



DCC
DEPARTAMENTO DE CIENCIA
DE LA COMPUTACIÓN

IIC2343 - Arquitectura de Computadores

Clase 12 - Arquitectura del set de instrucciones

Instruction Set Architecture

- La arquitectura del set de instrucciones (ISA) define cómo se puede manejar la CPU a través del software.
- Es la interfaz entre el hardware y software.
- Define qué cosas puede hacer el procesador.
- Para esto es necesario conocer parte de la microarquitectura del computador:
 - Cuantos registros tiene el procesador.
 - Tipos de datos que soporta.
 - Cuales son las instrucciones disponibles...

Todo esto define la ISA.

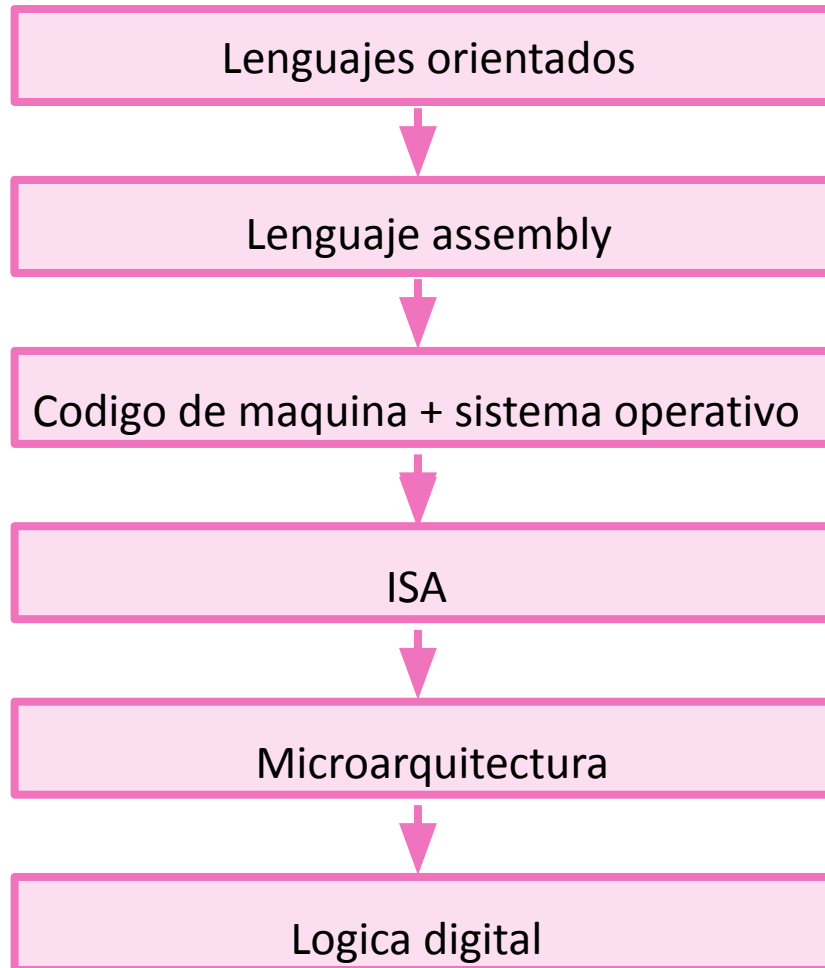
Instruction Set Architecture

- La arquitectura del set de instrucciones (ISA) define cómo se puede manejar la CPU a través del software.
- Es la interfaz entre el hardware y software.
- Define qué cosas puede hacer el procesador.
- Para esto es necesario conocer parte de la microarquitectura del computador:
 - Cuantos registros tiene el procesador.
 - Tipos de datos que soporta.
 - Cuales son las instrucciones disponibles...

Todo esto define la ISA.

**La ISA NO define la implementación en el hardware
(microarquitectura)**

Instruction Set Architecture



La ISA nos permite conectar el código escrito en lenguajes de alto nivel, con el hardware.

También que un mismo procesador pueda ejecutar programas escritos en diferentes lenguajes de programación. Solo es necesario tener el compilador adecuado que traduzca dicho lenguaje a un código de maquina soportado por la ISA.

