Incluye un reloj en las líneas de control y un protocolo de comunicaciones que se referencia a la señal de reloj
* Asíncronas
* No tiene reloj
* Puede adaptarse a una amplia variedad de dispositivos
* Utiliza un protocolo con acuse de recibo (es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción de un mensaje).

Gestión de la entrada/salida

Es mediado por el sistema operativo el cual actúa como interfaz entre el hardware y el programa que solicita la operación de E/S.

* A menudo usan interrupciones para comunicar información sobre operaciones de E/S.
* Diferentes programas que usan el procesador comparten los recursos de E/S
* El sistema operativo proporciona abstracciones para el acceso a los dispositivos

Direccionamiento de dispositivos de E/S

Para enviar un comando a un dispositivo de E/S, el procesador debe seleccionar o direccionar el dispositivo y proveer comandos.

Se puede direccionar mediante dos métodos:

* E/S asignada al espacio de memoria, donde se asignan porciones del espacio de direccionamiento a los dispositivos de E/S
* Con instrucciones especiales de E/S, donde se debe especificar el dispositivo y comando.

Comando de E/S

Causan que el dispositivo de E/S realice algo

En un proceso de interacción con el dispositivo se tiene:

* Registro de estado
* Indica la finalización en la tarea de un dispositivo y la ocurrencia de errores
* Registro de datos
* Escritura: Transfiere datos a un dispositivo
* Lectura: Transfiere datos desde un dispositivo

Comunicación con el procesador

Mediante:

* Encuestas
* Comprueba periódicamente el estado de un dispositivo de E/S
* Si el dispositivo está listo, realiza la operación, caso contrario, toma medidas al respecto.
* Comúnmente usado en sistemas de bajo rendimiento
* En algunos sistemas se considera una pérdida de tiempo del CPU.
* E/S dirigida por interrupciones
* Emplea interrupciones para indicar al procesador que un dispositivo de E/S necesita atención
* Cuando un dispositivo está listo o hay errores, interrumpe al CPU
* Los dispositivos que necesitan atención más urgente consiguen prioridad más alta (prioridad de interrupciones)
* DMA (acceso directo a memoria)
* Proporciona a un controlador de dispositivo la capacidad de transferir datos desde la memoria sin que el procesador participe.
* El sistema operativo suministra dirección de inicio en memoria
* El controlador interrumpe al completar la operación o cuando sucede un error.

Error: en el pico de rendimiento

El pico en las tasas de E/S es casi imposible de alcanzar

* Generalmente algún otro componente del sistema limita el rendimiento, como en la transferencia sobre el bus de datos a la memoria, donde puede haber problemas con otros buses.

Programación paralela

Es una forma de programación en la que muchas [instrucciones](https://es.wikipedia.org/wiki/Instrucci%C3%B3n_(inform%C3%A1tica)) se ejecutan simultáneamente, operando sobre el principio de que problemas grandes, a menudo se pueden dividir en unos más pequeños, que luego son resueltos simultáneamente.

Tiene como dificultades el particionamiento, la coordinación (tiempo de sincronización) y la sobrecarga de las comunicaciones.

Ley de Amdahl

Es un modelo que es utilizada para evaluar cómo cambia el rendimiento al mejorar una parte de la computadora. Dice que la mejora de rendimiento de un sistema debido a un cambio está limitada por el tiempo que se utiliza dicha mejora.

Todo programa consta de:

* Una o más porciones que no se pueden paralelizar
* Una o más porciones paralelizables

La parte secuencial puede limitar la rapidez.

Speedup (aceleración) = Tiempo de ejecución sin la mejora/Tiempo de ejecución con la mejora

Speedup global = 1 / ((1-fracción mejora) + (fracción mejora/speedup mejora))

Tiempo de ejecución con la mejora = Tiempo de ejecución original x ((1-fracción mejora) + (fracción mejora/speedup mejora))

Multiprocesadores

Un computador que te permite abrir programas en más de una [CPU](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento).

* Multiprocesador de memoria compartida

Es aquel tipo de [memoria](https://es.wikipedia.org/wiki/Memoria_(inform%C3%A1tica)) que puede ser accedida por múltiples [programas](https://es.wikipedia.org/wiki/Programa_(inform%C3%A1tica)), ya sea para comunicarse entre ellos o para evitar copias redundantes.

