CS-200
Computer Architecture
—
Part 4g. Instruction Level Parallelism
Intel x86 and ARM

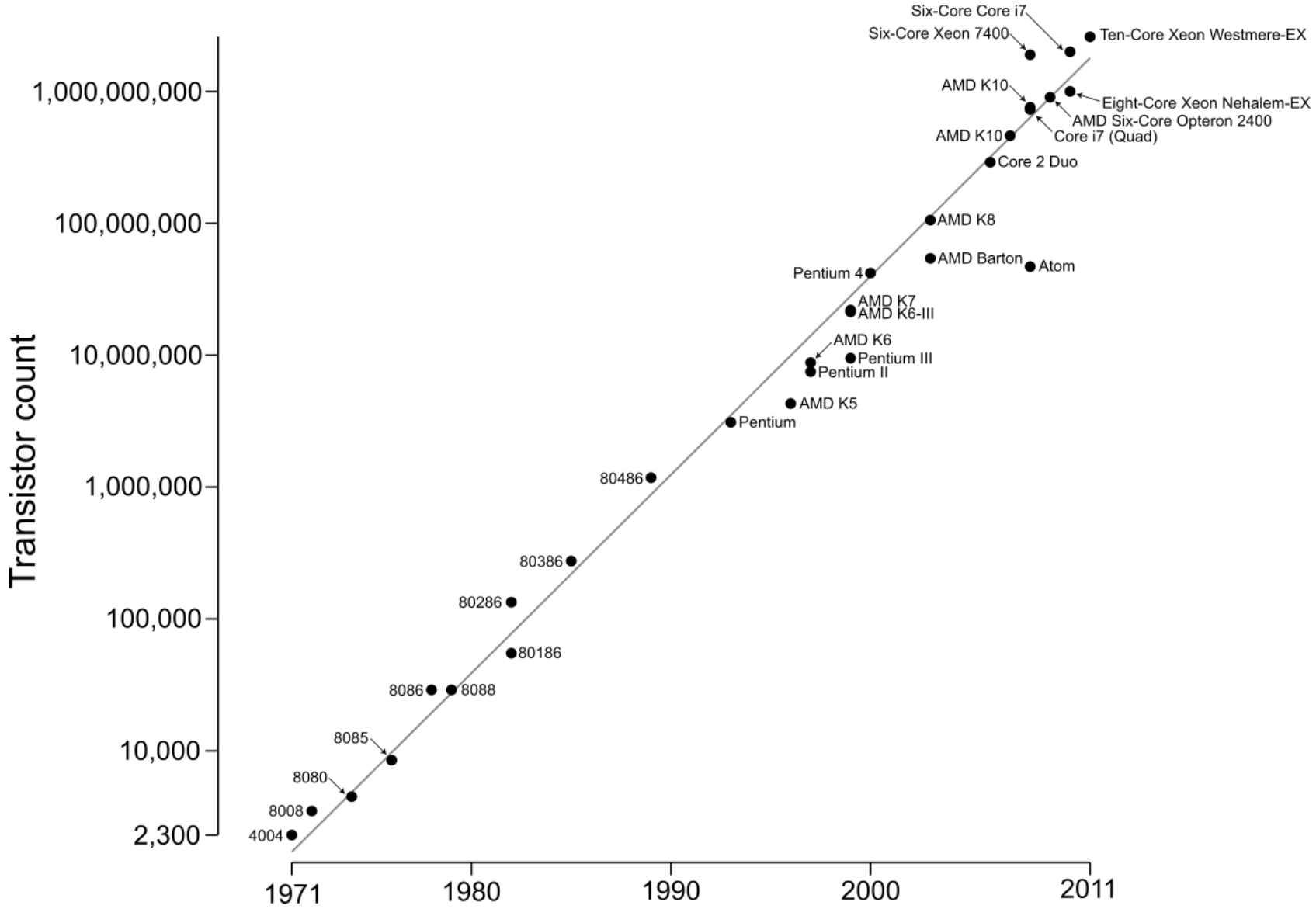
Paolo Ienne
<paolo.ienne@epfl.ch>

Some slides by Prof. Sanchez (EPFL)