Instruction Set Architecture

Características de la ISA de nuestro computador básico	
Registros	Tiene 2 registros; A y B
Formato de instrucciones	opcode + cero, uno o dos operandos; 15 bits
Tipos de instrucciones	Aritmeticas, lógicas y con acceso a memoria
Direccionamiento	Inmediato, directo, por registro e indirecto por registro
Tipos de datos	Números enteros con y sin signo
Flujo de control	Secuencial, condicional (jumps), call y return
Ciclos por instrucciones	Uno, dos en caso de RET, POP A y POP B

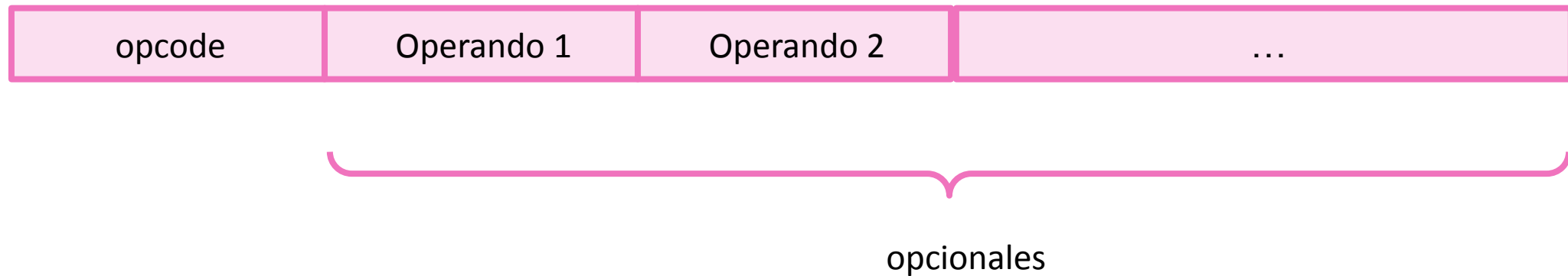
Clasificación de ISAs

- El tipo de almacenamiento interno en un procesador es la diferenciación más básica:
 - Arquitectura de **stack**: Los operandos están implícitamente en el tope del stack.
 - Arquitectura de **acumulador**: Un operando es implícitamente el acumulador (donde se almacena el resultado).
 - Arquitectura de **registros de propósito general**: Aprovecha más el uso de los registros. Puede ser tipo **registro-memoria** o **load-store**.

	registro-memoria	load-store
Definición	Cualquier instrucción puede tener acceso a la memoria	Sólo las instrucciones <i>load</i> y <i>store</i> tienen acceso a la memoria
Ejemplo para el código $c = a + b$	<pre>load r1, a add r3, r1, b store r3, c</pre>	<pre>load r1, a load r2, b add r3, r1, r2 store r3, c</pre>
Ventajas (relativas)	Permite acceso a los datos sin una instrucción <i>load</i> ; el formato de las instrucciones es fácil de codificar y produce buena densidad	La codificación de instrucciones es simple y de largo fijo; todas las instrucciones toman el mismo número de ciclos para ejecutarse
Desventajas (relativas)	Como los operandos son distintos, se desaprovecha en algunos casos los bits destinados al registro fuente. Además al tener codificado en cada instrucción el número del registro destino, el número del registro fuente y una dirección de memoria, se limita la cantidad de registros que se pueden tener.	Mayor número de instrucciones; más instrucciones y menor densidad de instrucciones produce programas más largos que puede tener efectos en la caché de instrucciones.
Ejemplos de ISAs	Intel 80x86, IBM 360/370	ARM, MIPS, RISC-V

