Tema 0. Introducción al Paralelismo y Organización de un Computador

Organización de Computadores

LUIS ENRIQUE MORENO LORENTE
RAÚL PÉRULA MARTÍNEZ
ALBERTO BRUNETE GONZALEZ
DOMINGO MIGUEL GUINEA GARCIA ALEGRE
CESAR AUGUSTO ARISMENDI GUTIERREZ
JOSÉ CARLOS CASTILLO MONTOYA

Departamento de Ingeniería de Sistemas y Automática

RESUMEN

- Qué es la arquitectura de computadores
- Anticipación, paralelismo y segmentación
- Clasificación de Flynn
- Máquinas MIMD
  - Multiprocesadores
  - Multicomputadores
- Supercomputadores:
  - Computadores vectoriales
  - Computadores SIMD
- Entornos de programación:
  - Paralelismo implícito
  - Paralelismo explícito
- Evolución de las arquitecturas de procesadores

ARQUITECTURA DE COMPUTADORES

- Versa sobre la organización del hardware y los requisitos de programación / software.
- Enganche entre el código ensamblador y el hardware.
- Hardware: CPUs, caches, buses, registros, pipelines, etc.
- Software: juego de instrucciones (código de operación, registros, modos de direccionamiento, etc.).

INSTRUCCIONES

C:

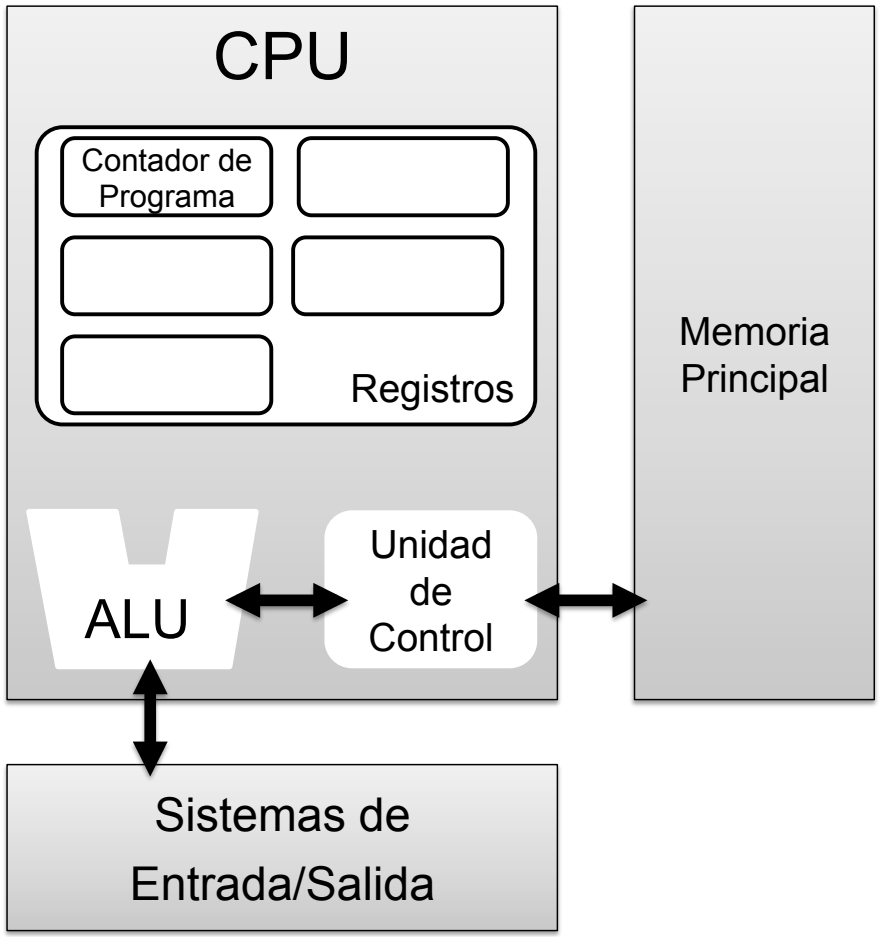
```
x = (a + b) - c;
```

Assembler:

```
ADR r4,a       ; get address for a
LDR r0,[r4]    ; get value of a
ADR r4,b       ; get address for b, reusing r4
LDR r1,[r4]    ; get value of b
ADD r3,r0,r1   ; compute a+b
ADR r4,c       ; get address for c
LDR r2,[r4]    ; get value of c
SUB r3,r3,r2   ; complete computation of x
ADR r4,x       ; get address for x
STR r3,[r4]    ; store value of x
```

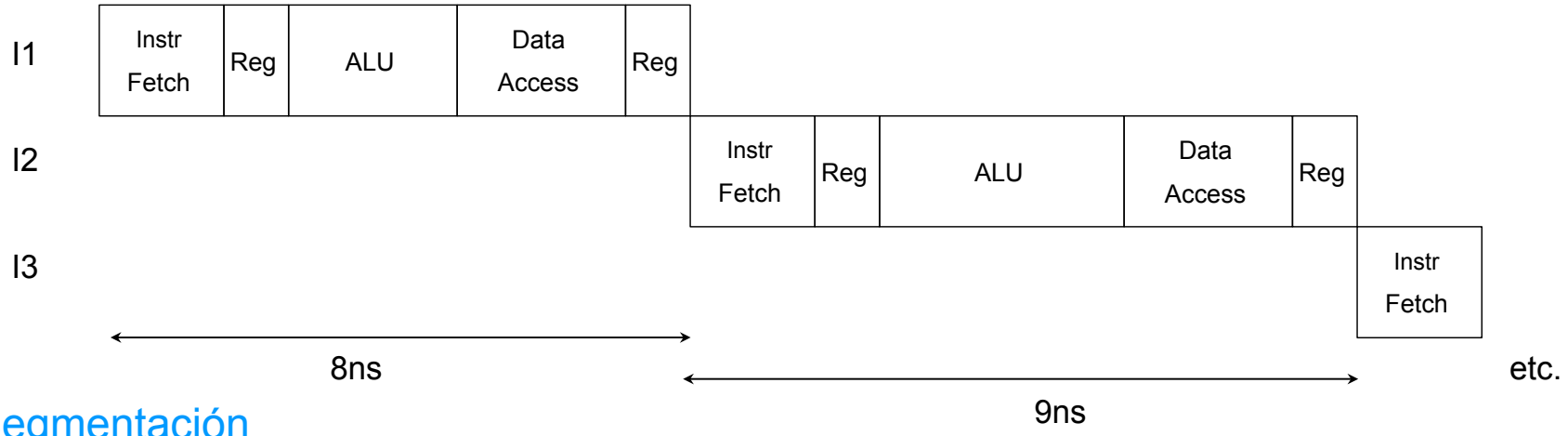
VON NEUMAN

- Mismo almacenamiento para instrucciones y para datos
- La mayoría de las computadoras modernas

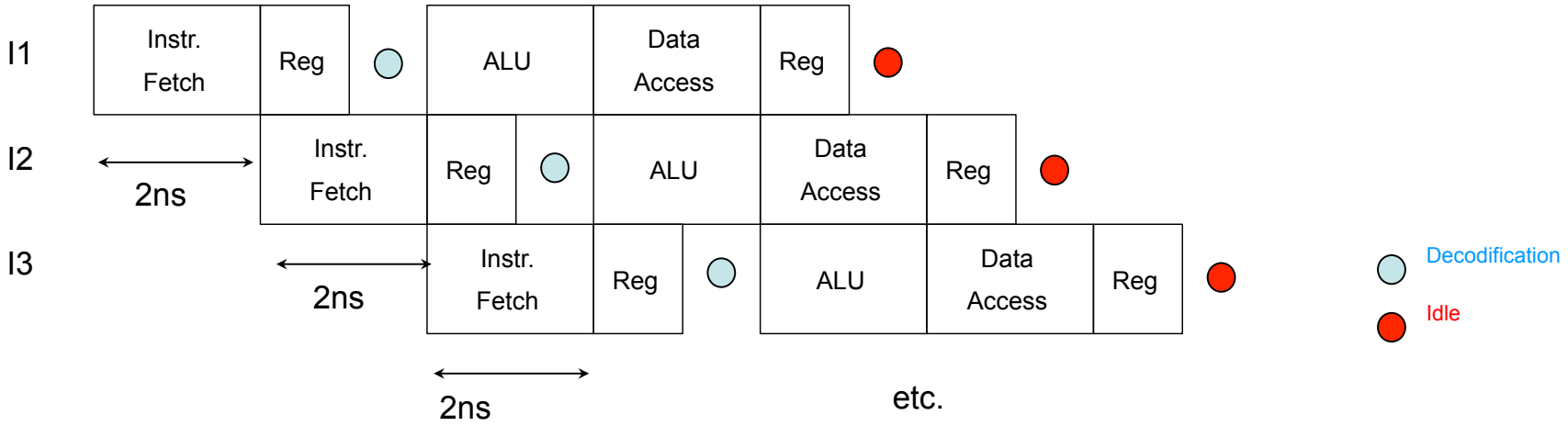


EJECUCIÓN DE INSTRUCCIONES

Secuencial



Segmentación



ANTICIPACIÓN, PARALELISMO Y SEGMENTACIÓN

SEGMENTACIÓN

Descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

ANTICIPACIÓN

Búsqueda "anticipada" de instrucciones para superponer las fases de búsqueda, descodificación y ejecución.

