



## Clasificación de las Arquitecturas

MIA José Rafael Rojano Cáceres  
Arquitectura de Computadoras I



Por la taxonomía de Flynn

## Flynn

		Flujo de datos	
		Simple	Múltiple
Flujo de datos	Simple	SISD	SIMD
	Múltiple	MISD	MIMD

## SISD

- ✦ *Single Instruction – Single Data*: Instrucción simple que trabaja sobre dato simple
- ✦ Arquitectura simple, ejemplo: von neumann

## SIMD

- ✘ *Single Instruction – Multiple Data*: Múltiples flujos de datos pero solo un flujo de instrucciones
- ✘ Corresponden a computadoras vectoriales para cálculos con matrices

## MISD

- ✘ *Multiple Instruction – Simple Data*: Múltiples flujos de instrucciones operan sobre uno de datos
- ✘ Estas computadoras son las más lejanas a los modelos comunes
- ✘ Ejemplo: *Data Flow Machine* (Maq. de flujo de datos)

## MIMD

- ✂ *Multiple Instruction – Multiple Data*: Múltiples flujos de Instrucciones y de datos
- ✂ Bajo este esquema se encuentran las computadoras multiprocesadores (memoria compartida) y los multicomputadoras (memoria independiente)

Por el juego de instrucciones

## Clasificación por instrucciones

### ✚ CISC

- complex instruction set computer

### ✚ RISC

- reduced instruction set computer

## Arquitecturas CISC

✚ La *microprogramación* es una característica importante y esencial de casi todas las arquitecturas CISC.

✚ La microprogramación significa que cada instrucción de máquina es interpretada por un microprograma localizado en una memoria en el circuito integrado del procesador.

✚ Ejemplos son:

- Intel 8086, 8088, 80286, 80386, 80486.
- Motorola 68000, 68010, 68020, 68030, 6840

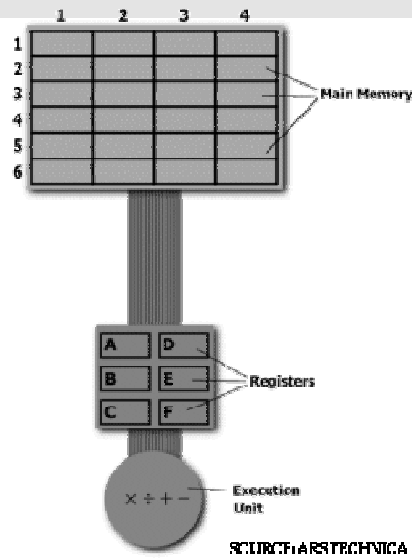
## Arquitectura RISC

- ✦ Debido a que se tiene un conjunto de instrucciones simplificado, éstas se pueden implantar por hardware directamente en la CPU, lo cual elimina la microprogramación y la necesidad de decodificar instrucciones complejas.
- ✦ Permite el uso de la técnica denominada *Pipeline* para paralelizar la ejecución de instrucciones
- ✦ Ejemplos son:

PowerPC, Alpha, Dec, MIPS

La aproximación de las dos arquitecturas

## Computadora Genérica



## Planteamiento

✎ multiplicar dos números en memoria:

- Se tiene un diagrama que representa una memoria para una computadora genérica. La **memoria principal** se divide en localidades numeradas. Renglones del 1 al 6 y columnas del 1 al 4. La **unidad de ejecución** se encarga de llevar a cabo las instrucciones. PERO ésta solo puede operar en datos cargados a los registros (llamados A, B, C, D, E, o F). Imaginemos que obtendremos el producto de dos número – uno esta en la posición 2:3 y el otro en posición 5:2 – el resultado se colocará en

## CISC

- ✖ El objetivo de CISC será completar la tarea en tan pocas líneas de código como sea posible. Esto se lleva a cabo a través de la construcción de un microprocesador que entienda y ejecute un conjunto de operaciones.
- ✖ Imaginemos que para éste ejemplo tenemos la instrucción "MULT".
- ✖ Cuando se ejecuta la instrucción, ésta carga los dos operandos en registros seoadados
- ✖ Se multiplica el operando en la UE, y se almacena entonces el producto en un registro particular.
- ✖ La ejecución de la instrucción concluye con solo la siguiente instrucción:  
MULT 2:3, 5:2
- ✖ MULT es una "instrucción compleja", ya que opera directamente en la memoria de la computadora y no requiere indicaciones explícitas por el programador para realizar la carga de los datos. Analizando un poco la expresión tendríamos que ésta representa una instrucción como:  
"a = a \* b."
- ✖ Una ventaja es que el compilador tiene poco trabajo que hacer al traducir la instrucción a un lenguaje de alto nivel.

## RISC (1)

- ✖ Los procesadores RISC usan instrucciones simples que se pueden ejecutar en un ciclo de reloj. Por lo tanto la instrucción anterior se vería como:
- ✖ "LOAD," cuya tarea es mover datos de la memoria al registro
- ✖ "PROD," que multiplica los operandos y registra el resultado a la memoria. El código apropiado para esta ejecución es;  
LOAD A, 2:3  
LOAD B, 5:2  
PROD A, B  
STORE 2:3, A



## RISC (2)

- ✖ De entrada parece que el programa es menos eficiente ya que requiere de más instrucciones para ejecutarse, más memoria para almacenarse y el compilador deberá trabajar más para realizar la traducción a partir de un lenguaje de alto nivel.
- ✖ Sin embargo los microprocesadores RISC tienen ventajas al ejecutar cada instrucción en un ciclo de reloj. El tiempo de ejecución sería muy similar al de CISC.
- ✖ También requiere de menos transistores en el hardware que las instrucciones complejas permitiendo un número