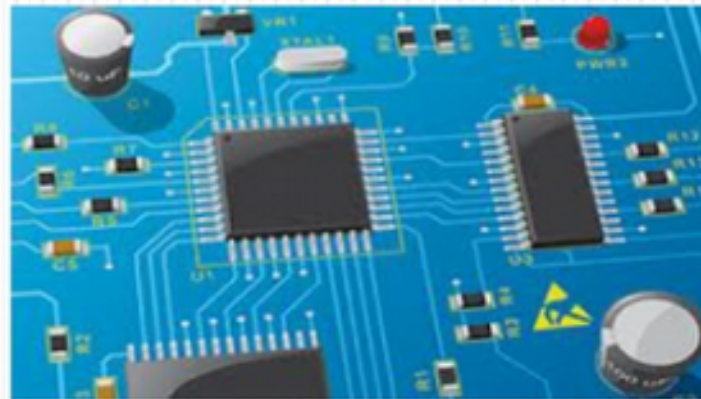




T2. Microprocesadores: 2.2. Repertorio y Ejecución de Instrucciones

FUNDAMENTOS DE ARQUITECTURA DE COMPUTADORES



Contenido del capítulo

- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Contenido del capítulo

- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Importancia de una instrucción

- Dentro del computador la actividad se controla por medio de **instrucciones**
- Realizar una tarea → almacenar en memoria un programa (conjunto de instrucciones que operan sobre unos datos).
- Las instrucciones se traen de M.P. al procesador, donde se ejecutan.
- Como operandos se utilizan datos almacenados en M.P. o en registros.

Juego de Instrucciones

- Una **instrucción** es un patrón binario diseñado dentro de un microprocesador para realizar una función específica.
- El grupo completo de instrucciones, llamado **juego de instrucciones**, determina qué funciones puede realizar el microprocesador.
- Estas instrucciones se pueden clasificar en las siguientes cinco categorías funcionales:
 1. Transferencia de datos
 2. Realización de operaciones aritméticas
 3. Realización de operaciones lógicas
 4. Pruebas de condiciones y alteración de la secuencia del programa
 5. Operaciones de control

Juego de Instrucciones (2)

- El repertorio o juego de instrucciones debe ser:
 - Completo: debe permitir resolver cualquier problema.
 - Eficaz: los programas deben ser ejecutados en un tiempo razonable.
- Está relacionado con:
 - El número de registros disponibles.
 - El tamaño de los datos.
 - Los modos de direccionamiento (maneras de acceder a los datos).
- Clasificación de los computadores según su repertorio:
 - RISC: Reduced Instruction Set Computer (ej.: MIPS, PowerPC).
 - CISC: Complex Instruction Set Computer (x86).
 - VLIW: Very Long Instruction Word (Itanium).

Juego de Instrucciones (3)

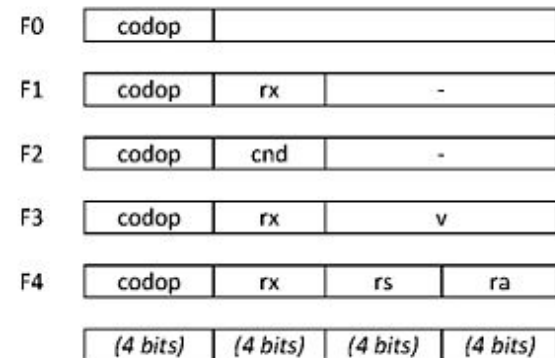
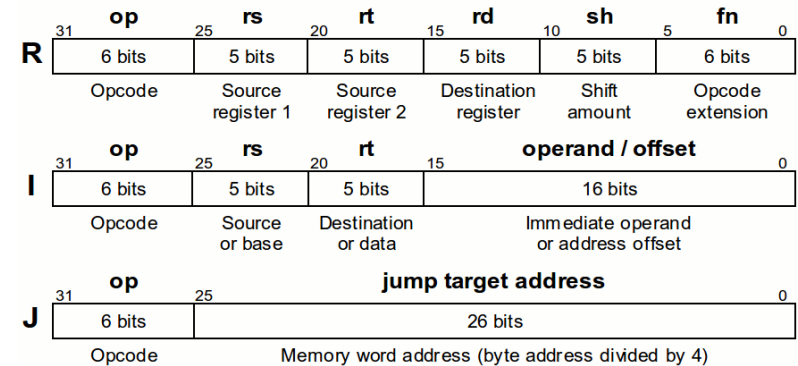
- El repertorio de instrucciones debe especificar:
 - Formato de las instrucciones: tamaño fijo, variable o híbrido.
 - Localización de operandos y resultado, junto con modos de direccionamiento.
 - Tipos de datos y tamaños: representación de información.
 - Operaciones soportadas: lógicas, aritméticas, etc.
 - Mecanismos de bifurcación: instrucciones de salto/subrutinas.
- Propiedades de las instrucciones:
 - Realizan una función única y sencilla.
 - Emplean un número fijo de operandos.
 - Su codificación binaria es bastante sistemática.
 - Contienen toda la información necesaria para su ejecución.