PARALELISMO FUNCIONAL: tareas que se pueden ejecutar al mismo tiempo.

Espacial: replicación de hardware.

Temporal: solapamiento de operaciones funcionales en el tiempo, segmentación.

PARALELISMO DE DATOS

Se ejecuta una instrucción sobre un conjunto de datos.

PARALELISMO A NIVEL DE MÁQUINA VS PROCESADOR

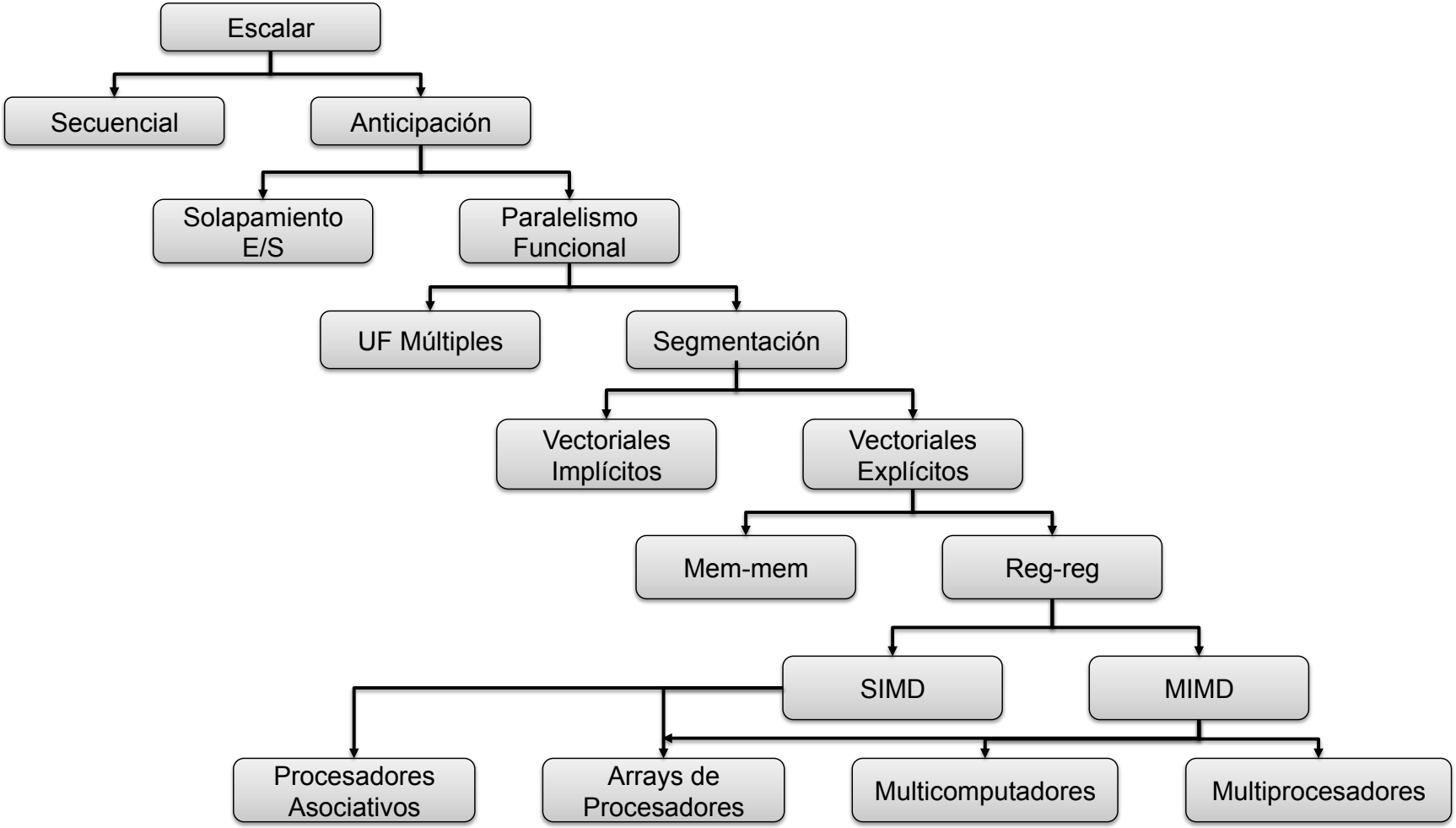
- El paralelismo en los computadores ha tenido dos aspectos que han evolucionado en paralelo:
  - Paralelismo a nivel de máquina:

    En este nivel se distinguen diferentes tipos de organización de máquinas que en este curso se mencionan y se ampliarán en cursos posteriores.
  - Paralelismo a nivel de procesador:

    En este nivel se pueden estudiar una serie de avances organizativos que han conducido a una mejora en la eficiencia.

    El curso de organización de computadores se centrara en este nivel.

Evolución de la Arquitectura de Computadores



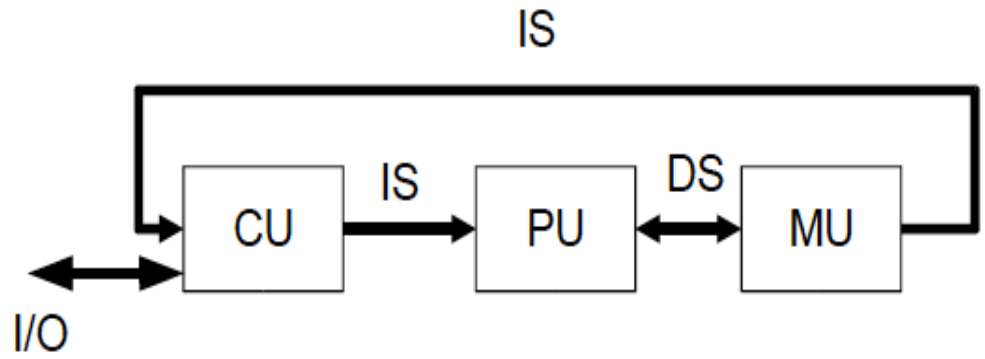
CLASIFICACIÓN DE FLYNN

- Se considera:
  - Flujo de instrucciones: secuencia de instrucciones ejecutadas por la máquina.
  - Flujo de datos: secuencia de datos, tanto de entrada como resultados finales e intermedios.
    - SISD (single instruction stream over a single data stream).
    - SIMD (single instruction stream over multiple data stream).
    - MIMD (multiple instruction streams over multiple data streams).
    - MISD (multiple instruction streams and a single data streams).

Flujo de instrucciones \ Flujo de datos	ÚNICO	MÚLTIPLE
ÚNICO	SISD	SIMD
MÚLTIPLE	MISD	MIMD

ARQUITECTURA SISD

- Corresponde a un uni-procesador.
- Puede estar segmentado o no.



CU = control unit
PU = Processing unit
MU = memory unit
IS = instruction stream
DS = data stream
PE = processing element
LM = Local Memory

Imagen de Hwang, 1993

ARQUITECTURA SIMD

- Corresponde a procesadores vectoriales / matriciales.
- Una única memoria de instrucciones y varias memorias de datos (distribuidas).
- Control centralizado y operaciones distribuidas.

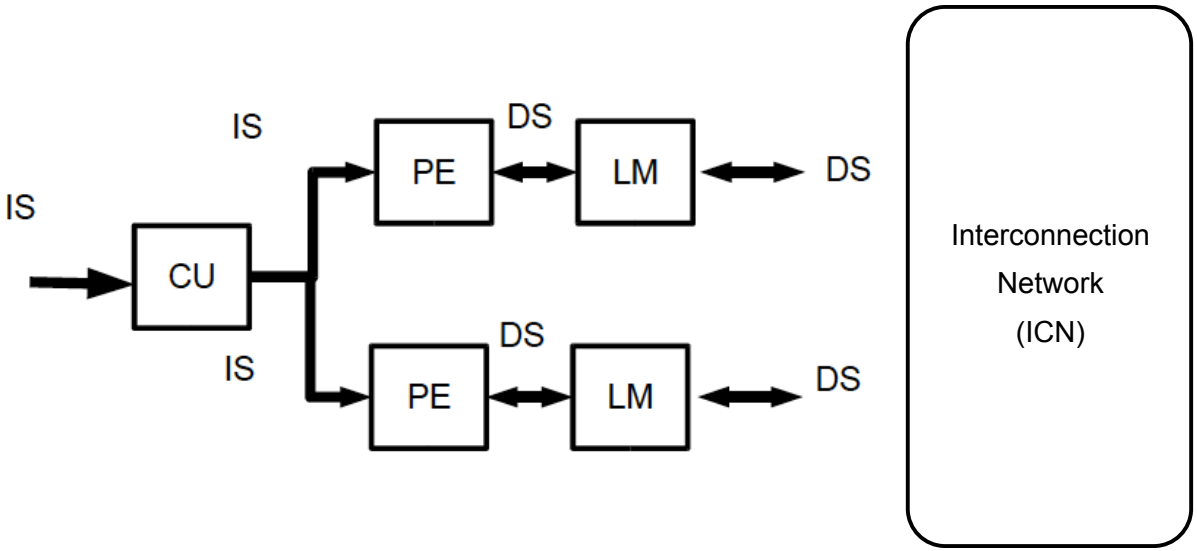


Imagen de Hwang, 1993

ARQUITECTURA MIMD

- Memoria compartida.
- Procesadores con paralelismo intrínseco.
- Ordenadores multi-procesador.

