

Arquitectura de Computadores I

Luis Alberto Chavarría Zamora

ITCR

lachavarria@tec.ac.cr

8 de agosto de 2023

Contenido

- ① Lección Anterior
- ② Formas de Organización / Microarquitectura
 - Arquitectura Von Neumann
 - Arquitectura Harvard
 - Arquitectura Harvard Modificada
 - Comparación
- ③ Clasificación de los ISA
 - Tipo de Operando
 - Load/Store
 - Register/Mem
 - Complejidad de las instrucciones
 - CISC vs RISC
 - Otros ISAs
- ④ Referencias

Lección Anterior

¿Qué vimos?

- 1 ¿Qué es una arquitectura de un computador?
- 2 ¿Qué es microarquitectura?
- 3 ¿Qué consideraciones tiene la ley de Amdahl respecto al tipo de procesador y al paralelismo?
- 4 ¿Cuáles son las principales clasificaciones según Flynn?
- 5 ¿Qué diferencia existe entre una arquitectura simétrica y una no simétrica?
- 6 ¿Qué es benchmarking?
- 7 ¿Qué es mantenibilidad, reparabilidad, disponibilidad?

Lección Anterior

Formas de Organización / Microarquitectura

Arquitectura Von
Neumann

Arquitectura Harvard

Arquitectura Harvard
Modificada

Comparación

Clasificación de los ISA

Tipo de Operando

Load/Store

Register/Mem

Complejidad de las
instrucciones

CISC vs RISC

Otros ISAs

Referencias

Lección Anterior

¿Qué vimos?



Formas de Organización/Microarquitectura

Tradicionalmente se habla de dos tipos:

- 1 Arquitectura Von Neumann.
- 2 Arquitectura Harvard.

Formas de Organización/Microarquitectura

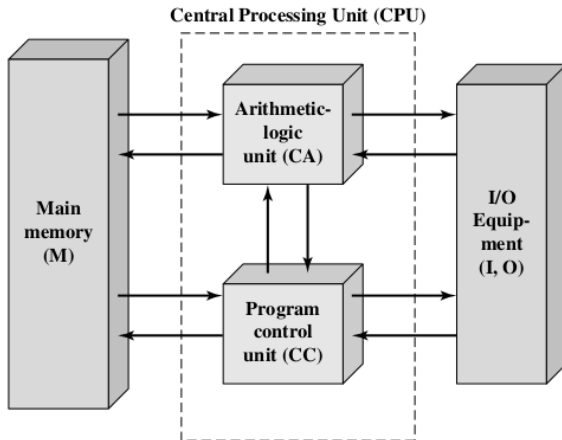
Arquitectura Von Neumann

- Jon von Neumann (nombre real: Neumann János Lajos) (1903-1957).
- Matemático, ingeniero químico.
- Contribuciones en el campo de la mecánica cuántica, física, economía, militar, informática.
- Murió de cáncer de huesos a los 53 años.



Formas de Organización/Microarquitectura

Arquitectura Von Neumann



Formas de Organización/Microarquitectura

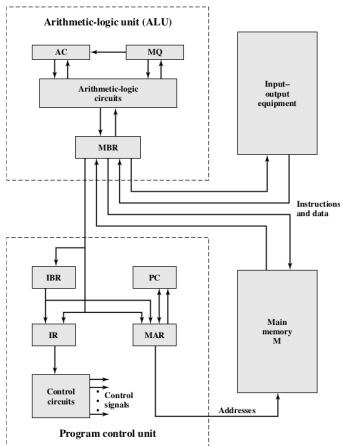
Arquitectura Von Neumann

En la versión original se listaron las siguientes partes:

- *Central Arithmetic (CA)*: Unidad encargada de llevar a cabo las operaciones aritméticas de suma, resta, multiplicación y división.
- *Central Control (CC)*: Lógica de control del computador, encargado de llevar la secuencia correcta del programa.
- *Memoria (M)*: Almacena largas cantidades de operaciones (programa). Se ejecuta secuencialmente.
- *I/O Equipment (I,O)*: Periféricos del sistema.