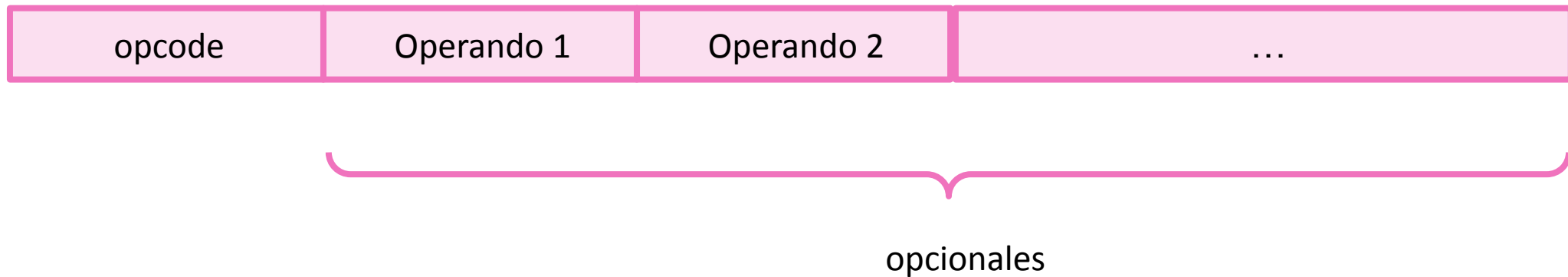
Clasificación de ISAs

- **Formato de las instrucciones**, cómo se representan.
 - La instrucción debe tener el opcode.
 - Y también información adicional, como los operandos y en donde se espera almacenar el resultado.



Clasificación de ISAs

- **Formato de las instrucciones**, cómo se representan.
 - La instrucción debe tener el opcode.
 - Y también información adicional, como los operandos y en donde se espera almacenar el resultado.



- Como todas nuestras instrucciones deben seguir el formato definido, al diseñar la ISA vamos a tener que definir y balancear ciertas cosas.
- Como por ejemplo, cuantos registros queremos, pero tener muchos de ellos puede afectar al tamaño de los datos o de direcciones de memoria que aceptamos.

	Todas las instrucciones son del mismo largo	... o bien el largo de las instrucciones puede variar
Ventajas (relativas)	El hardware es más simple, facilitando la lectura (fetch) y decodificación de las instrucciones.	Hay procesadores en que algunas instrucciones ocupan una palabra, otras ocupan media y hasta un cuarto de palabra, y aún otras ocupan dos palabras.
Desventajas (relativas)	Desperdicia espacio, ya que hay instrucciones que evidentemente podrían ser más cortas, por lo que se ve como ineficiente. Los bits que no se usan de una instrucción simplemente son ignorados por el hardware.	El hardware necesario para leer y decodificar una instrucción es más complejo
Ejemplos de ISAs	RISC-V, ARM, MIPS	Intel 80x86

Clasificación de ISAs

- En general, instrucciones más cortas (menos bits) son mejores que instrucciones más largas.
- Es fácil ver porque instrucciones más cortas serían más eficientes en memoria (contienen menos bits, usan menos espacio).
- Pero también es necesario que nuestro opcode sea capaz de distinguir entre todas las operaciones que tenemos. Al menos vamos a requerir 2^n bits para almacenar el opcode, siendo n el número de operaciones.
- También es necesario considerar espacio para crecer, tal vez después queramos tener más operaciones, el espacio almacenado para el opcode tiene que tener en cuenta esto.
- Por ultimo, el largo de las direcciones de memoria. Si en nuestra memoria almacenamos de a bytes vamos a requerir más direcciones que si almacenamos de a palabras de 4 bytes. En el primer caso vamos a requerir 4 veces más direcciones de memoria que para el segundo considerando una memoria de igual capacidad. Entonces uno puede “ahorrar” 2 bits de dirección almacenando de a palabras más grandes, aunque tiene sus desventajas.

Clasificación de ISAs

- **Tipo de instrucciones**, es decir que hace pueden hacer.
 - Movimiento de datos
 - Operaciones de dos operandos
 - Operaciones de un operando
 - Comparaciones, saltos condicionales, y control de loops
 - Llamadas a funciones
 - Input/output

Clasificación de ISAs

- **Tipo de instrucciones**, es decir que hace pueden hacer.
 - Movimiento de datos (duplicación de datos):
 - Copiar un dato de un lugar a otro (en realidad, crear un nuevo objeto con un patrón de bits idéntico al original) es la operación más fundamental debido a asignaciones en el programa del usuario debido a la necesidad de llevar datos de la memoria a los registros (load) o vice versa (store), o copiar datos entre registros (move), ya que muchas instrucciones de máquina pueden tener acceso a variables sólo cuando éstas están en los registros

Clasificación de ISAs

- **Tipo de instrucciones**, es decir que hace pueden hacer.
 - Operaciones de dos operandos:
 - Combinan dos operandos para producir un resultado.
 - La operación pueden ser de suma, resta, multiplicación y división de números enteros. También podría ser operaciones de punto flotante, funciones booleanas de dos variables.

Clasificación de ISAs

- **Tipo de instrucciones**, es decir que hace pueden hacer.
 - Operaciones de un operando:
 - Tienen un solo operando y producen un resultado.
 - Por ejemplo, la función booleana not, SHL y SHR.
 - Operaciones de dos operandos en que uno de ellos es muy común. Por ejemplo, asignar 0 (CALL address) o sumar 1 (INC address) a una variable.