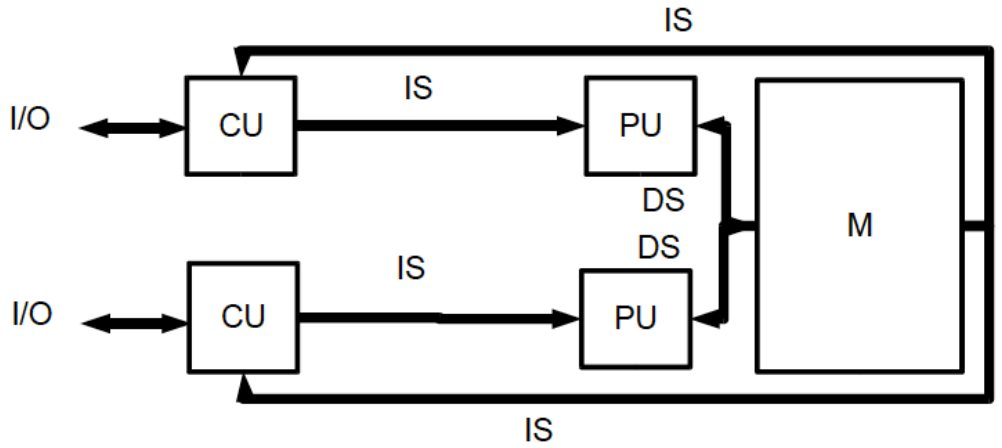


Imagen de Hwang, 1993

ARQUITECTURA MISD

- Procesadores sistólicos.
- Cada unidad de proceso realiza una operación sobre los datos.
- Los datos pasan por todas o parte de la unidades de proceso hasta que se ha completado el algoritmo.
- Pipelined SIMD.
- No muy utilizados.

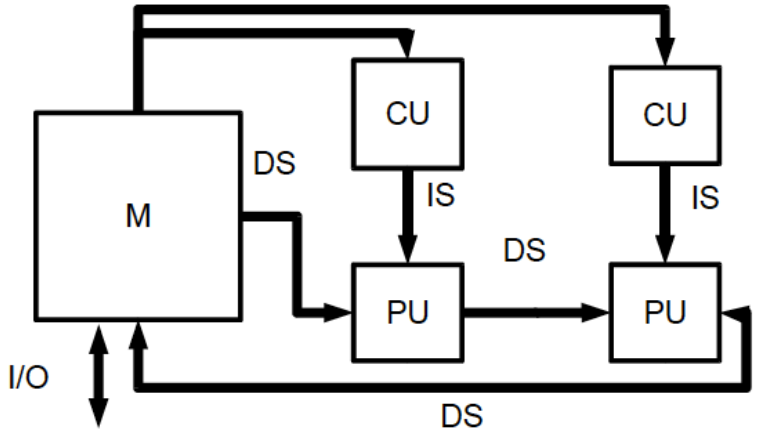


Imagen de Hwang, 1993

SISD VS SIMD

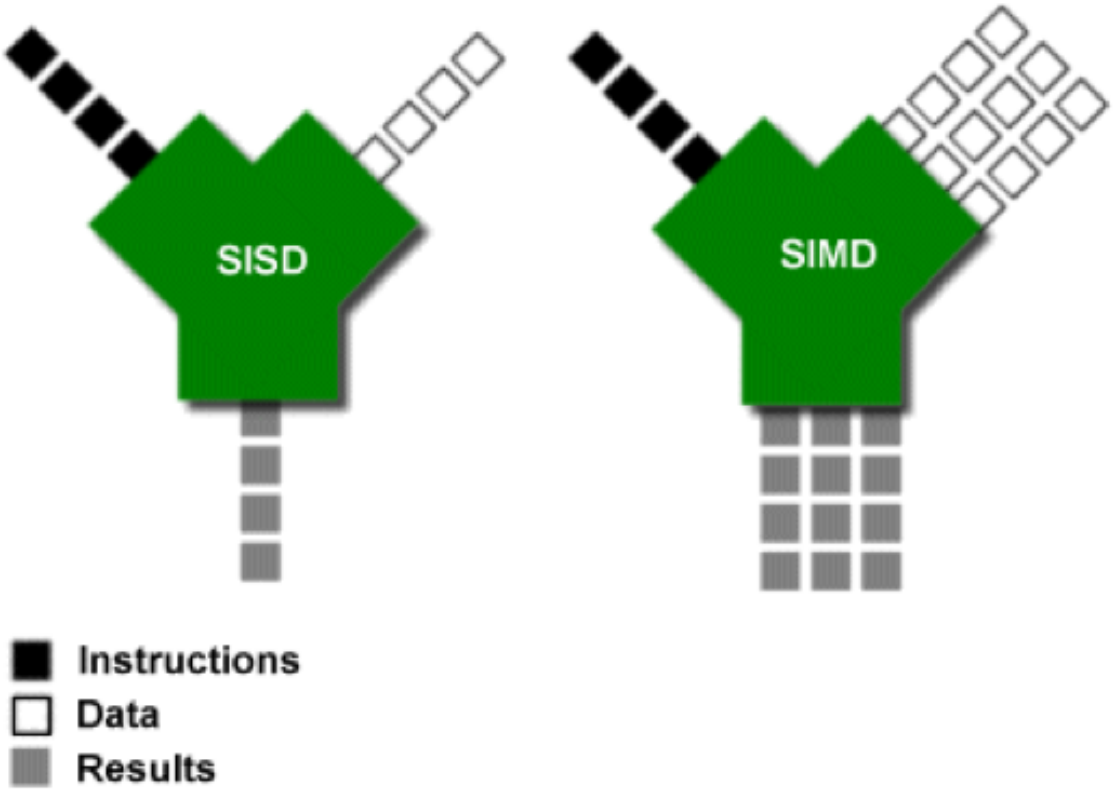


Imagen de Ars Tecnica, 2006

ARQUITECTURA MIMD

- Es el modelo de máquina más general y versátil.
- Hay dos grandes tipos de máquinas:
  - Procesadores MIMD con memoria compartida:
    - Multiprocesadores (comparten el mismo espacio de direcciones de memoria).
    - Fuertemente acoplados.
  - Procesadores MIMD con memoria no compartida:
    - Multicomputadores (paso de mensajes).
    - Acoplamiento débil.

MULTIPROCESADORES

- Pueden tener diferentes tipos de estructuras dependiendo de como estén organizadas los accesos a las diferentes memorias que constituyen la memoria de la máquina.

- Tipos:
  - UMA (Uniform Memory Access).
  - NUMA (Non Uniform Memory Access).
  - COMA (Cache Only Memory Access).

MULTIPROCESADORES: MODELO UMA (UNIFORM MEMORY ACCESS)

- El tiempo de acceso a cualquiera de las memorias es el mismo.

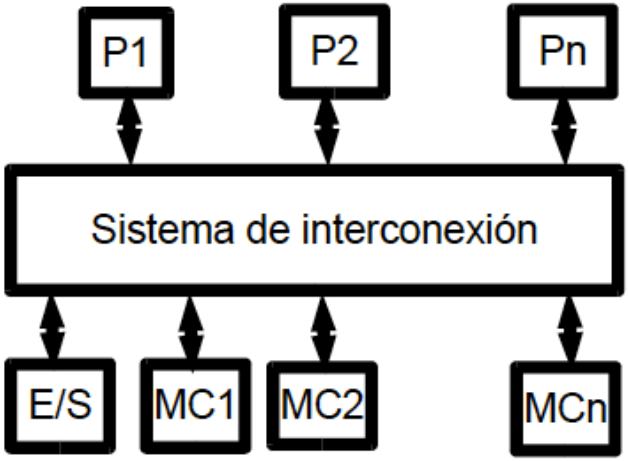


Imagen de Hwang, 1993

P - Procesador

MC - Memoria Compartida

MULTIPROCESADORES: NUMA (NON UNIFORM MEMORY ACCESS)

- El tiempo de acceso a las memorias NO es el mismo
- Las memorias locales son de acceso más rápido, puede haber memorias globales,

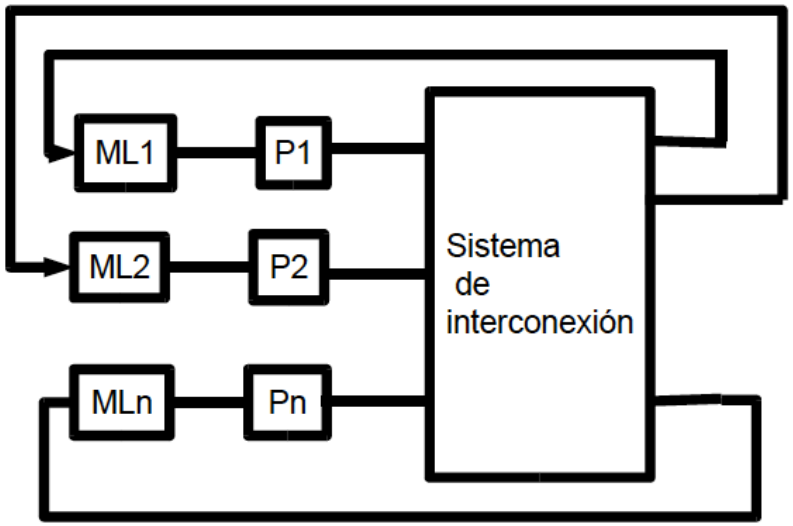


Imagen de Hwang, 1993

P - Procesador

ML - Memoria local

MULTIPROCESADORES: MODELO NUMA

- Pueden estructurarse en clusters de máquinas.