Formas de Organización / Microarquitectura

Arquitectura Von Neumann



Formas de Organización/Microarquitectura

Arquitectura Von Neumann

- **Unidad Aritmética y Lógica (ALU):**
 - **AC** o acumulador: Guarda los resultados de los cálculos hecho por la ALU.
 - **MQ** o multiplicador cociente: Se usa para guardar los bits menos significativos producto de una multiplicación.
 - **Memory Buffer Register (MBR):** Almacena las instrucciones obtenidas de la memoria o cualquier dato que se transfiera y almacene en la memoria. También se conoce como MDR (*Memory Data Register*).
- **Unidad que Control:**
 - **Program Counter (PC):** Realiza un seguimiento de la ubicación de la memoria de las siguientes instrucciones a tratar. La PC luego pasa esta siguiente dirección al *Memory Buffer Register (MBR)*.
 - **Instruction Register (IR):** Mantiene la instrucción que está siendo ejecutada.
 - **Instruction Buffer Register (IBR):** La instrucción que no se ejecutará inmediatamente se coloca en el registro de búfer de instrucciones IBR.
 - **Memory Address Register (MAR):** Almacena las ubicaciones de memoria de las instrucciones que deben recuperarse de la memoria o almacenarse en la memoria.

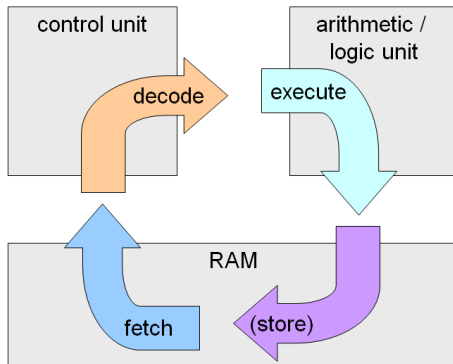
Formas de Organización/Microarquitectura

Arquitectura Von Neumann: Características

- La información se representa por medio de direcciones.
- **Memoria unificada**, una única memoria para datos y programa.
- Las instrucciones almacenadas y ejecutadas secuencialmente: Program counter debe actualizarse ($PC=PC+1$) para obtener siguiente instrucción.
- Ciclo de Fetch: Búsqueda, Decodificación, Ejecución, Almacenado.
- Cuenta con un ISA de 21 instrucciones.

Formas de Organización / Microarquitectura

Arquitectura Von Neumann



Simulador de Von Neumann

Formas de Organización/Microarquitectura

Arquitectura Harvard

- Fue presentado a IBM por Howard Aiken en 1937.
- Fue construída en 1944 por personal de la Universidad de Harvard.
- Era una computadora electromecánica con propósito militar para la Segunda Guerra Mundial.



Formas de Organización/Microarquitectura

Arquitectura Harvard

- La evolución en los computadores y la necesidad de rapidez y paralelismo generaron cambios en la organización de los computadores.
- La memoria unificada de acceso secuencial representa un obstáculo (cuello de botella o *Von Neumann Bottleneck*).

Formas de Organización/Microarquitectura

Arquitectura Harvard

- La evolución en los computadores y la necesidad de rapidez y paralelismo generaron cambios en la organización de los computadores.
- La memoria unificada de acceso secuencial representa un obstáculo (cuello de botella o *Von Neumann Bottleneck*).
- Solución: Separar la memoria - Memoria para instrucciones (I) y memoria para datos (D).
- Paralelismo porque puede haber acceso simultáneo a memoria para instrucciones y memoria para datos.