Clasificación de ISAs

- **Tipo de datos**, es decir si la arquitectura soporta un tipo de dato en particular.
 - Hay instrucciones que esperan un tipo de dato en específico, esto no se puede cambiar (a menos que se cambie la arquitectura).
 - Los tipos de datos pueden ser divididos en dos categorías:
 - Numéricos: enteros sin signo, enteros con signo, punto flotante, ...
 - No numéricos: Caracteres ASCII, caracteres Unicode, Booleanos, punteros (direcciones de memoria).

Clasificación de ISAs

- **Modelo de memoria**, es decir la instrucción ve la memoria.
 - Como esta dividida la memoria?
 - Las memorias dividen su espacio en celdas. El tamaño de estas depende de la arquitectura.
 - Normalmente son de un byte (8 bits), con direcciones consecutivas.
 - Los bytes son agrupados en palabras de 4 u 8 bytes (32 o 64 bits).
 - Si la instrucción quiere utilizar la palabra completa hay que tener en cuenta como fue almacenada en primer lugar; big endian o little endian.

Endianness

- Cómo guardamos un dato de más de un byte en memoria? El orden en el que se guarde la secuencia de bytes es muy importante.
- Si no tenemos un estándar de cómo se guardan los datos en memoria, puede ser que se guarde de cierta manera, pero al momento de leer y utilizar el dato puede ser que se haga de otra forma obteniendo un valor distinto al almacenado originalmente.
- Entonces, cuales son las formas de ordenarlos?
 - Big Endian: Se guarda el byte más significativo primero.
 - Little Endian: Se guarda el byte menos significativo primero.

Endianness: Ejemplo

0x00	0x00	0x00	0x00	0x00
0x01	0xA3	0x01	0x13	0xE9
0x02	0x00	0x00	0x00	0x00
0x03	0x00	0x00	0x00	0x00

- Cómo se interpreta el valor del dato almacenado en la dirección 1, en big endian?
- Ahora, cómo se interpreta el valor del dato almacenado en la dirección 1, en little endian?

Endianness: Ejemplo

0x00	0x00	0x00	0x00	0x00
0x01	0xA3	0x01	0x13	0xE9
0x02	0x00	0x00	0x00	0x00
0x03	0x00	0x00	0x00	0x00

- Cómo se interpreta el valor del dato almacenado en la dirección 1, en big endian?
 - 0xA30113E9
- Ahora, cómo se interpreta el valor del dato almacenado en la dirección 1, en little endian?
 - 0xE91301A3