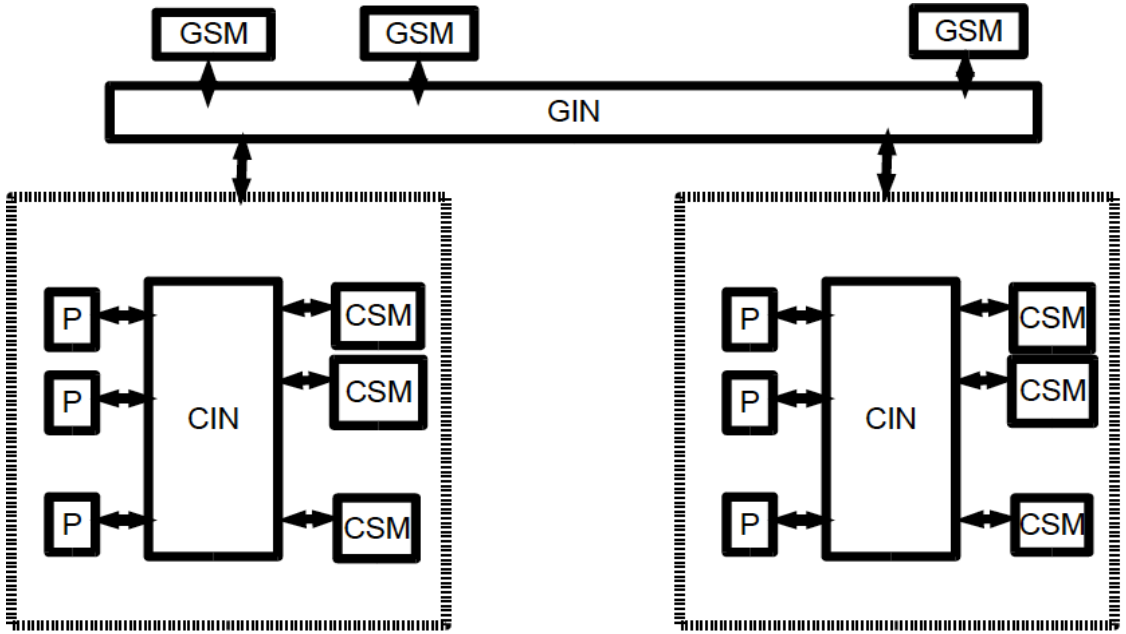


Imagen de Hwang, 1993

P - Procesador

GSM - Global Shared Memory

CSM - Cluster Shared Memory

GIN - Global Interconnection Network

MULTIPROCESADORES: COMA (CACHE ONLY MEMORY ACCESS)

- En los modelos UMA y NUMA aparece el problema de coherencia de caches.
- El modelo COMA deriva de los protocolos de coherencia de caches.
- No existe una memoria física lineal propiamente dicha sino sólo las caches y los protocolos para localizar toda la información.

P - Procesador

C - Cache

D - Diretorio

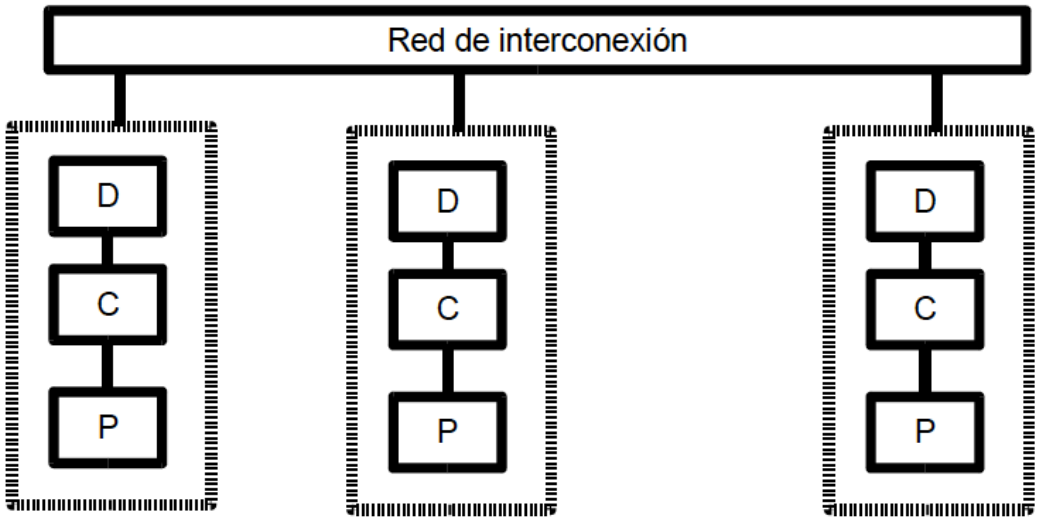


Imagen de Hwang, 1993

MULTICOMPUTADORES

- Es básicamente una red de computadores que se comunica por paso de mensajes, el acoplamiento es más débil que en los multiprocesadores.
- NORMA - NO Remote Memory Access

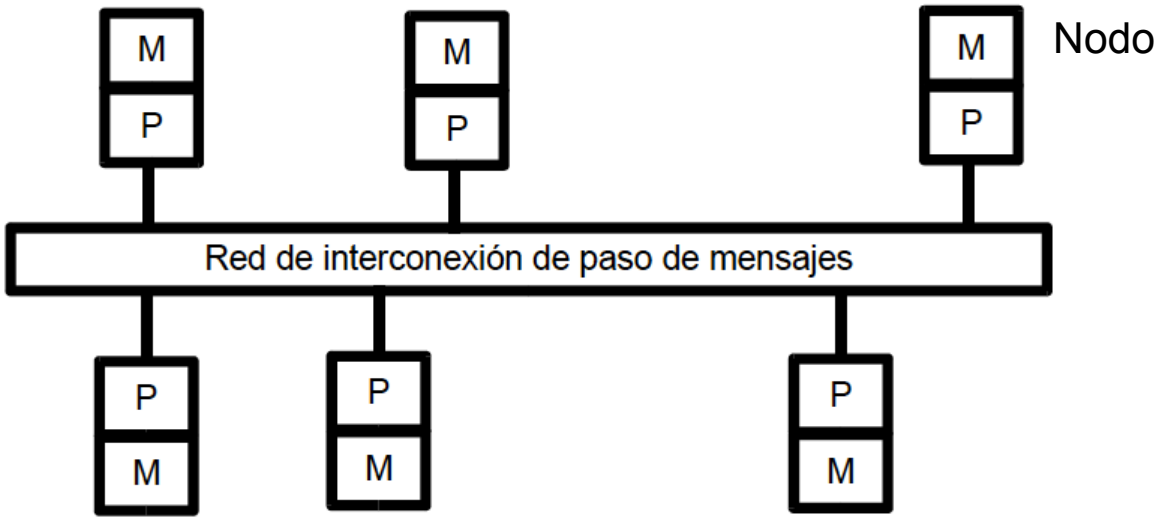


Imagen de Hwang, 1993

MULTICOMPUTADORES

Ventajas:

- aumento del ancho de banda de la memoria
- reduce latencias en el acceso a memoria
- soporta mayor número de procesadores

Particularidades:

- Memorias locales privadas
- Comunicación mediante paso de mensajes
- MPI Message Passing Interface
- PVM: parallel Virtual Machine

MULTICOMPUTADORES

- 1ª generación (83-87)
  - Arquitectura hipercubo
  - Conmutación software de mensajes
  - CosmicCube
  - Intel iPSC860
- 2ª generación (87-92)
  - Arquitectura en retícula (mesh)
  - Enrutamiento hardware de mensajes
  - Parsys Supernodo 1000
  - Intel Paragon
- 3ª generación (93-...)
  - Un sólo chip con el procesador y comunicaciones
  - MIT J-Machine
  - Caltech Mosaic