Intel Processors

Processor	Date	f (MHz)	Trans.	Features
4004	4/71	0.108	2300	First µP
8008	4/72	0.108	3500	First 8-bit µP
8080	4/74	2	6000	Popular 8-bit
8086	6/78	5-10	29k	First 16-bit µP; 20-bit addressing
8088	6/79	5-8	29k	Simpler; IBM PC
80286	2/82	8-12	134k	Protected mode, 24-bit addressing
80386	10/85	16-33	275k	32-bit (**x86**)
80486	4/89	25-100	1.2M	Pipelined (5-stage); cache
Pentium	3/93	60-233	3.1M	Superscalar, dual pipeline
PentiumPro	3/95	150-200	5.5M	Out-of-order; L2 cache
Pentium II	5/97	233-400	7.5M	MMX (SIMD instructions)
Pentium III	3/99	450-1400	9.5-28M	SSE (incl. SIMD-FP); 10-stage pipeline
Pentium 4	12/00	1300-2200	42M	SSE2 (128-bit); TC; 20-stage pipeline
NetBurst	04/04	800-3800	376M	64-bit (**x86-64**); SSE3; HT; 31-stage p.
Core	06/06	up to 3300	105M	SSSE3; SSE4; 12-14-stage pipeline
Nehalem	11/08	up to 3600	731M	20-24-stage pipeline
Sandy Br.	01/11	up to 4000		AVX (256-bit and 3-op); 14-19-stage p.
Haswell	06/13	up to 4400		AVX2
Skylake	08/15	up to 3700		AVX-512 (512-bit)

Intel Processors



Some High-End Processors in 2009

Processor	Intel 1-core Xeon	AMD 1-core Opteron 854	Intel 2-core Xeon X5270 [1]	AMD 2-core Opteron 8224SE	Intel 4-core Xeon X7350 [2]	AMD 4-core Opteron 8360SE [3]	Intel 6-core Xeon X7460 [4]
Bit-width	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit
Cores/chip x Threads/core	1 x 2	1 x 1	2 x 1	2 x 1	4 x 1	4 x 1	6 x 1
Clock Rate	3.80GHz	2.80GHz	3.50GHz	3.20GHz	2.93GHz	2.50GHz	2.67GHz
Cache: L1-L2-L3 - I/D or Unified	12K/16K - 2M - N/A	64K/64K - 1M - N/A	2 x 32K/32K - 6M - NA	2 x 64K/64K - 2 x 1M - N/A	4 x 32K/32K - 2 x 4M - N/A	4 x 64K/64K - 4 x 512K - 2M	6 x 32K/32K - 3 x 3M - 16M
Execution Rate/Core	3 Instructions	3 Instructions	1 Complex + 3 Simple	3 Instructions	1 Complex + 3 Simple	3 Instructions	1 Complex + 3 Simple
Pipeline Stages	31	12 int / 17 fp	14	12 int / 17 fp	14	12 int / 17 fp	14
Out of Order	126	72	96	72	96	72	96
Memory Bus	800MHz	6.4GB/s	1333MHz	10.6GB/s	1066MHz	10.6GB/s	1064MHz
Package	LGA-775	uPGA 940	LGA-771	LGA-1207	LGA-771	LGA-1207	LGA-771
IC Process	90nm 7M	90nm 9M	45nm	90nm 9M	65nm 8M	65nm 11M	45nm
Die Size	109mm^2	106 mm^2	107mm^2	227mm^2	2 x 143mm^2	283mm^2	503mm^2
Transistors	169M	120M	410M	233M	2 x 291M	463M	1900M
List Price (Intro)	$903	$1,514	$1,172	$2,149	$2,301	$2,149	$2,729
Power (Max)	110W	93W	80W	120W	130W	105W	130W
Availability	3Q05	3Q05	3Q08	3Q07	3Q07	2Q08	4Q08
Scalability	1–2 Chips	2–4 Chips	1–2 Chips	1–4 Chips	1–4 Chips	2–4 Chips	1–4 Chips
SPECint/fp2006 [Cores]	11.4/11.7 [2]	11.2/12.1 [2]	26.5^x/25.5^x [4]	14.1/14.2 [8]	21.7^y/18.9^y [16]	14.4^x/18.5^x [8]	22.0^x/22.3^x [24]
SPECint/fp2006_rate [Cores]	20.9/18.8 [2]	41.4/45.6 [4]	84.9^x/57.7^x [4]	105/96.7 [8]	184^y/108 [16]	170^x/156^x [16]	274^x/142^x [24]
x86 Codename	Irwindale	Athens	Wolfdale	Santa Rosa	Tigerton	Barcelona	Dunnington
Microarchitecture	Netburst	K8	Core	K8	Core	K10	Core