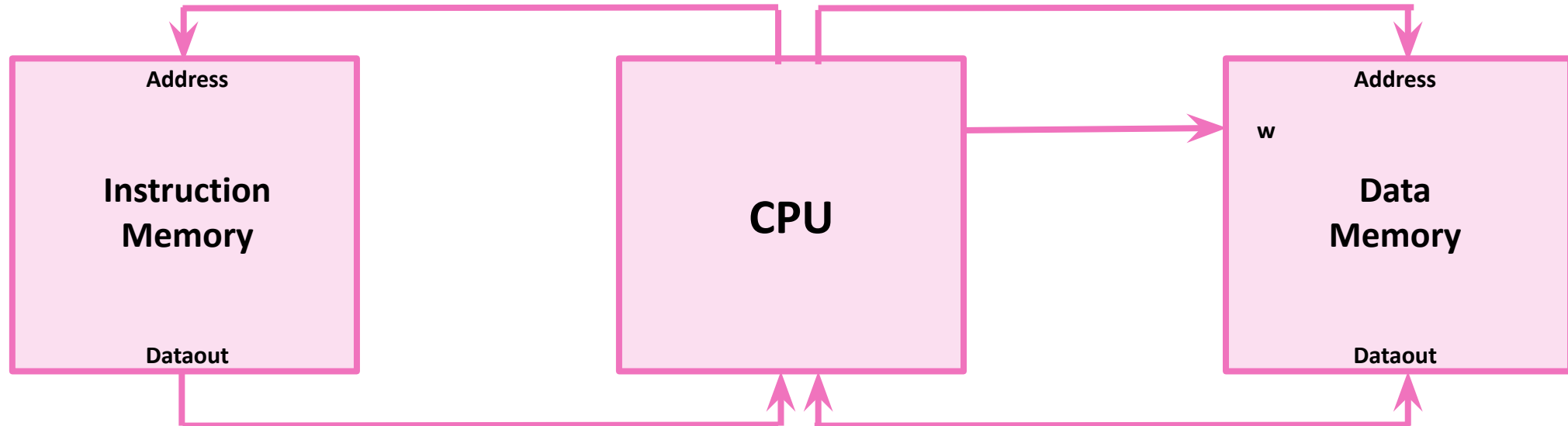
Dos grandes clases de ISAs

- **RISC (Reduced Instruction Set Computer):**
 - Conjunto mínimo de instrucciones que sea suficiente para realizar todas las operaciones
 - Cada una de estas ejecuta una operación básica, normalmente en un ciclo de reloj.
 - Todas las instrucciones tienen el mismo tamaño.
 - Ejemplos de esta clase de ISA son las de los procesadores ARM y MIPS.
- **CISC (Complex Instruction Set Computer):**
 - Muchas instrucciones (varios cientos).
 - Cada una de estas puede ejecutar una operación arbitrariamente compleja, que puede tomar mucho tiempo.
 - Las instrucciones pueden tener tamaños diferentes.
 - Ejemplo de esta clase de ISA son las de los procesadores Intel y AMD

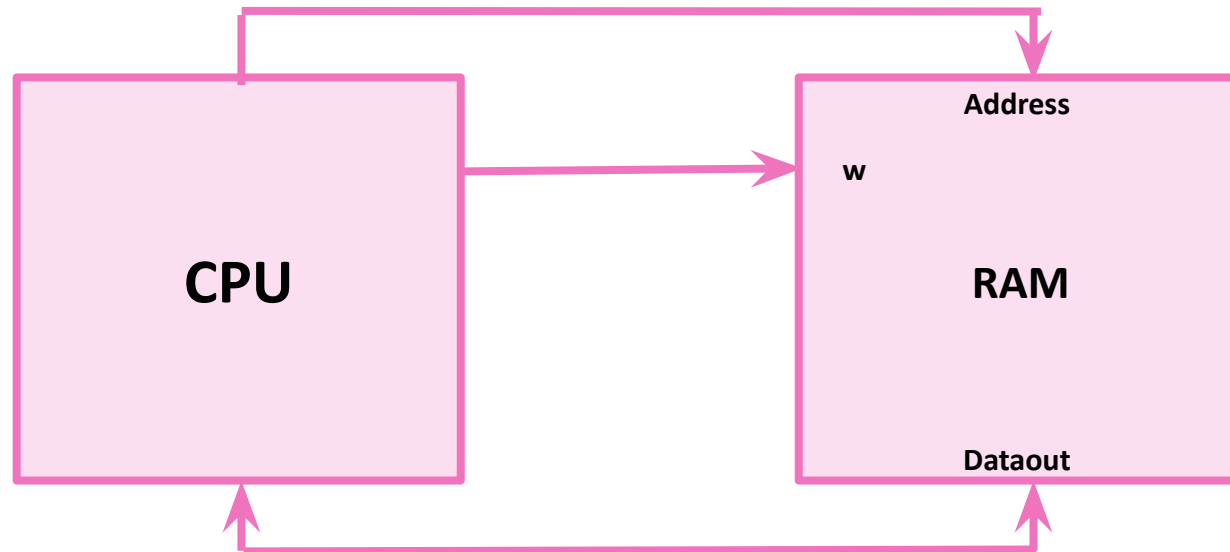
Microarquitectura

- **Arquitectura Harvard:**
 - Memoria de datos independiente a la memoria de instrucciones.
 - El computador que hemos visto hasta ahora (y el del laboratorio) tiene este tipo de arquitectura.
- **Arquitectura Von Neumann:**
 - Tiene una sola memoria en donde se almacenan los datos e instrucciones.
 - Al tener una memoria compartida permite la auto programabilidad, es decir que puedes almacenar “datos” que realmente sean nuevas instrucciones o sobreescribirlas.

Microarquitectura: Harvard



Microarquitectura: Von Neumann





DCC
DEPARTAMENTO DE CIENCIA
DE LA COMPUTACIÓN

IIC2343 - Arquitectura de Computadores

Clase 12 - Arquitectura del set de instrucciones