SUPERCOMPUTADORES

(1) Computadores vectoriales

Procesador escalar + procesador vectorial

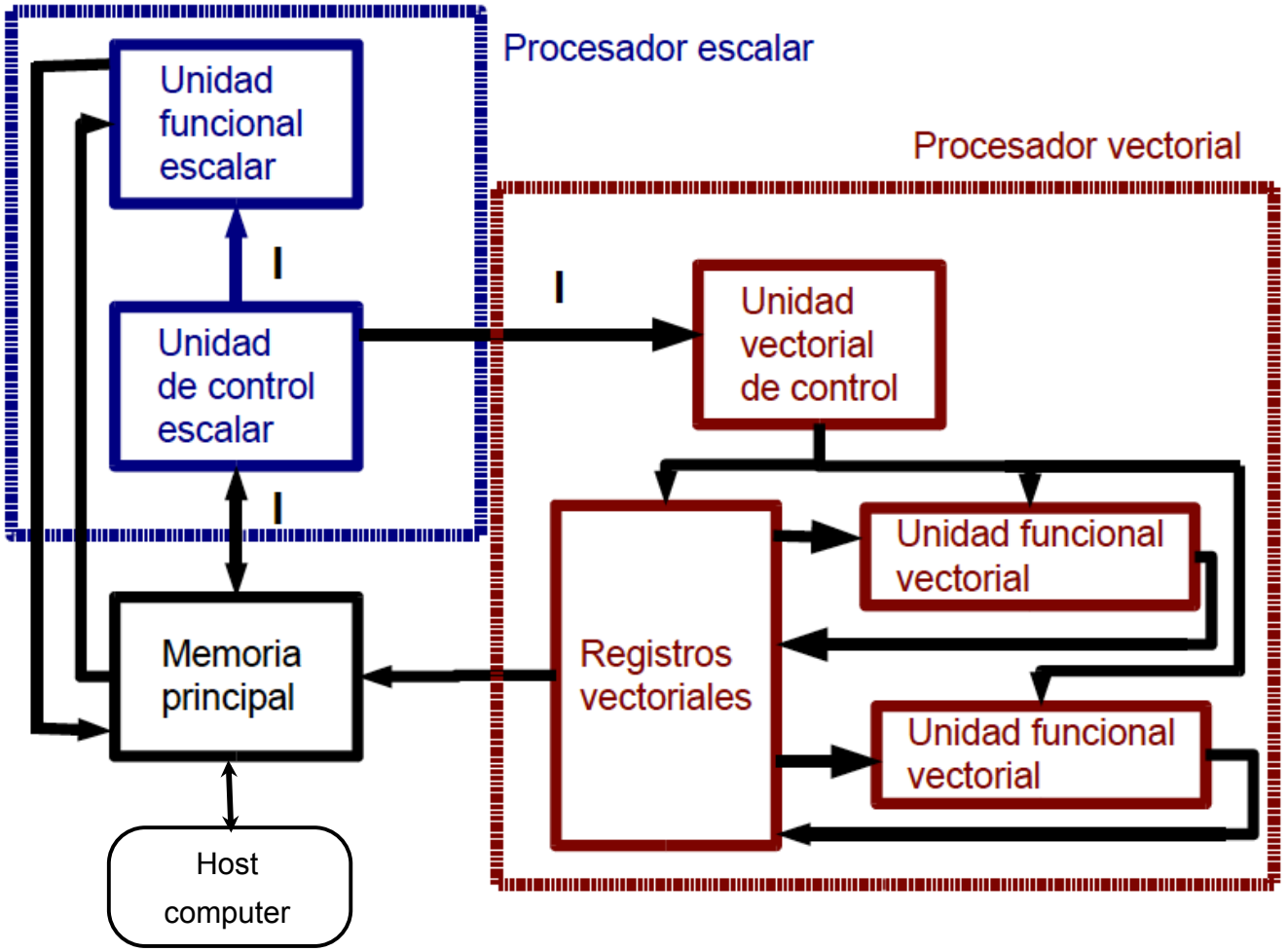
Número pequeño de procesadores potentes con hardware específico para procesamiento vectorial

(2) Máquinas SIMD (matriciales)

Múltiples elementos de proceso bajo la supervisión de una única unidad de control

Enfocadas al paralelismo masivo de datos

COMPUTADORES VECTORIALES



COMPUTADORES VECTORIALES: ARQUITECTURAS

Memoria a memoria

Emplean una unidad de memoria en la que almacenan los operandos y resultados vectoriales.

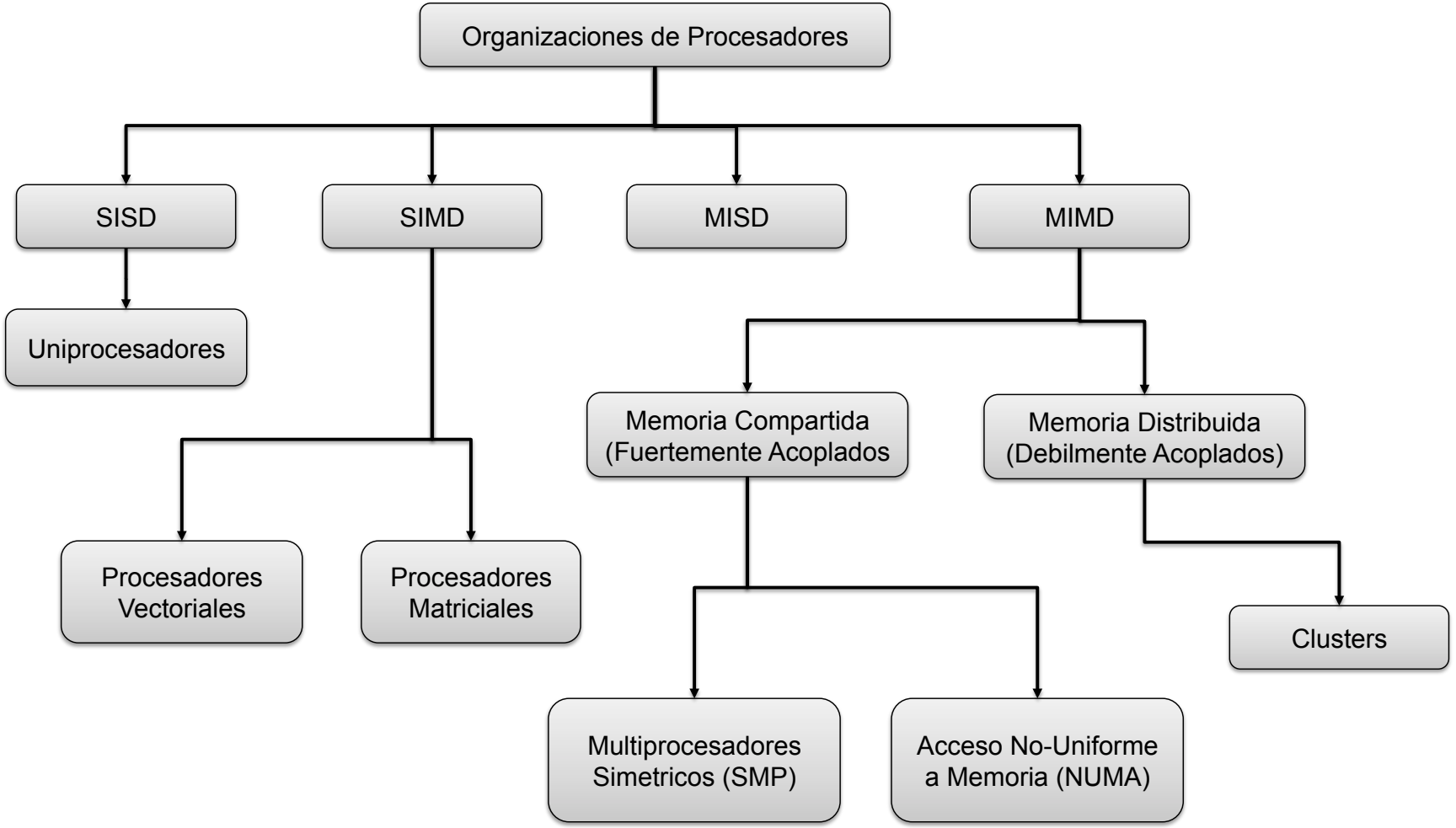
CDC-STAR100
TI-ASC
CYBER-205

Registro a registro

Los registros vectoriales almacenan operandos y resultados vectoriales.

CRAY-1
C-1, C-2
VP100/200
J-90

RESUMEN



ENTORNOS DE PROGRAMACIÓN

El paralelismo debe ser soportado a nivel de:

- Sistema operativo
- Herramientas de programación

Los sistemas operativos paralelos deben permitir:

- Planificación paralela
- Asignación de recursos (memoria, periféricos...)
- Comunicación
- Protección de memoria y conjuntos de datos
- Prevención de bloqueos
- Manejo de excepciones