Juego de Instrucciones (4)

- Información contenida

formato instrucción:

- Código de operación
- Dirección operandos:
 - Valor inmediato
 - Registro(s)
 - Memoria
- Dirección resultado

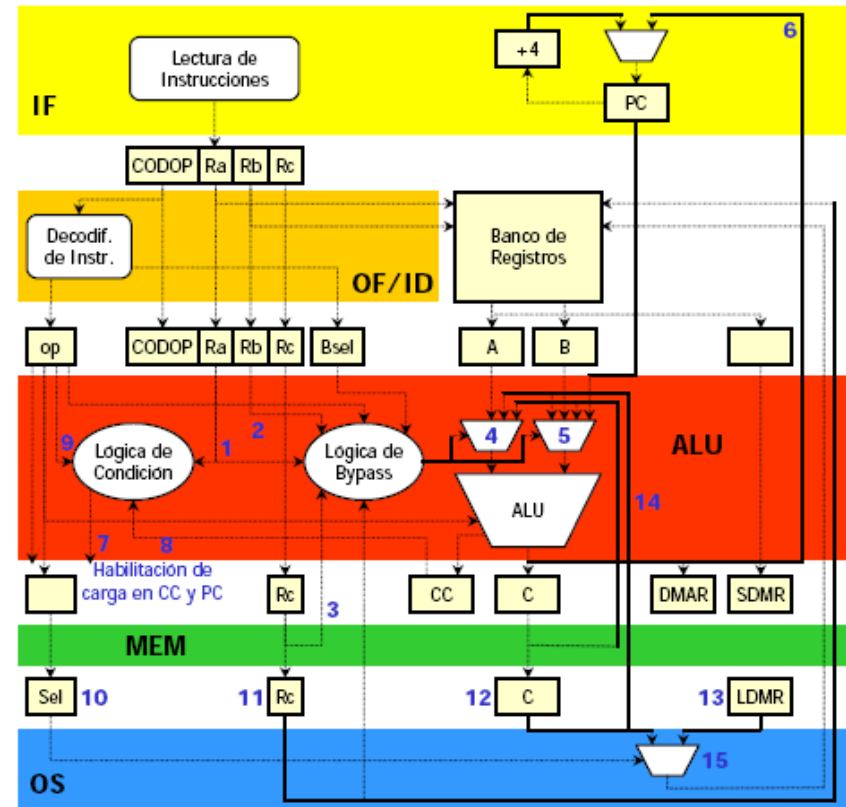


Contenido del capítulo

- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Fases de Ejecución de una Instr.

- Esquema moderno de cinco fases
- Mucho más complejo en cualquier procesador actual
- Podemos ejecutar varias instrucciones a la vez:
 - Concepto de paralelización (Tema 2.2)
 - Procesadores segmentados y superescalares: Arquit. de Computadores (2º curso)



Fases de Ejecución de una Instr. (2)

- **Captación** de instrucción (*instruction fetch* \sim *IF*)
 - La UC envía a MP la dirección de la instrucción a ejecutar, almacenada en el contador de programa PC.
 - Activa las señales de control necesarias para que la MP le entregue dicha instrucción.
 - El PC se incrementa.
- **Decodificación** instrucción (*instruction decode* \sim *ID*) y captura de operandos (*operand fetch* \sim *OF*)
- **Ejecución** de la instrucción (*execution* \sim *ALU*): la UC genera señales necesarias para que la ALU efectúe una operación.
- **Memoria** (*memory* \sim *MEM*): carga de datos desde MP.
- **Almacenamiento** del resultado en registro o MP (*write back* \sim *OS*)

Fases de Ejecución de una Instr. (3)

- Cada una de estas etapas, comprende a su vez un número indeterminado de micro-operaciones.
- Como se ha comentado previamente, la cantidad de micro-operaciones a procesar dependerá en gran medida de la “complejidad” de la instrucción.
- ¡OJO! No todas las etapas implican un mismo número de micro-operaciones.

