

Conjunto de instrucciones

Departamento de Arquitectura¹

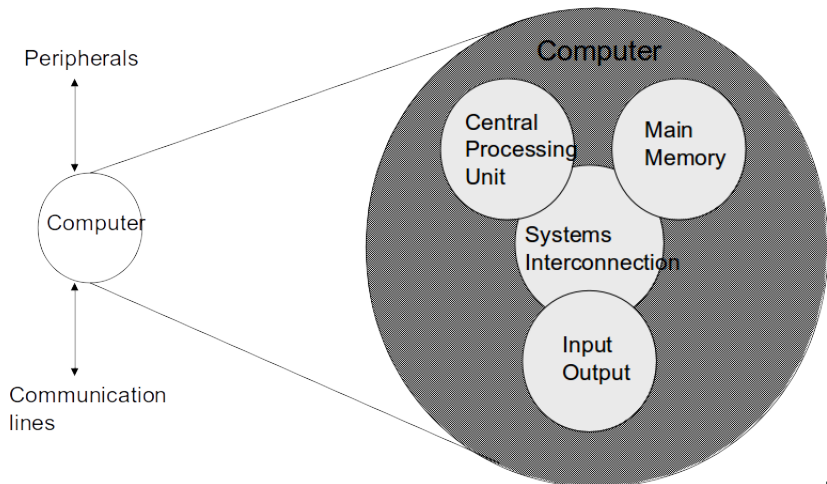
¹Instituto de Computación
Facultad de Ingeniería
Universidad de la República

Arquitectura de Computadoras, 2024

Contextualización

- Arquitectura Von Neumann.
- Assembler 8086.
- La ISA llena la brecha semántica entre usuario y máquina.
- La ISA sirve como punto de partida para el diseño de una nueva máquina o la modificación de una existente.
- Veremos las principales características de una IS y como diseñarlo.

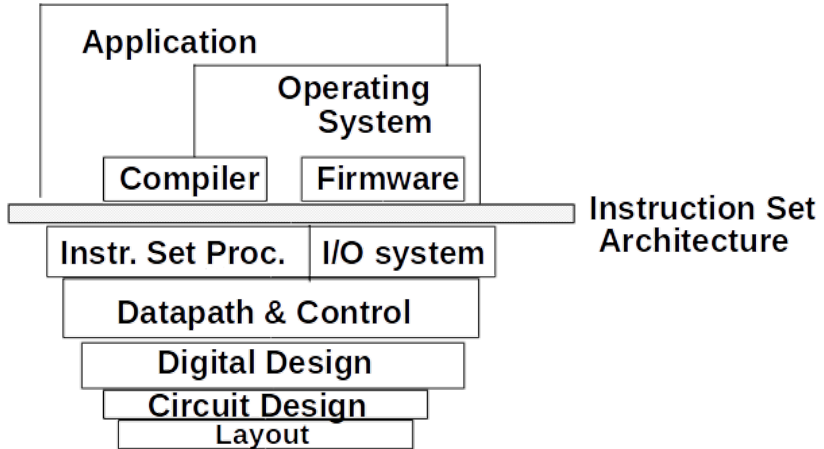
Estructura del sistema computador



Contenido

- 1 Introducción
- 2 Conjunto de instrucciones
 - Instrucciones
 - Tipos de operandos
 - Modos de direccionamiento
 - Formato de instrucción

Hardware y software



Lenguajes

- Lenguaje de alto nivel.
- Lenguaje ensamblador.
- Lenguaje máquina.

Ensamblador

- Es difícil para el programador trabajar con representaciones binarias.
- Se utilizan representaciones simbólicas de las instrucciones máquina.
 - Mnemónicos para los códigos de operación.
 - Nombrar los operandos.
 - Etiquetas para los cambios del flujo.

Abstracción

Programa en C

```
swap(int v[], int k)
{int temp;
    temp=v[k];
    v[k]=v[k+1];
    v[k+1]=temp;
}
```

Compilador de C (MIPS)

Programa en Assembler

```
swap:
mulh $t1, $a1,4
add $t1, $a0,$t1
lw $t0, 0($t1)
lw $t2, 4($t1)
sw $t2, 0($t1)
sw $t0, 4($t1)
jr $ra
```

Ensamblador (MIPS)