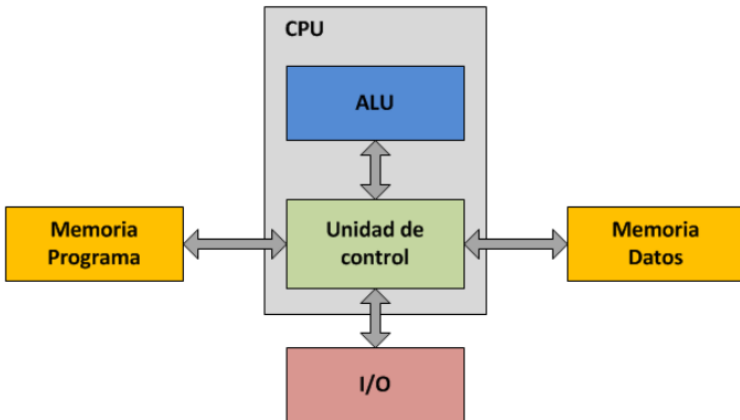
Formas de Organización/Microarquitectura

Arquitectura Harvard: Características

- Memoria de datos y memoria de programa están **físicamente** separadas.
- Acceso a memoria de datos e instrucciones puede ser simultáneo.
- Ventaja: mayor rendimiento (paralelismo).
- La mayoría de DSP poseen una arquitectura tipo Harvard pues necesitan buscar datos y operaciones al mismo tiempo.

Formas de Organización/Microarquitectura

Arquitectura Harvard: Características



Formas de Organización/Microarquitectura

Arquitectura Harvard Modificada

- Desventajas de la arquitectura de Harvard:
 - El espacio de direccionamiento separado implica 2 memorias físicas diferentes: mayor espacio, consumo de potencia.
 - Rutas diferentes (mayor ancho de banda) generan mayor consumo de potencia dinámica.

Formas de Organización/Microarquitectura

Arquitectura Harvard Modificada

- Desventajas de la arquitectura de Harvard:
 - El espacio de direccionamiento separado implica 2 memorias físicas diferentes: mayor espacio, consumo de potencia.
 - Rutas diferentes (mayor ancho de banda) generan mayor consumo de potencia dinámica.
- Arquitectura de Harvard Modificada:
 - Disminuye el impacto de la separación de memoria.
 - Rutas separadas para instrucciones y datos, con único espacio de direccionamiento.
 - Provee instrucciones para acceder a los contenidos de la memoria de instrucciones como si fueran datos.
 - Una única memoria principal.

Formas de Organización/Microarquitectura

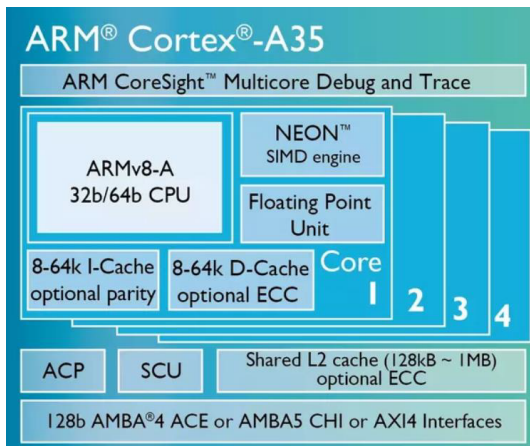
Arquitectura Harvard Modificada: Implementación

- Utiliza dos memorias caché de CPU, para la separación de datos e instrucciones.
- Una única memoria principal.
- Desde el punto de vista macro se comporta como una arquitectura Von Neumann, pero internamente separa instrucciones y datos.
- ¿Dónde está lo complicado?

La mayoría de las arquitecturas modernas Harvard son en realidad Harvard Modificada.

Formas de Organización/Microarquitectura

Arquitectura Harvard Modificada: Implementación



Formas de Organización/Microarquitectura

Arquitectura Harvard Modificada: Implementación

- Arquitectura Von Neumann: Un único espacio de direccionamiento, y única ruta de acceso al CPU.
- Arquitectura Harvard: Memoria de datos y memoria de instrucciones tienen rutas de hardware diferentes hacia el CPU, además de espacios de direccionamiento separados.
- Arquitectura Harvard Modificada: rutas de hardware diferentes para el CPU Cache, y un espacio de direccionamiento único.

Clasificación de los ISA

Tipo de operando

Tipos de operando que existen:

- Load / Store.
- Register / Mem.