Fases de Ejecución de una Instr. (4)



- **Lectura de una instrucción (IF + ID)**
 - Copia el contenido del PC en el MAR y comienza la lectura de la memoria.
 - Incrementa el PC.
 - Copia en MDR el dato que hay en la dirección de memoria de MAR
 - Copia en el IR la instrucción que está ahora en el MDR.
 - Decodifica el IR (es decir, examina la instrucción para determinar qué instrucción es).
- **Ejecución de la instrucción (ALU + MEM + OS).**
 - Interpretar instrucción.
 - Captar datos.
 - Procesar datos
 - Almacenar datos.
- Comprobación de la existencia de **interrupción** (IRQ ~ Interruption ReQuest).
- Repite desde el paso 1.

Fases de Ejecución de una Instr. (5)

- Suma registro – registro
- Se suma el registro R4 y R7 y se almacena en R4
- Formato de la instrucción:



- Fases de la instrucción
 1. MAR = PC
 2. MDR = MP(MAR)
 3. IR = MDR
 4. Decodificación de la instrucción leída
 5. ACC = R4 + R7
 6. PSW ← Bits de estado, si procede
 7. R4 = ACC
- Diagram illustrating the phases of instruction execution:
- Captación (IF)** (Instruction Fetch) covers steps 1, 2, and 3.
 - Decodificación (ID)** (Instruction Decode) covers step 4.
 - Ejecución (EX)** (Instruction Execute) covers steps 5 and 6.
 - Almacenamiento (WB)** (Instruction Write Back) covers step 7.

Contenido del capítulo

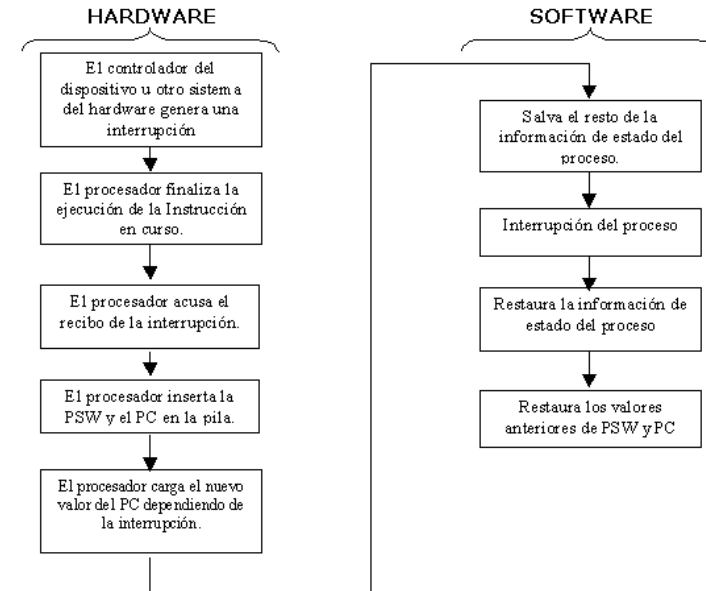
- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Tratamiento de Interrupciones

- Una interrupción es una señal recibida por la CPU, avisando que debe “pausar” el curso de ejecución actual y ejecutar algún código específico para tratar dicha situación.
- Este código se denomina “subrutina de servicio de interrupción”, generalmente perteneciente al sistema operativo, o a la BIOS.
- Finalizada dicha subrutina, se reanuda la ejecución del programa.
- Tipos de interrupciones:
 - Interrupción Hardware: comunicación con un periférico, o un problema “grave” con el equipo. Por tanto son **asíncronas**
 - Trap: operaciones no permitidas. Ej: división por 0, desbordamiento, acceso a posiciones de memoria no permitidas, etc.
 - Interrupción Software: se generan a través del propio programa. Se usan para solicitar alguna acción al sistema operativo.

Tratamiento de Interrupciones (2)

- IRQ, HW / SW. procedimiento
 - Guardar contexto: estado de los registros.
 - Almacenar PSW.
 - Rutina de Tratamiento de Interrupción.
 - Volver al estado anterior
- Trap. Procedimiento: tabla de saltos.