HERRAMIENTAS DE PROGRAMACIÓN

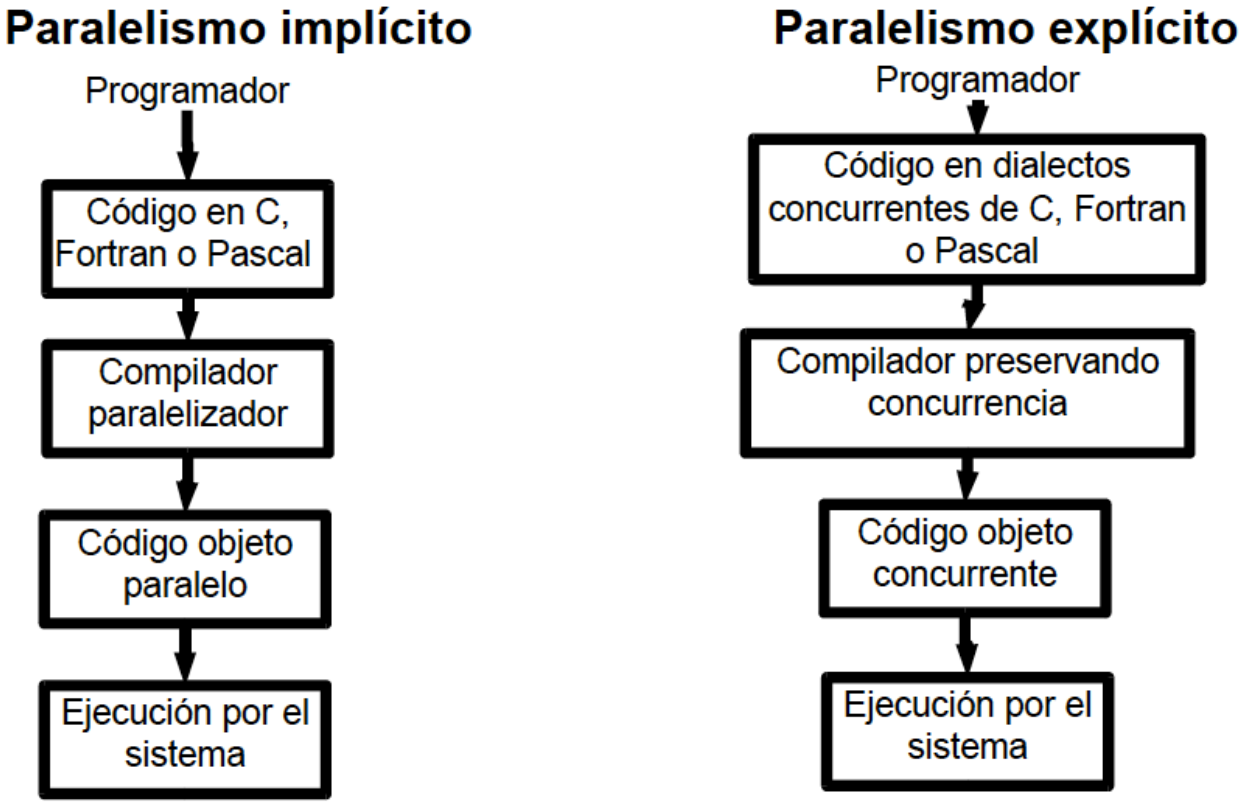


Imagen de Hwang, 1993

ARQUITECTURAS DEL PROCESADOR: EVOLUCIÓN

- Chip uniprocesador
  - Micros convencionales
  - Procesadores superescalares
  - Procesadores superespeculativos
  - Procesadores multithread
- Trace processors
  - Chip multiprocesador
  - Vector IRAM processors
  - Single-chip multiprocesors (multicore)
  - Raw configurable processors.

PROCESADORES SUPERESCALARES MULTIVÍA

- Capaces de comenzar hasta 16-32 ejecuciones de instrucciones por ciclo. Requieren una alta capacidad de suministro de instrucciones y datos.

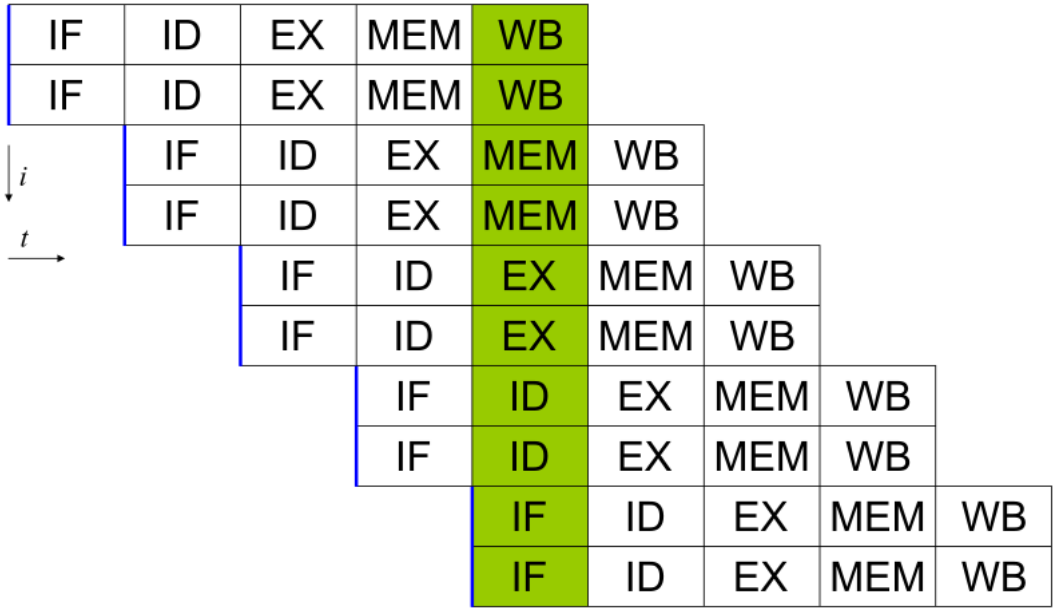


Imagen de Superescalar, 2014

ARQUITECTURAS SUPERESPECULATIVAS

- La ejecución especulativa es la ejecución de código por parte del procesador que no tiene por qué ser necesaria a priori.
- La ejecución especulativa no es más que una optimización. Sólo es útil cuando la ejecución previa requiere menos tiempo y espacio que el que requeriría la ejecución posterior, siendo este ahorro lo suficientemente importante como para compensar el esfuerzo gastado en caso de que el resultado de la operación nunca llegue a usarse.
- Hay dos líneas:
  - Microarquitecturas con búsquedas de hasta 32 instr (anchura), buffer de reordenación de 128, y colas de almacenamiento de 128 entradas para varias configuraciones de memoria.
  - Utilización de modelos de dependencia débil para conseguir prestaciones superescalares de hasta 19 instrucciones por ciclo (IPC) puntuales y 9 IPC de media en el test SPEC-95.

MULTI-THREADING

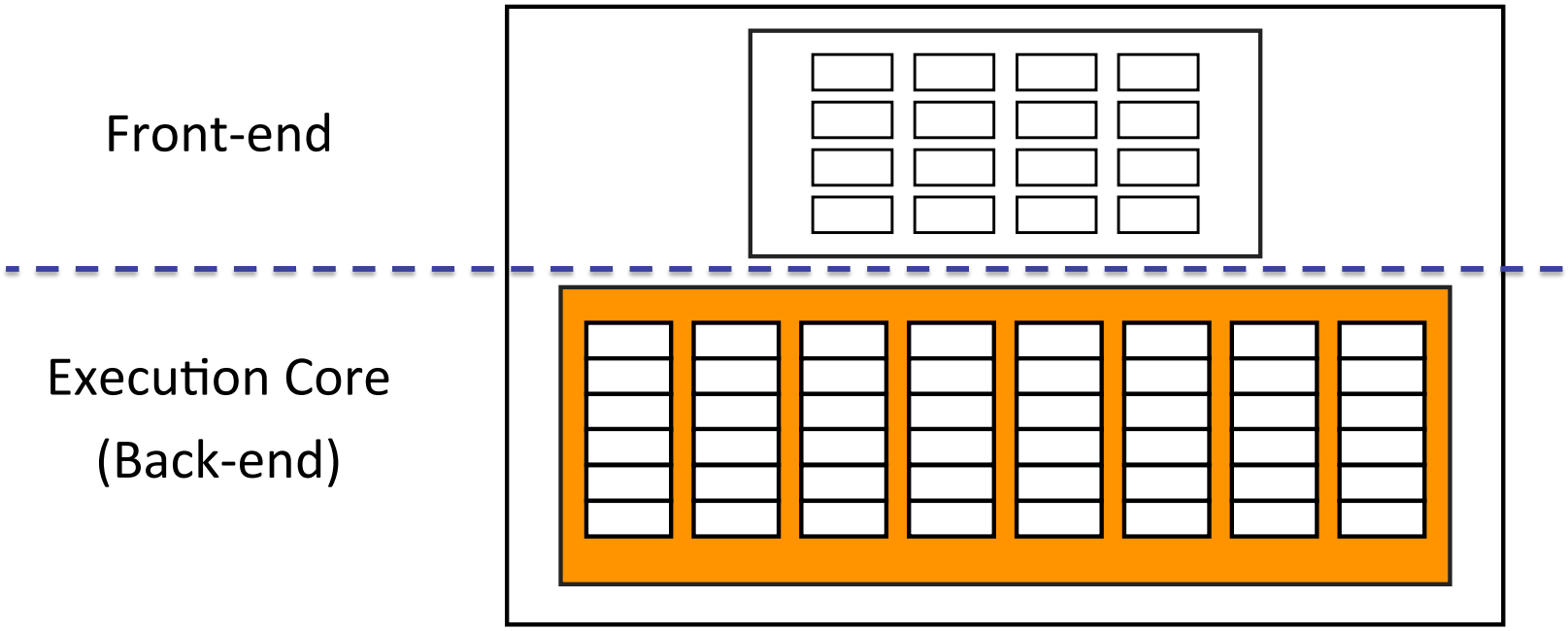
- El paradigma de multihilo ha llegado a ser más popular a medida que los esfuerzos para llevar más adelante el paralelismo a nivel de instrucción se han atascado desde finales de los años 1990.

- Hardware Support para threads:
  - SuperThreading
  - HyperThreading
  - Simultaneous Multi Threading (SMT)

SINGLE-THREADED PROCESSOR

- El procesador proporciona la ilusión de la ejecución concurrente.
  - Front-end: fetching/decoding/reordering.
  - Execution core: ejecución real.
- Múltiples programas en la memoria, pero sólo uno se ejecuta en cada momento.
- Time-slicing via context switching.

