

Superescalares

Ejemplos

Eficiencia de energía

Microprocessor	Year	Clock Rate	Pipeline Stages	Issue Width	Out-of-Order/ Speculation	Cores/ Chip	Power
Intel 486	1989	25 MHz	5	1	No	1	5W
Intel Pentium	1993	66 MHz	5	2	No	1	10W
Intel Pentium Pro	1997	200 MHz	10	3	Yes	1	29W
Intel Pentium 4 Willamette	2001	2000 MHz	22	3	Yes	1	75W
Intel Pentium 4 Prescott	2004	3600 MHz	31	3	Yes	1	103W
Intel Core	2006	3000 MHz	14	4	Yes	2	75W
Intel Core i7 Nehalem	2008	3600 MHz	14	4	Yes	2-4	87W
Intel Core Westmere	2010	3730 MHz	14	4	Yes	6	130W
Intel Core i7 Ivy Bridge	2012	3400 MHz	14	4	Yes	6	130W
Intel Core Broadwell	2014	3700 MHz	14	4	Yes	10	140W
Intel Core i9 Skylake	2016	3100 MHz	14	4	Yes	14	165W
Intel Ice Lake	2018	4200 MHz	14	4	Yes	16	185W

Fuente: COD2-RISC-V, p. 353

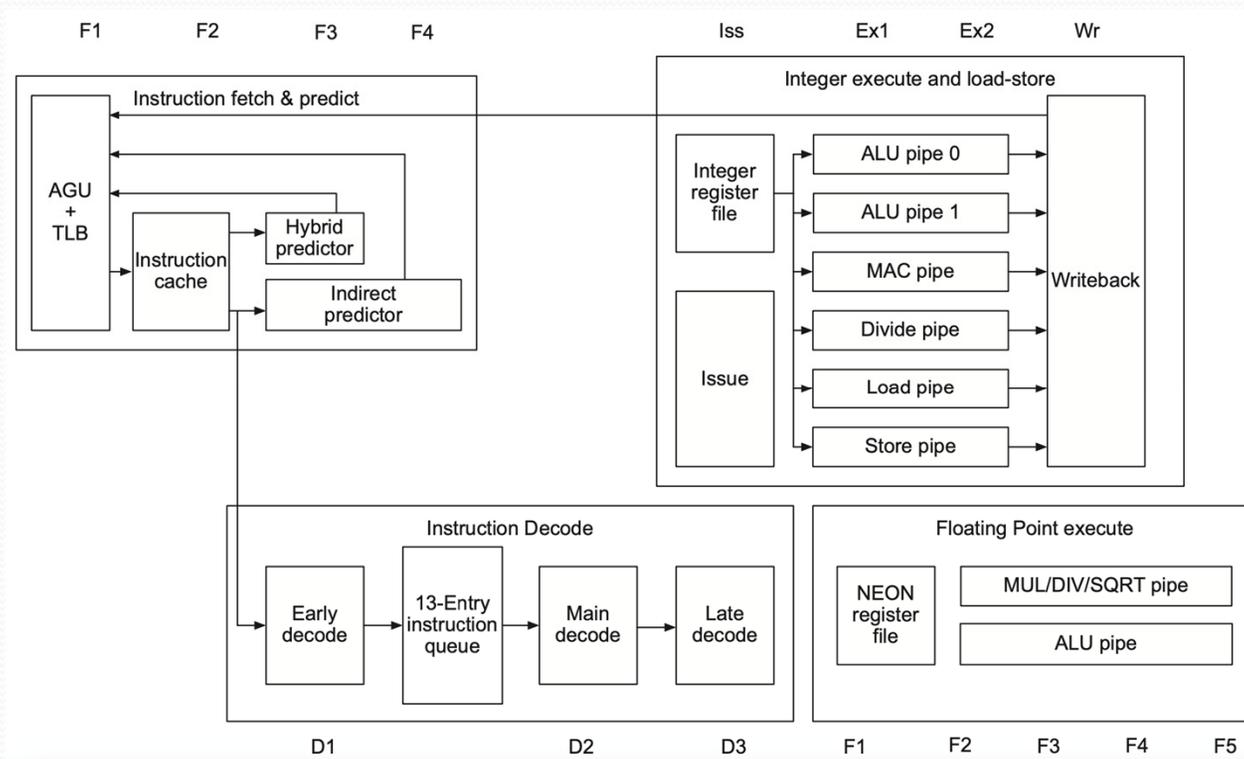
Explicación

- La desventaja de incrementar el paralelismo mediante emisión múltiple dinámica y especulación es la potencial ineficiencia de energía.
- La tendencia es usar pipelines más cortos que antes y buscar diseños multicore.