Dirección	Contenido	Administrador
60	JMP 2000	Instrucción inválida
64	JMP 3000	Overflow
68	JMP 3600	Underflow
72	JMP 5224	División 0
76	JMP 4180	Disco
78	JMP 5364	Teclado

Contenido del capítulo

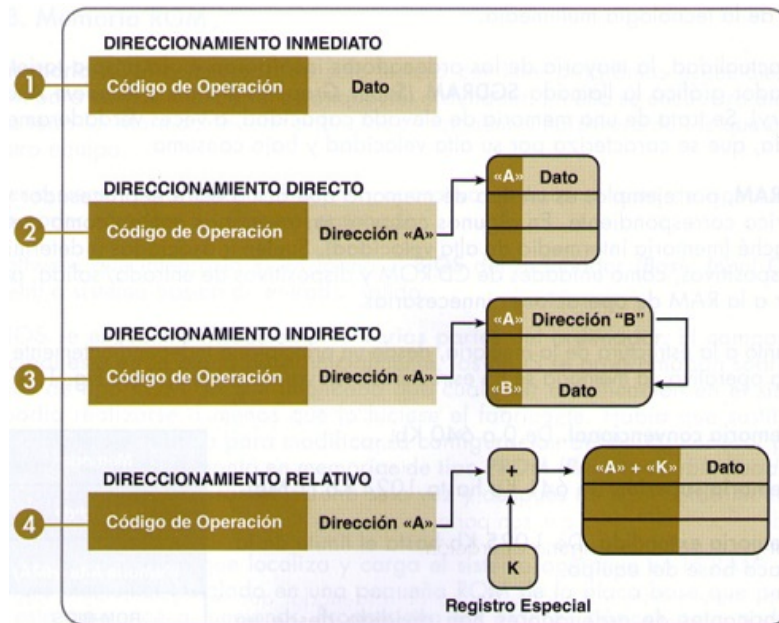
- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Modos de direccionamiento

- Número de operandos en las instrucciones:
 - (0) Sin operando explícito: Control del procesador, cambio de un indicador de estado, ...
 - (1) Con un operando explícito: Operaciones con la pila, desplazamientos, rotaciones, incremento, ...
 - (2) Con dos operandos explícitos: Operaciones entre registros y entre registro y memoria.
- **¡Nunca se realizan operaciones entre memoria y memoria!**
 - Uno de los operandos debe residir obligatoriamente en los registros de la CPU.

Modos de direccionamiento: Tipos

Modo	Ejemplo	Resultado
Inmediato	ADD R1, #5	$R1 \leftarrow R1 + 5$
Registro	ADD R1, R2	$R1 \leftarrow R1 + R2$
Directo	ADD R1, 100	$R1 \leftarrow R1 + M[100]$
Indirecto	ADD R1, (R2)	$R1 \leftarrow R1 + M[R2]$
Desplazamiento / Relativo	ADD R1, 8(R2)	$R1 \leftarrow R1 + M[R2+8]$



Contenido del capítulo

- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Bibliografía

- Patterson y Hennessy: Estructura y Diseño de Computadores: Capítulos 2 y 4.
- Prieto, Lloris, Torres: Introducción a la Informática: Capítulos 5, 6 y 7.
- Murdocca y Heuring: Principios de Arquitectura de Computadoras: Capítulos 4 y 6.

Contenido del capítulo

- Repertorio de instrucciones y ejecución
 - Importancia de una instrucción y juego de instrucciones
 - Fases de ejecución de una instrucción
 - Tratamiento de interrupciones
 - Modos de direccionamiento
- Bibliografía
- Actividades

Actividades

- La CPU en un computador de palabra de 16 bits dispone de los siguientes elementos: Registro de dirección de memoria (MAR), registro de memoria (MDR), Contador de programa (PC), puntero de pila (SP), Registro de instrucción (IR), conjunto de registros (R0 a R7), y Registro auxiliar (Ra). Suponga que:
 - La instrucción maquina de resta SUBX R (código 35C0) se realiza entre R0 (que actúa de acumulador) y un dato de memoria cuya dirección se encuentra en R7
 - Internamente las operaciones de la ALU se realizan con los registros R0 y Ra.
 - Existe el siguiente contenido inicial:
 - Instrucción: $M(370A) = 35C0$
 - $M(48A0) = B732$
 - $R0 = 0037$
 - $R7 = 48A0$
- Realice una tabla donde se indiquen las distintas microoperaciones que se realizan al ejecutar la instrucción de la posición 370A de memoria y los contenidos (cambios) en los registros correspondientes.