SIMPLIFIED EXAMPLE CPU

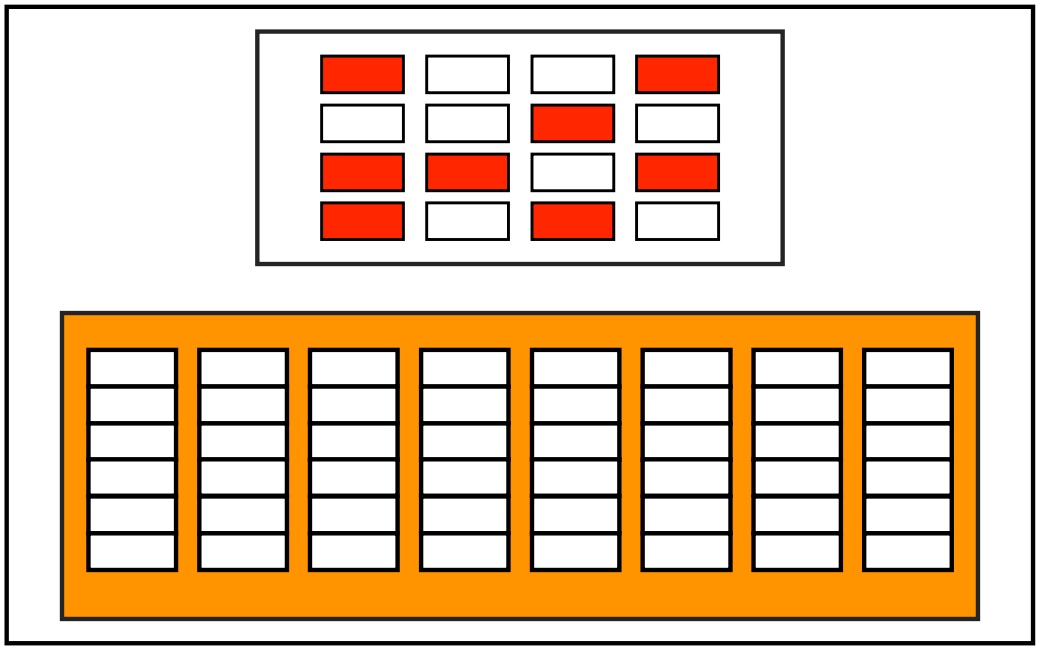


- El front-end puede emitir 4 instrucciones simultáneas al execution core.
- 4-stage pipeline.
- El execution core tiene 8 unidades funcionales.
- Cada una: 6-stage pipeline.

SIMPLIFIED EXAMPLE CPU

Front-end

Execution Core



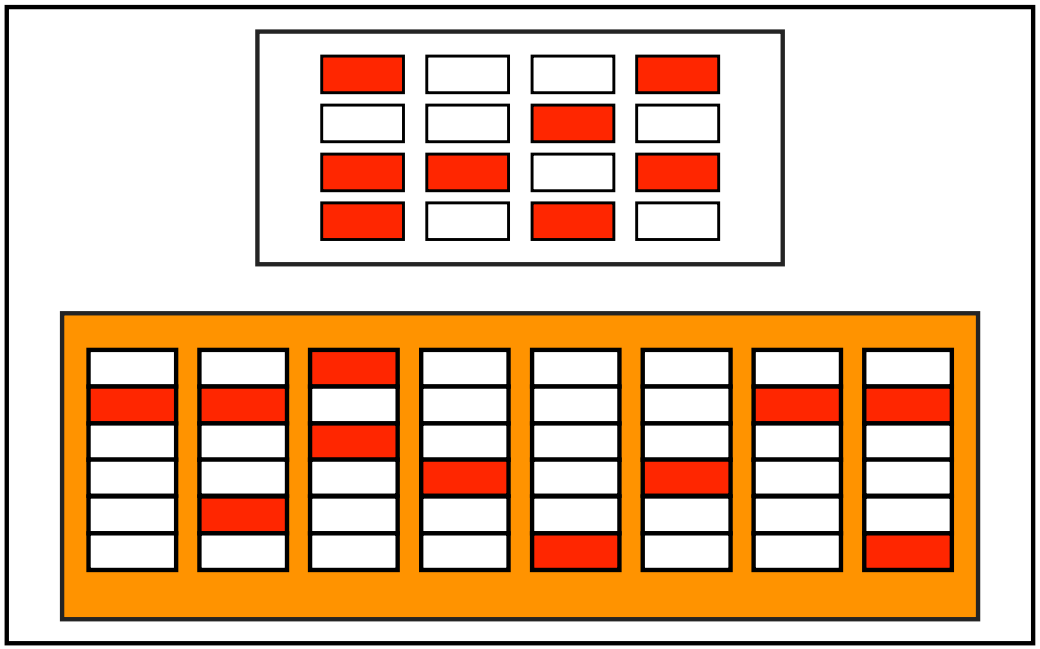
- El front-end está a punto de emitir 2 instrucciones.
- En el ciclo siguiente se emitirán 3.
- En el siguiente 1, etc.
- Un hardware complejo se encarga de decidir la siguiente instrucción en ser emitida.

SIMPLIFIED EXAMPLE CPU

Front-end

Execution

Core



- En el ciclo actual, 2 unidades funcionales están siendo usadas.
- En el siguiente sólo una, etc.
- Los huecos en blanco son pipeline “bubbles”: oportunidades perdidas para hacer algo útil, debido a un mal paralelismo a nivel de instrucciones

BLOCK MULTI-THREADING

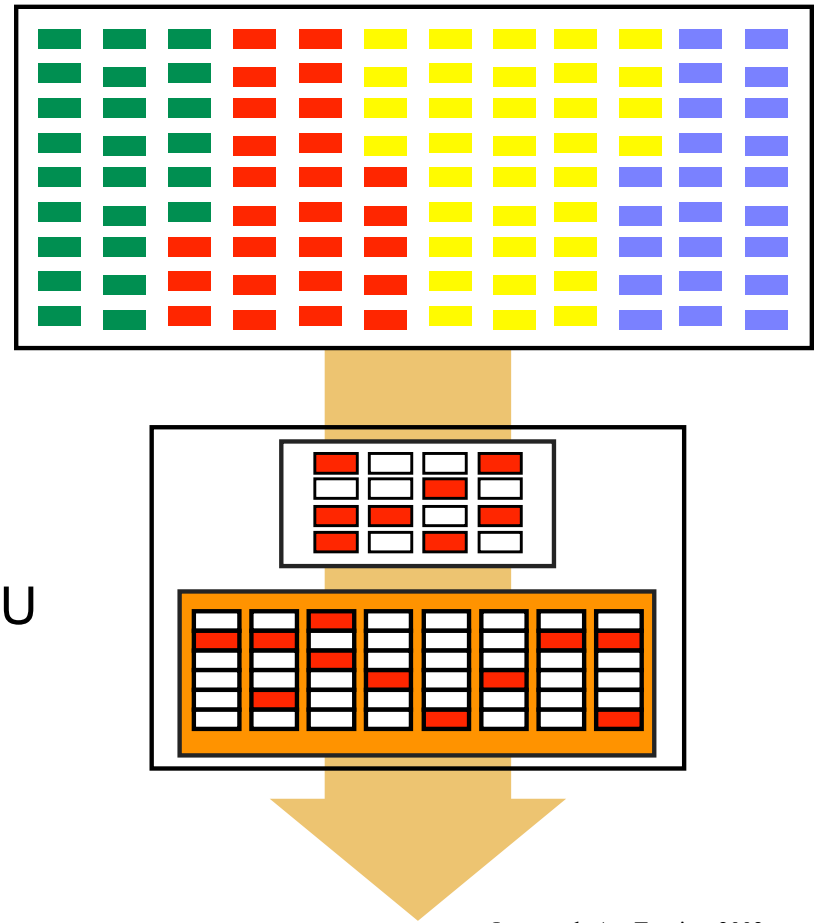


Imagen de Ars Tecnica, 2002

- 4 threads en memoria.
- En la arquitectura tradicional, sólo se ejcuta el rojo.
- Cuando el thread rojo se atasca o termina, se pasa a otro thread.
- Coarse-grained multithreading.

SINGLE-THREADED SMP (SYMMETRIC MULTI-PROCESSING)

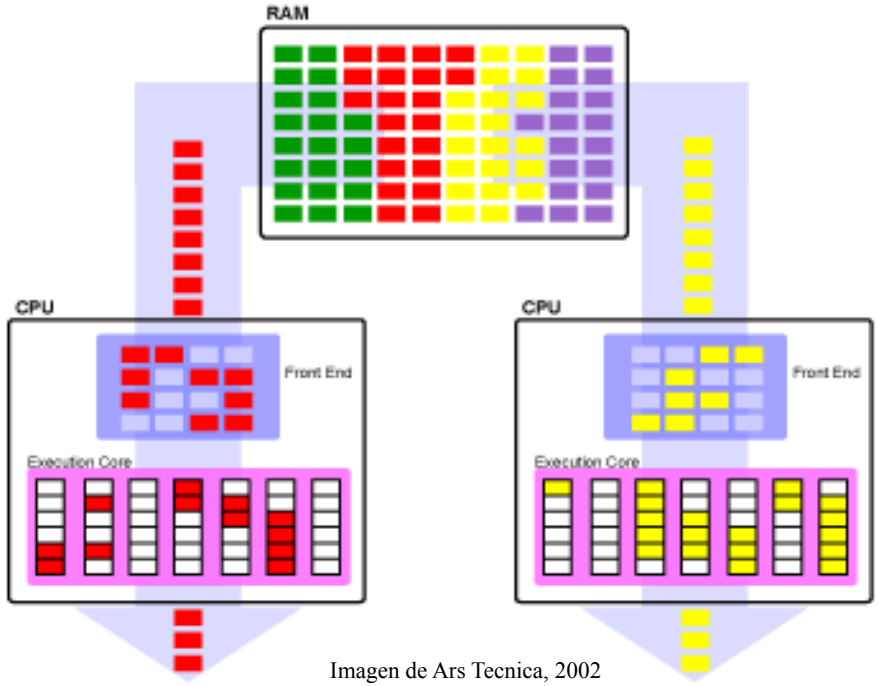


Imagen de Ars Tecnica, 2002

- 2 threads se ejecutan a la vez, pero en CPUs distintas.
- Se dobla el número de burbujas.
- Doble velocidad y doble gasto.