Programa en Lenguaje de máquina

```
000000001010000100000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```


Conjunto de instrucciones

- ISA.
- Computadoras RISC o CISC.
- Una instrucción contiene toda la información necesaria para que el procesador la ejecute.
- El conjunto de instrucciones debe ser suficiente para expresar las instrucciones de alto nivel.

Aspectos de diseño del IS

- Dónde se almacenan los operandos (stack, registro, memoria, AC).
- Cuántos operandos explícitos incluye
- Cómo se especifica la posición de los operandos (modo direccionamiento)
- Qué tipo de operandos se permite (byte, int, float, string, array, ...)
- Qué operaciones se soportan (add, cmp, call, ...)

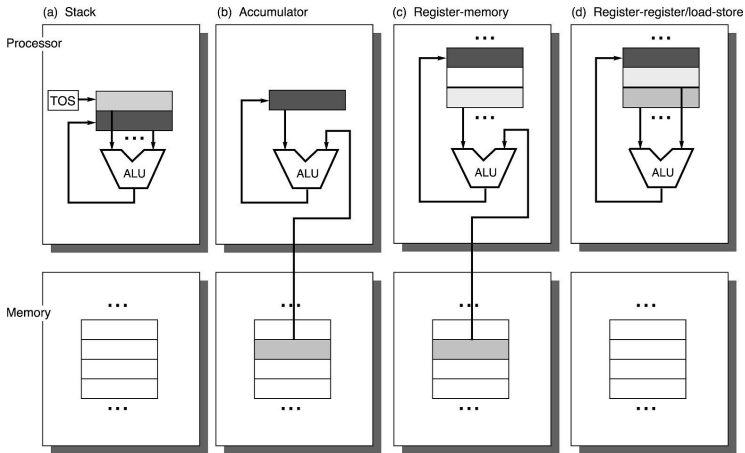
Clasificación de los ISA

Modo de acceso a los datos

- Arquitectura de Pila.
- Arquitecturas de Acumulador.
- Arquitecturas registro - memoria.
- Arquitecturas registro - registro (Load - Store).

Modo de acceso a los datos

Organización



Modo de acceso a los datos

Ejemplo

$$A := B + C$$

stack	acumulador	registro-memoria	registro-registro
push C	load B	load r1, B	load r1, B
push B	add C	add r1, C	load r2, C
add	store A	store A, r1	add r3, r1, r2
pop A			store A, r3

Cuadro: ejemplos de ensamblado

Clasificación de los ISA

Stack

- Buena densidad de código.
- Bajos requerimientos de hardware
- Fácil de implementar compiladores simples.
- Complica la implementación de optimizaciones de compilación.
- Poca capacidad para paralelismo y pipeline.
- El stack se transforma en un cuello de botella para el sistema.

- Fácil de diseñar y entender.
- Bajos requerimientos de hardware
- Poca capacidad para paralelismo y pipeline.
- El acumulador se transforma en un cuello de botella para el sistema.
- Alto tráfico entre memoria y el procesador.

Clasificación de los ISA

Memoria - Registro

- Buena densidad de código.
- Fácil de codificar las instrucciones
- Los operandos no son equivalentes (baja ortogonalidad).
- Número variable de ciclos por instrucción.

Clasificación de los ISA

Registro-Registro

- Largo de instrucciones fijo.
- La cantidad de ciclos por instrucción es similar para cada instrucción.
- Simplifica el diseño del pipeline.
- Requieren de buenos compiladores.
- Generan mayor cantidad de instrucciones.

Elementos de una instrucción

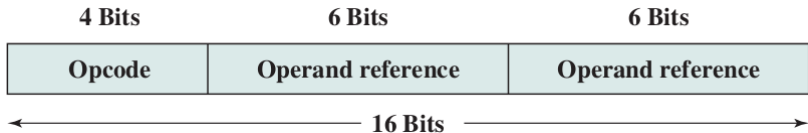
- Código de operación.
- Referencia a operandos origen.
- Referencia a operandos destino.
- Referencia a la siguiente instrucción.

Instrucción

- Componente básico de un lenguaje máquina.
- Macrooperación compuesta de un secuencia de microoperaciones.
- Representadas en código binario según el Formato de Instrucción.
- Ejecutadas por la UC.

Representación de las instrucciones

- Una instrucción es una tira de bits.
- Se estructura en campos.



Elementos de una instrucción

Ejemplo

- ADD AX, BX
- JMP fin

Dónde se ubican los operandos

- Memoria.
- Registros del procesador.
- Inmediato.
- Dispositivo de E/S.

Tipos de instrucciones

Categorización