Processor	Intel Itanium 2 9050	Intel Itanium 9150M	IBM POWER5+	IBM POWER6	Fujitsu SPARC64 VI	Fujitsu SPARC64 VII	Sun UltraSPARC T2+
Bit-width	64-bit	64-bit	64-bit	64-bit	64-bit	64-bit	64-bit
Cores/Chip x Threads/Core	2 x 2	2 x 2	2 x 2	2 x 2	2 x 2	4 x 2	8 x 8
Clock Rate	1.60GHz	1.67GHz	2.20GHz	5.00GHz	2.40GHz	2.52GHz	1.40GHz
Cache: L1-L2-L3 - I/D or Unified	2 x 16K/16K - 1M/256K - 12M(on)	2 x 16K/16K - 1M/256K - 12M(on)	2 x 64K/32K 1.92M - 36M(off)	2 x 64K/64K - 2 x 4M - 32M(off)	2 x 128K/128K - 6M - N/A	4 x 64K/64K - 6M - N/A	8 x 8K/16K - 4M - NA
Execution Rate/Core	6 Issue	6 Issue	5 Issue	7 Issue	4 Issue	4 Issue	16 Issue
Pipeline Stages	8	8	15	13	15	15	8 int / 12 fp
Out of Order	None	None	200	Limited	64	64	None
Memory Bus	8.5GB/s	10.6GB/s	12.8GB/s	75GB/s	8GB/s	8GB/s	42.7GB/s
Package	mPGA-700	mPGA-700	MCM-5370 Pins	N/A	412 I/O Pins	412 I/O Pins	1831 Pins
IC Process	90nm 7M	90nm 7M	90nm 10M	65nm 10M	90nm 10M	65nm 11M	65nm
Die Size	596mm^2	596mm^2	245mm^2	341mm^2	421mm^2	400mm^2	342mm^2
Transistors	1.72B	1.72B	276M	790M	540M	600M	503M
List Price (Intro)	$3,692	$3,692	N/A	N/A	N/A	N/A	N/A
Power (Max)	104W	104W	100W	>100W	120W	135W	95W
Availability	3Q06	4Q07	4Q05	2Q08	2Q07	3Q08	2Q08
Scalability	1–64 Chips	8–128 Chips	1–32 Chips	2–32 Chips	4–64 Chips	4–64 Chips	2 Chips
SPECint/fp2006 [Cores]	14.5/17.3 [2]	N/A	10.5/12.9 [1]	15.8*/20.1 [1]	9.7/21.7^y [32]	10.5^x/25.0^x [64]	N/A
SPECint/fp2006_rate [Cores]	1534/1671 [128]	2893/N/A [256]	197/229 [16]	1837^x/1822 [64]	1111/1160 [128]	2088^x/1861^x [256]	142/111 [16]
Architecture Status	Inactive	Active	Inactive	Active	Inactive	Active	Active

All SPEC scores are base. * Score measured at 4.20GHz (not 5.00GHz).

= x86

= IA-64

Source: Microprocessor Report, © Cahners 2009

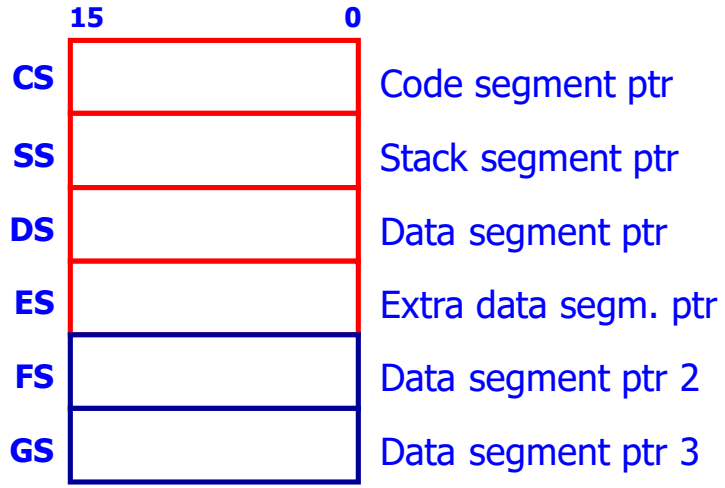
Legacy x86 Features

- Very **small number of registers**, partly dedicated or **specialised**
- Natively 16-bit, extended to 32 in successive steps requiring backward compatibility (e.g., 3 modes for address generation)
- **Highly variable instruction length** and encoding (1 to 17 bytes in original x86, prefixes, postfixes, etc.)
- **CISC** instruction set