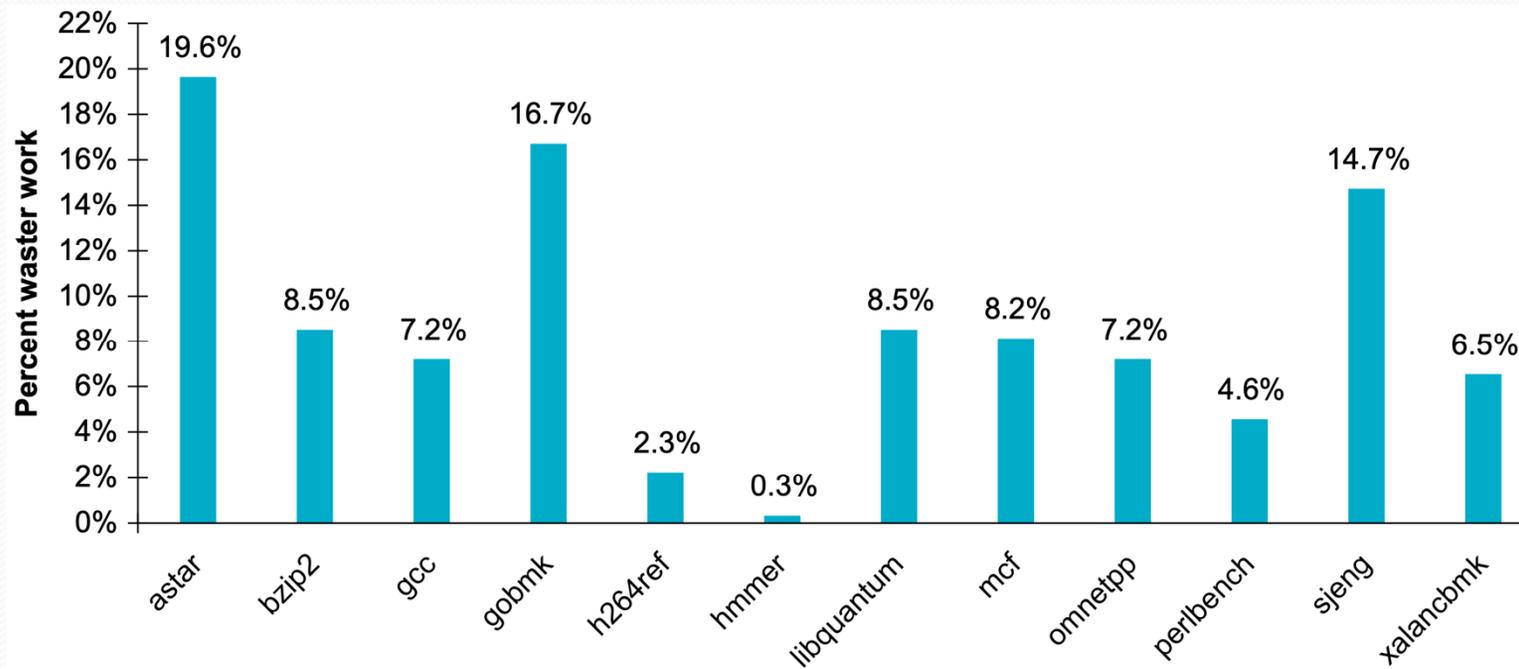
ARM Cortex-A53

- Lanzado en 2012.
- Se utiliza en varias tabletas y teléfonos celulares.
- Superescalar con scheduling estático.
- Emisión dinámica.
- Grado 2 (emite 2 instrucciones a la vez).
- Pipeline:
 - Instrucciones enteras no brincos: 8 etapas.
 - Instrucciones de punto flotante: 10 etapas.
- Castigo por mala predicción: 8 ciclos.

Pipeline ARM Cortex-A53



Desperdicio por malas predicciones

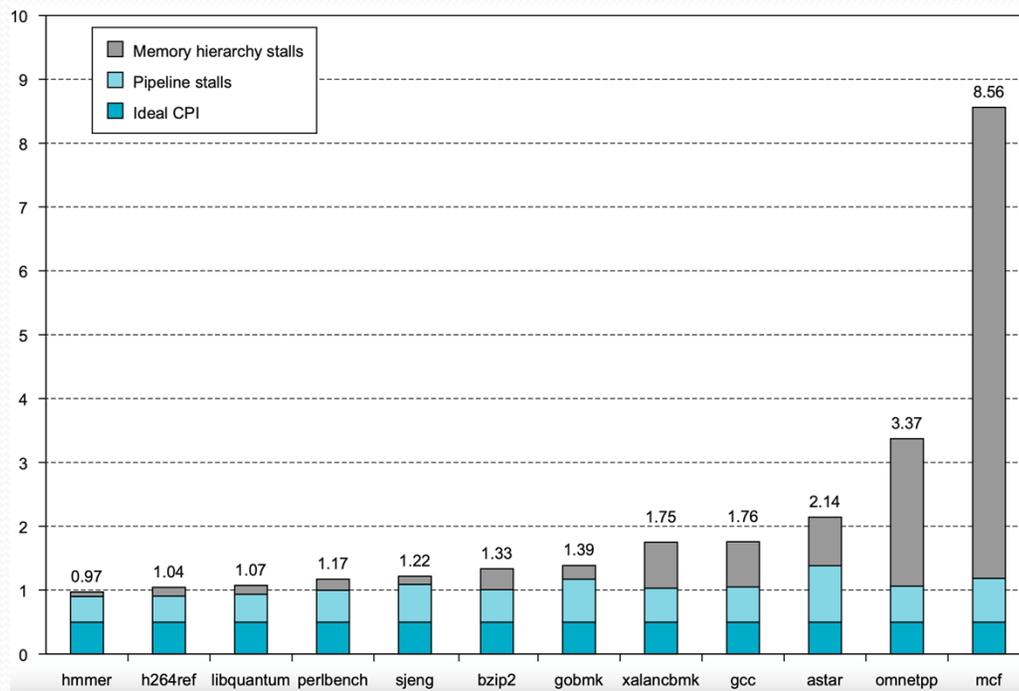


Fuente: COD-HSI-RISC, p. 356

Rendimiento

- CPI ideal: 0.5.
- Las detenciones o stalls se producen por:
 1. Peligros funcionales. Las dos instrucciones dentro del paquete de emisión utilizan la misma unidad funcional.
 2. Peligros de datos. Dependencias verdaderas en el paquete de emisión.
 3. Peligros de control. Brincos mal adivinados.
- Se espera que el compilador prevenga o reduzca los peligros funcionales y los de datos.

Composición del CPI



Fuente: COD-HSI-RISC, p. 357

Intel Core i7 6700

- Tiene 4 cores.
- Emisión dinámica.
- Scheduling dinámico (ejecución fuera de orden con especulación).
- El ISA es CISC.
- Internamente las instrucciones se traducen en micro-operaciones.
- Pipeline de 14 etapas.
- Pueden entrar al pipeline hasta 6 micro-operaciones por ciclo.
- Utiliza un ROB para resolver antidependencias y especulaciones incorrectas.