INTERLEAVED MULTI-THREADING. SUPER-THREADING

- Fine-grained o time-sliced multithreading.
- Principio: el procesador puede ejecutar más de un thread al mismo tiempo.
- Requiere mejor hardware.
- Menor gasto.
- En cada etapa del front-end solo se puede ejecutar un thread.

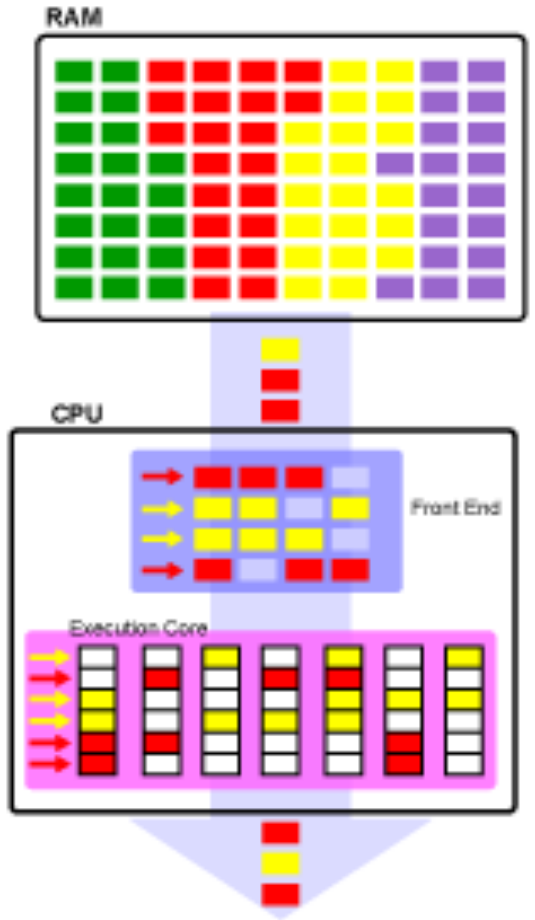


Imagen de Ars Tecnica, 2002

SIMULTANEOUS MULTI-THREADING. HYPER-THREADING

- Principio: el procesador puede ejecutar más de un hilo a la vez, incluso dentro del mismo ciclo de reloj!
- Requiere hardware aún más complejo.
- En el esquema: sólo dos threads se ejecutan simultáneamente.
- Desde el punto de vista operativo, hay dos procesadores "lógicos".

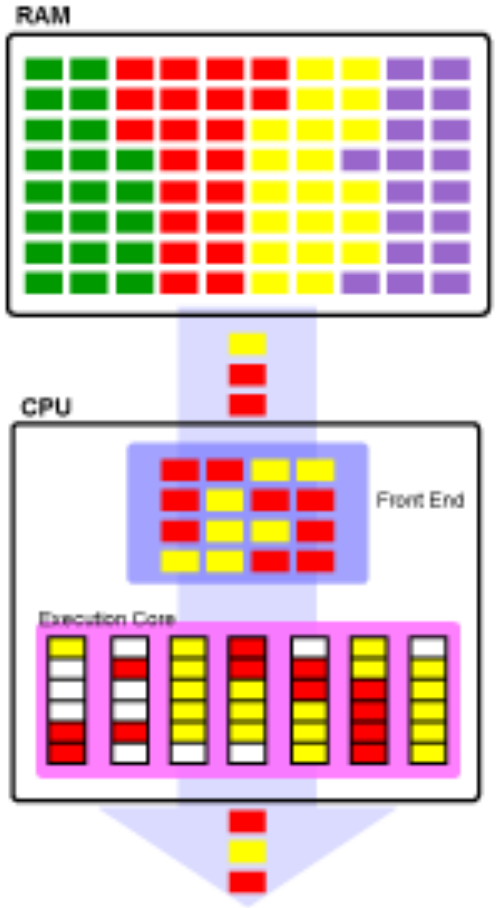


Imagen de Ars Tecnica, 2002

TRACE (MULTISCALAR) PROCESSORS

- La idea es utilizar un trace processor que consiste en múltiples cores de ejecución en un chip, cada uno de los cuales ejecuta simultáneamente una traza diferente del código.
  - Todos excepto uno de los cores ejecutan trazas especulativas utilizando predicción de saltos para seleccionar las trazas.
  - La idea es que 4 cores de procesamiento puedan ejecutar 16 o más instrucciones por ciclo si cada core ejecuta 4 instrucciones por ciclo.

VECTOR (IRAM) PROCESSORS

- Los Intelligen Random-Access Memory processors trabajan sobre tecnologías de acceso a memoria DRAM, en la hipótesis de que el cuello de botella en el futuro estará en el sistema de memoria.

  – Buscan construir multiprocesadores escalables empotrados en un único chip con arrays de memoria de gran escala.
  – Al meter la memoria en el chip aumenta el ancho de banda de acceso a la memoria lo que permitiría un procesamiento vectorial más barato.
  – Para ejecutar código vectorial el código tiene que ser vectorizable.

SINGLE-CHIP MULTIPROCESSORS (MULTICORE)

- Buscan meter en un chip entre 4 y 16 procesadores monocore avanzados.
  - Cada uno de los procesadores está fuertemente acoplados con una cache pequeña de 1er nivel, y todos los procesadores comparten una cache de 2º nivel.
  - Los procesadores pueden colaborar en tareas paralelas o ejecutar tareas independientes.
  - Requiere código con paralelismo explícito para aprovechar todo el potencial.

RAW (CONFIGURABLE) PROCESSORS

- Buscan explotar el paralelismo masivo dentro del chip, con cientos de procesadores muy simples, cada uno con una cierta lógica reconfigurable.
  - La ejecución paralela y las comunicaciones se reordenan por software para su control y coordinación.
  - Esto elimina las interfaces del juego de instrucciones ya que la arquitectura es replicada directamente para su exposición al compilador.
  - El compilador adapta el hardware a cada aplicación.

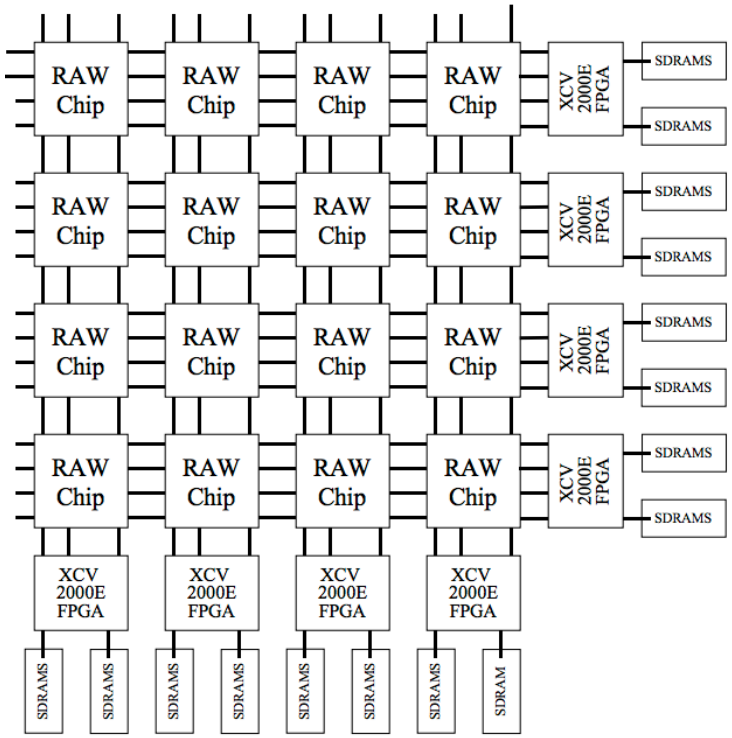


Imagen de Raw Architecture Processor, 2014

REFERENCIAS

[Silc, 1999] J. SILC. Processor Architecture. Springer 1999

[Hwang, 1993] K. HWANG Advanced Computer Architecture, Mc Graw-Hill, 1993

[Ars Tecnica, 2006] Peakstream article, Ars Technica, http://arstechnica.com/gadgets/2006/09/7763/

[Superescalar, 2014] http://en.wikipedia.org/wiki/Superscalar

[Ars Tecnica, 2002] Introduction to Multithreading, Superthreading and Hyperthreading, Ars Technica, http://arstechnica.com/features/2002/10/hyperthreading/

[Raw Architecture Processor, 2014] Raw Architecture Processor, http://groups.csail.mit.edu/cag/raw/