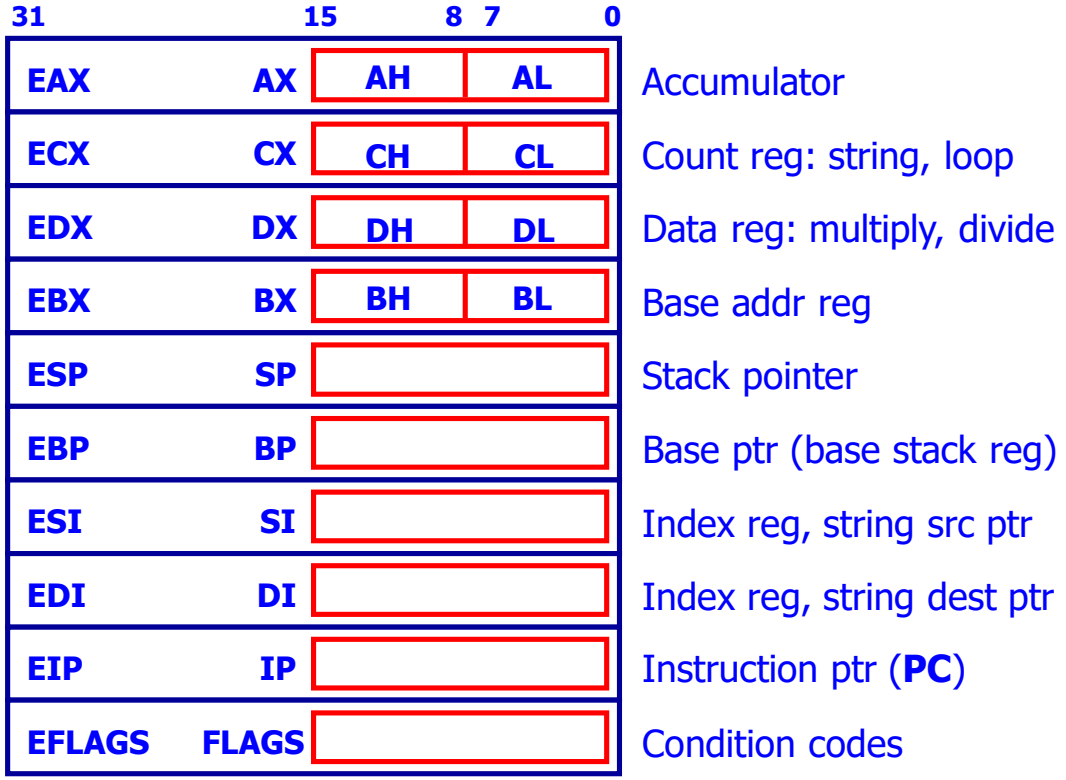
Registers (I)

- Very small number of general purpose registers (approx. 4 integer plus 8 FP—not shown, versus 32+32 typ. RISC)