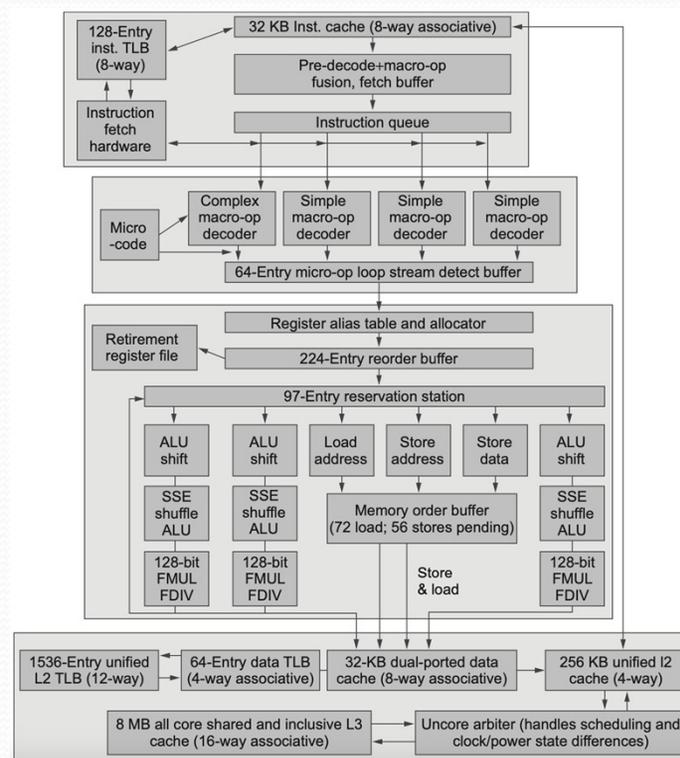
Pipeline

- Etapas principales:
- Leer la instrucción (instruction fetch). No es trivial porque las instrucciones en x86 pueden ocupar entre 1 y 17 bytes.
- Traducción de instrucciones a micro-operaciones.
- Emisión de micro-operaciones (hasta 4 por ciclo).
- Estación de reserva centralizada con 6 unidades funcionales.
- Hasta 6 micro-operaciones pueden ser despachadas a las unidades funcionales en cada ciclo.

Pipeline

- Los resultados de cada unidad funcional se envían a la estación de reserva (por si hay dependencias) y a la unidad de retiro de registros donde se actualiza su estado cuando se sabe que la instrucción no es especulada. Además, su entrada en el ROB se marca como completada.
- Cuando una o más instrucciones al frente del ROB se han marcado como completas, se ejecutan las escrituras pendientes en la unidad de retiro de registros y las instrucciones se eliminan del ROB.

Intel Core i7 6700



Intel Core i7 6700

- CPI en el benchmark SPEC CPUint2006: 0.71.
- Castigo por mala predicción: 17 ciclos.
- Castigo por falla caché L1: 10 ciclos.
- Castigo por falla caché L2: 30 – 35 ciclos.
- Castigo por falla caché L3: 130 – 135 ciclos.

Comparación A53 vs i7

- Tasas de fallas por malas predicciones en SPEC2006: 2.3% (i7) vs 3.9% (A53).
- CPI en SPEC2006: 0.64 (i7) vs 1.36 (A53).
- Velocidad de reloj: 3.54 GHz (i7) vs 1.3 GHz (A53).
- Tiempo promedio por instrucción:
 - i7: $\frac{1 \times 0.64}{3.54 \times 10^9} = 0.18ns$
 - A53: $\frac{1 \times 1.36}{1.3 \times 10^9} = 1.05ns$
- i7 es 5 veces más rápida que A53.

Comparación A53 vs i7

- i7 consume 200 veces más energía que A53.