Clasificación de los ISA

Tipo de operando: Load/Store

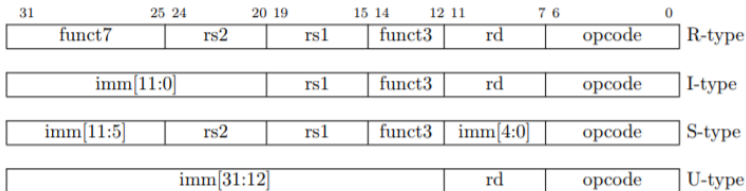
Load/Store → Divide las operaciones en dos categorías:

- Accesos a memoria (instrucciones: Load/Store en memorias y registros).
- Operaciones con ALU (solo entre registros).

Ejemplos: ARM, RISC-V, MIPS.

Clasificación de los ISA

Tipo de operando: Load/Store



Clasificación de los ISA

Tipo de operando: Register/Mem

Operaciones pueden ser entre registros y entre espacios de memoria.

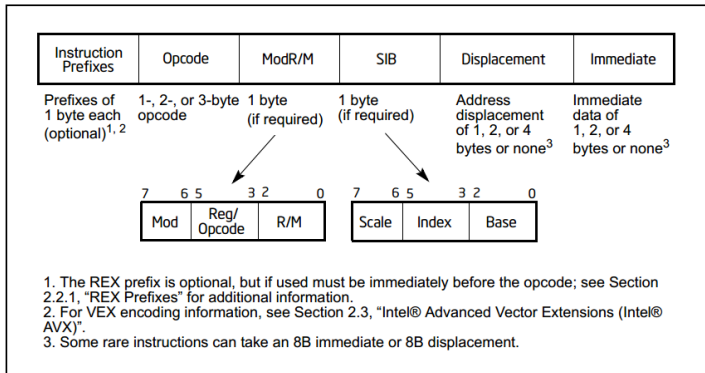


Figure 2-1. Intel 64 and IA-32 Architectures Instruction Format

Clasificación de los ISA

Complejidad de instrucciones: CISC

Complex Instruction Set Computer:

- Enfoque inicial de arquitectura.
- El ISA contiene gran variedad de instrucciones: instrucciones poderosas y complicadas.
- Facilidad de programación.
- El compilador no realiza traducciones complejas: instrucciones son similares a lenguajes de alto nivel.

Ejemplo: x86.

Clasificación de los ISA

Complejidad de instrucciones: CISC

Complex Instruction Set Computer:

Características típicas del set:

- Múltiples modos de direccionamiento (forma de acceder a datos en memoria).
- Formato de instrucciones variable.
- Duración de instrucciones variable.
- Bajo número de registros de propósito general. x86: RAX, RBX, RCX, RDX.
- Las instrucciones son capaces de ejecutar tareas complejas.
- Decodificación de instrucciones implica mayor lógica de hardware.

Clasificación de los ISA

Complejidad de instrucciones: CISC

- Ventajas:
 - Facilidad de programación: Tareas complejas se realizan en menos tiempo.
 - Múltiples modos de direccionamiento simplifican las tareas.
 - Tamaño de código pequeño.
- Desventajas:
 - Instrucciones de tamaño variable: diferente tiempo de búsqueda y ejecución hacen muy complicado tener sistemas determinísticos. Hardware es complicado (área, dinero).
 - Muchas de las instrucciones especializadas no son utilizadas con frecuencia: El 98 % de las instrucciones en un programa típico corresponden al 20 % de las instrucciones del set.

Clasificación de los ISA

Complejidad de instrucciones: CISC

Ejemplo:

5.10.2 Floating-Point Dot Product Instructions

DPPD Perform double-precision dot product for up to 2 elements and broadcast.

DPPS Perform single-precision dot products for up to 4 elements and broadcast.