Son sistemas con múltiples procesadores que comparten un único espacio de direcciones de memoria, donde cualquier procesador puede acceder a los mismos datos, al igual que cualquier dispositivo de entrada/salida.

Acceso de memoria

* UMA (uniforme)

Todos los accesos de la memoria tardan el mismo tiempo

* NUMS (nouniforme)

Algunos accesos a memoria son más rápidos que otros, dependiendo de qué procesador quiere acceder a que palabra.

Clústeres

Red de computadoras independientes

Conjuntos de ordenadores unidos entre sí normalmente por una red y que se comportan como si fuesen una única computadora.

* Cada computadora tiene memoria privada y un sistema operativo.
* Los computadores están conectados vía puertos de entrada/salida – Envío recepción de mensajes entre procesadores.
* Apto para aplicaciones con tareas independientes
* Es asequible
* Problemas: Costo administrativo, menor ancho de banda de interconexión

Procesador Superescalar

Procesador capaz de ejecutar más de una instrucción por ciclo de reloj.

Todas las CPUs modernas son superescalares.

Es una arquitectura con pipeline (segmentación).

Para alcanzar buen rendimiento con este procesador, se requiere que el compilador planifique las instrucciones de forma que se eliminen dependencias y se mejore la frecuencia de ejecución.

Al emitir varias instrucciones por ciclo, el CPI (ciclo por instrucción) es menor a 1.

Multihilo en hardware

Realización de múltiples hilos de ejecución en paralelo. Permite que varios hilos compartan las unidades funcionales del procesador y se ejecuten simultáneamente.

Cada hilo tiene una copia del banco de registros, PC, etc. Hay una rápida conmutación de la ejecución entre diferentes hilos.

Dos formas de ejecución multihilo en hardware:

* Multihilo de grano fino
* Se cambia de hilo en cada instrucción, y como resultado se obtiene una ejecución entrelaza de varios hilos
* Multihilo de grano grueso
* La conmutación entre hilos se realiza solo cuando hay parada larga
* Simplifica el hardware
* Problema: Limitada capacidad de reducir pérdidas en las prestaciones debido a las paradas cortas.

Multihilo simultánea (SMT)

Las instrucciones a ejecutar se toman de varios flujos simultáneamente, de manera que, ante el bloqueo de uno de ellos, se puede continuar con la ejecución de cualquiera de los otros.

Procesadores vectoriales

Diseño de CPU capaz de ejecutar operaciones matemáticas sobre múltiples datos de forma simultánea, en contraste con los procesadores escalares, capaces de manejar sólo un dato cada vez.

Proporcionan operaciones de alto nivel que trabajan sobre vectores.

Posee un conjunto de registros vectoriales para sostener los operando y los resultados.

Recolecta los datos de memoria, los pone en orden en un conjunto de registros, los opera secuencialmente y luego escribe los resultados de regreso a memoria.

Unidades de procesamiento gráfico (GPUs)

Es un [coprocesador](https://es.wikipedia.org/wiki/Coprocesador) dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del [procesador](https://es.wikipedia.org/wiki/Microprocesador) central en aplicaciones como los videojuegos o aplicaciones 3D interactivas.

Son aceleradores que complementan la CPU, ya que no realizan todas las funciones del CPU.

CPU-GPU: Multiprocesamiento heterogéneo.

Los tipos de datos de gráficos son vértices que consta de coordenada (x, y, z, w) y píxeles, que consta de componentes de color (rojo, verde, azul, alfa).

Arquitectura GPU

El procesamiento de datos es paralelo (es aquel que se realiza al mismo tiempo que otro, siendo ejecutados ambos de modo simultáneo).

Utilizan ejecución de multihilos a gran escala.

Menos dependencia de los caches multinivel

La memoria gráfica es amplia y de alto ancho de banda

Tendencia hacia la GPU de propósito general: CPU para el código secuencial, GPU para código paralelo

Conclusión

Obtener mayor rendimiento mediante el uso de múltiples procesadores

Dificultades: El desarrollo de software paralelo y la elaboración de arquitecturas adecuadas.