- Procesamiento de datos
- Almacenamiento de datos
- Movimientos de datos (operaciones de IO)
- Control

Cantidad de operandos

<u>Instruction</u>	<u>Comment</u>
SUB Y, A, B	$Y \leftarrow A - B$
MPY T, D, E	$T \leftarrow D \times E$
ADD T, T, C	$T \leftarrow T + C$
DIV Y, Y, T	$Y \leftarrow Y \div T$

(a) Three-address instructions

<u>Instruction</u>	<u>Comment</u>
MOVE Y, A	$Y \leftarrow A$
SUB Y, B	$Y \leftarrow Y - B$
MOVE T, D	$T \leftarrow D$
MPY T, E	$T \leftarrow T \times E$
ADD T, C	$T \leftarrow T + C$
DIV Y, T	$Y \leftarrow Y \div T$

(b) Two-address instructions

$$Y = \frac{A - B}{C + (D \times E)}$$

<u>Instruction</u>	<u>Comment</u>
LOAD D	$AC \leftarrow D$
MPY E	$AC \leftarrow AC \times E$
ADD C	$AC \leftarrow AC + C$
STOR Y	$Y \leftarrow AC$
LOAD A	$AC \leftarrow A$
SUB B	$AC \leftarrow AC - B$
DIV Y	$AC \leftarrow AC \div Y$
STOR Y	$Y \leftarrow AC$

(c) One-address instructions

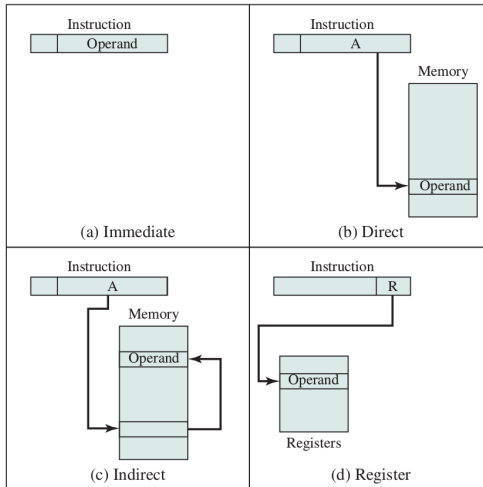
Categoría de datos

- Direcciones
- Números
- Caracteres
- Datos lógicos

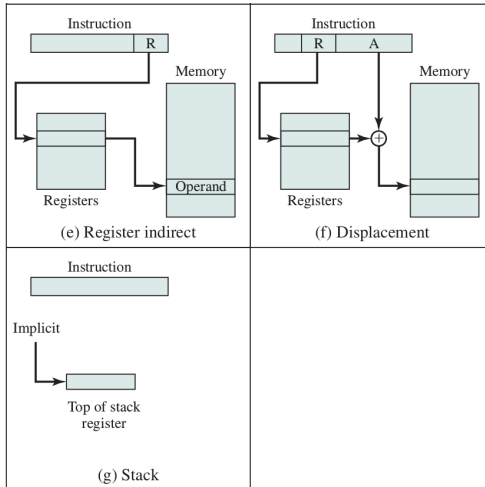
Tipos de datos numéricos

- Enteros
- Punto flotante
- Decimal

Modos de direccionamiento



Modos de direccionamiento



Formato de instrucción

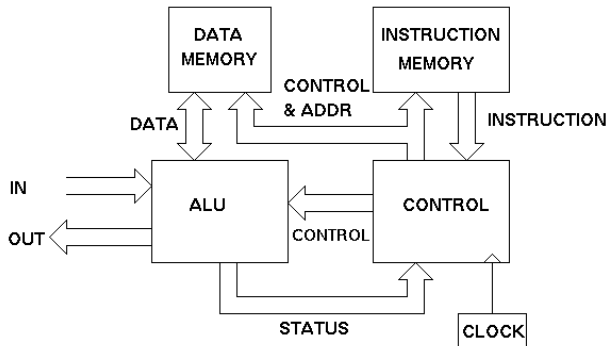
- Define la estructura de los bits en la instrucción.
- Campos (opcode, operandos y modos de direccionamiento).
- Complejo de diseñar.

Largo de la instrucción

- Fijo (Sparc 4 bytes), variable (VAX, 1-53 bytes) o híbridos (80X86, 7 formatos diferentes de 1-5 bytes).
- Von Neuman o Harvard.

Arquitectura Harvard

HARVARD ARCHITECTURE MICROPROCESSOR



Bits por campo

- Largo fijo
 - dilema entre cantidad de bits para opcode y bits para operandos.
 - campos de largo variable.
- Campos para operandos:
 - modo de direccionamiento.
 - número de operandos.
 - número de registros.
 - número de bancos.
 - rango de direcciones.
 - granularidad de la dirección (byte, word, ...).

Codificación - SPARC



op	Format
00	SETHI/Branch
01	CALL
10	Arithmetic
11	Memory

op2	Inst.
010	branch
100	sethi

op3 (op=10)	
010000	addec
010001	andec
010010	orcc
010110	orncec
100110	srl
111000	jmp1

op3 (op=11)	
000000	ld
000100	st

cond	branch
0001	be
0101	bcs
0110	bneg
0111	bvs
1000	ba

Interfaz de Entrada/Salida con el procesador

- Direccionamiento
 - E/S aislada
 - E/S mapeada en Memoria
- Control de E/S
 - Polling
 - Interrupciones
 - DMA

Direccionamiento

- E/S aislada
 - Espacios de direcciones separados
 - Se necesitan líneas de selección entre E/S y memoria
 - Instrucciones específicas para E/S
- E/S mapeada en memoria
 - Dispositivos y memoria comparten espacio de direcciones
 - Accesos a E/S se ven como simples accesos a memoria. El HW debe resolver la diferencia
 - No hay instrucciones especiales para E/S

Lecturas Recomendadas



[William Stallings.](#)

Computer Organization and Architecture: Designing for Performance.
Capítulos 12 y 13. 2010.



[OpenFing.](#)

Video - Teórico [Máquina Lógica General / Arquitectura Von Neumann].
Unidad 10. 2024.

Preguntas