DPPD — Dot Product of Packed Double Precision Floating-Point Values

Opcode/ Instruction	Op/ En	64/32-bit Mode	CPUID Feature Flag	Description
66 0F 3A 41 /r lb DPPD <i>xmm1</i> , <i>xmm2</i> /m128, <i>imm8</i>	RMI	V/V	SSE4_1	Selectively multiply packed DP floating-point values from <i>xmm1</i> with packed DP floating-point values from <i>xmm2</i> , add and selectively store the packed DP floating-point values to <i>xmm1</i> .
VEX.128.66.0F3A.WIG 41 /r lb VDPPD <i>xmm1</i> , <i>xmm2</i> , <i>xmm3</i> /m128, <i>imm8</i>	RVMI	V/V	AVX	Selectively multiply packed DP floating-point values from <i>xmm2</i> with packed DP floating-point values from <i>xmm3</i> , add and selectively store the packed DP floating-point values to <i>xmm1</i> .

Clasificación de los ISA

Complejidad de instrucciones: CISC

Ejemplo:

Operation

DP_primitive (SRC1, SRC2)

```
IF (Imm8[4] = 1)
    THEN Temp1[63:0] ← DEST[63:0] * SRC[63:0]; // update SIMD exception flags
    ELSE Temp1[63:0] ← +0.0; FI;
IF (Imm8[5] = 1)
    THEN Temp1[127:64] ← DEST[127:64] * SRC[127:64]; // update SIMD exception flags
    ELSE Temp1[127:64] ← +0.0; FI;
/* If unmasked exception reported, execute exception handler*/

Temp2[63:0] ← Temp1[63:0] + Temp1[127:64]; // update SIMD exception flags
/* If unmasked exception reported, execute exception handler*/
```

```
IF (Imm8[0] = 1)
    THEN DEST[63:0] ← Temp2[63:0];
    ELSE DEST[63:0] ← +0.0; FI;
IF (Imm8[1] = 1)
    THEN DEST[127:64] ← Temp2[63:0];
    ELSE DEST[127:64] ← +0.0; FI;
```

DPPD (128-bit Legacy SSE version)

```
DEST[127:0] ← DP_Primitive(SRC1[127:0], SRC2[127:0]);
DEST[MAXVL-1:128] (Unmodified)
```

VDPPD (VEX.128 encoded version)

```
DEST[127:0] ← DP_Primitive(SRC1[127:0], SRC2[127:0]);
DEST[MAXVL-1:128] ← 0
```

Flags Affected

None

Clasificación de los ISA

Complejidad de instrucciones: RISC

Reduced Instruction Set Computer:

- Enfoque moderno: DSPs, CPUs para sistemas embebidos.
- El set está compuesto por pocas instrucciones con funcionalidad simple.
- La dificultad está en el programador (bajo nivel) o el compilador.

Ejemplos: MIPS, ARM.

Clasificación de los ISA

Complejidad de instrucciones: RISC

Reduced Instruction Set Computer:

Características típicas del set:

- Pocos modos de direccionamiento (1-4).
- Las instrucciones tienen un tamaño fijo.
- El tiempo de ejecución de cada instrucción es el mismo.
- Alto número de registros de propósito general (16, +32).
- Decodificación de instrucciones simple.

Clasificación de los ISA

Complejidad de instrucciones: RISC

- **Ventajas:**
 - Instrucciones de tiempo y tamaño fijo: simplifica hardware y brinda determinismo.
 - Mejor aprovechamiento de hardware.
 - Permite pipeline.
- **Desventajas:**
 - Tamaño de código mayor.
 - Carga pesada para el software (programa de bajo nivel o compilador).

Clasificación de los ISA

Complejidad de instrucciones: RISC

Ejemplo:

Table F1-26 Advanced SIMD dot product instructions

Mnemonic	Instruction	See
VSDOT	Signed dot product (vector form)	<i>VSDOT (vector)</i> on page F6-4475
VUDOT	Unsigned dot product (vector form)	<i>VUDOT (vector)</i> on page F6-4577
VSDOT	Signed dot product (indexed form)	<i>VSDOT (by element)</i> on page F6-4473
VUDOT	Unsigned dot product (indexed form)	<i>VUDOT (by element)</i> on page F6-4575

Clasificación de los ISA

Complejidad de instrucciones: RISC

Ejemplo:

Operation for all encodings

```
bits(64) operand1;
bits(64) operand2;
bits(64) result;
CheckAdvSIMDEnabled();
for r = 0 to regs-1
    operand1 = D[n+r];
    operand2 = D[m+r];
    result = D[d+r];
    integer element1, element2;
    for e = 0 to 1
        integer res = 0;
        for i = 0 to 3
            if signed then
                element1 = SInt(Elem[operand1, 4 * e + i, esize DIV 4]);
                element2 = SInt(Elem[operand2, 4 * e + i, esize DIV 4]);
            else
                element1 = UInt(Elem[operand1, 4 * e + i, esize DIV 4]);
                element2 = UInt(Elem[operand2, 4 * e + i, esize DIV 4]);
            res = res + element1 * element2;
        Elem[result, e, esize] = Elem[result, e, esize] + res;
    D[d+r] = result;
```

Clasificación de los ISA

CISC vs RISC: Breakout room

Discuta en grupos las siguientes preguntas:

- 1 ¿Cuál es la razón por la que CISC tiene pocos GPR?
- 2 ¿Qué efecto tiene en la potencia la implementación de ISA reducido y sencillo (RISC)?
- 3 ¿Cuál es la razón por la que CISC es más antiguo que RISC?
- 4 ¿Por qué RISC tiene menos modos de direccionamiento que CISC?

Clasificación de los ISA

CISC vs RISC

Ejemplo:

	Complex Instruction Set (CISC) Computer			Reduced Instruction Set (RISC) Computer	
Characteristic	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000
Year developed	1973	1978	1989	1987	1991
Number of instructions	208	303	235	69	94
Instruction size (bytes)	2-6	2-57	1-11	4	4
Addressing modes	4	22	11	1	1
Number of general- purpose registers	16	16	8	40-520	32
Control memory size (Kbits)	420	480	246	—	—
Cache size (KBytes)	64	64	8	32	128

Clasificación de los ISA

CISC vs RISC

Ejemplo:

Table VI. Geometric Mean CPI a cross All Benchmark Suites

ISA	ARM	x86	MIPS	ARM	ARM	x86	x86
Implementation	A8	Atom	Loongson	A9	A15	Bobcat	i7
CPI	2.5	1.9	1.5	1.6	1.2	1.3	0.7

Table VII. Instruction Size Summary

		(a) Binary Size (MB)			(b) Instruction Length (B)		
		MIPS	ARM	x86	MIPS	ARM	x86
Mobile	Minimum	–	0.02	0.02	4.0	4.0	2.4
	Average	0.55	0.95	0.87	4.0	4.0	3.3
	Maximum	–	1.30	1.42	4.0	4.0	3.7
Desktop INT	Minimum	0.61	0.53	0.65	4.0	4.0	2.7
	Average	1.58	1.47	1.46	4.0	4.0	3.1
	Maximum	4.35	3.88	4.05	4.0	4.0	3.5
Desktop FP	Minimum	0.76	0.66	0.74	4.0	4.0	2.6
	Average	1.81	1.70	1.73	4.0	4.0	3.4
	Maximum	5.21	4.75	5.24	4.0	4.0	6.4
Server	Minimum	0.16	0.12	0.18	4.0	4.0	2.5
	Average	0.51	0.39	0.59	4.0	4.0	3.2
	Maximum	0.84	0.47	1.00	4.0	4.0	3.7

Clasificación de los ISA

Otros ISAs

- ISA ortogonal: El código de operación y el operando son independientes.
- Cualquier instrucción puede usar cualquier operando.

Longitud fija vs longitud variable:

- Longitud fija: *Fetching* y *decoding* por hardware es rápido.
- Longitud variable: *Fetching* y *decoding* por hardware es lento.

Referencias



J. Hennesy y D. Patterson (2012)

Computer Architecture: A Quantitative Approach. 5th Edition.
Elsevier – Morgan Kaufmann.



J. González y R. García (2019)

Notas de clase de los profesores: Jeferson González y Ronald García.

Arquitectura de Computadores I

Luis Alberto Chavarría Zamora

ITCR

lachavarria@tec.ac.cr

8 de agosto de 2023