Segment Registers



General Registers + PC + Flags



◆ → **80386**

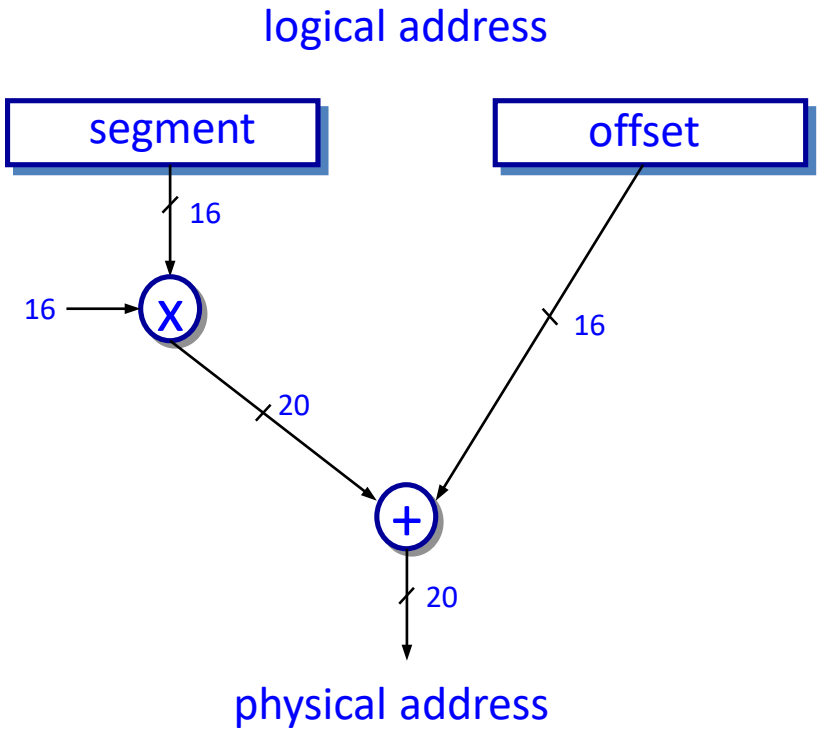
◆ → **8086**

Registers (II)

- Small number of registers makes spilling more frequent
- Advanced compiler techniques (e.g., loop unrolling) increase register pressure
- Partial specialization of the registers makes **effective compiler use difficult**

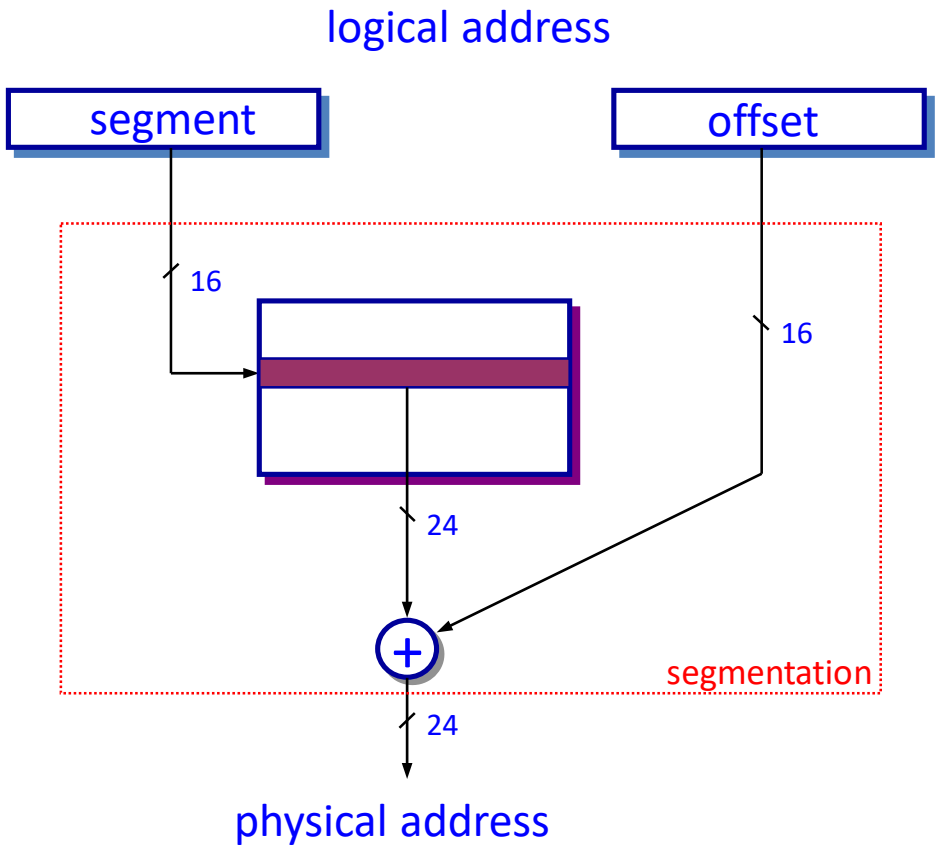
Memory Addressing (I)

- Real Mode (8086)



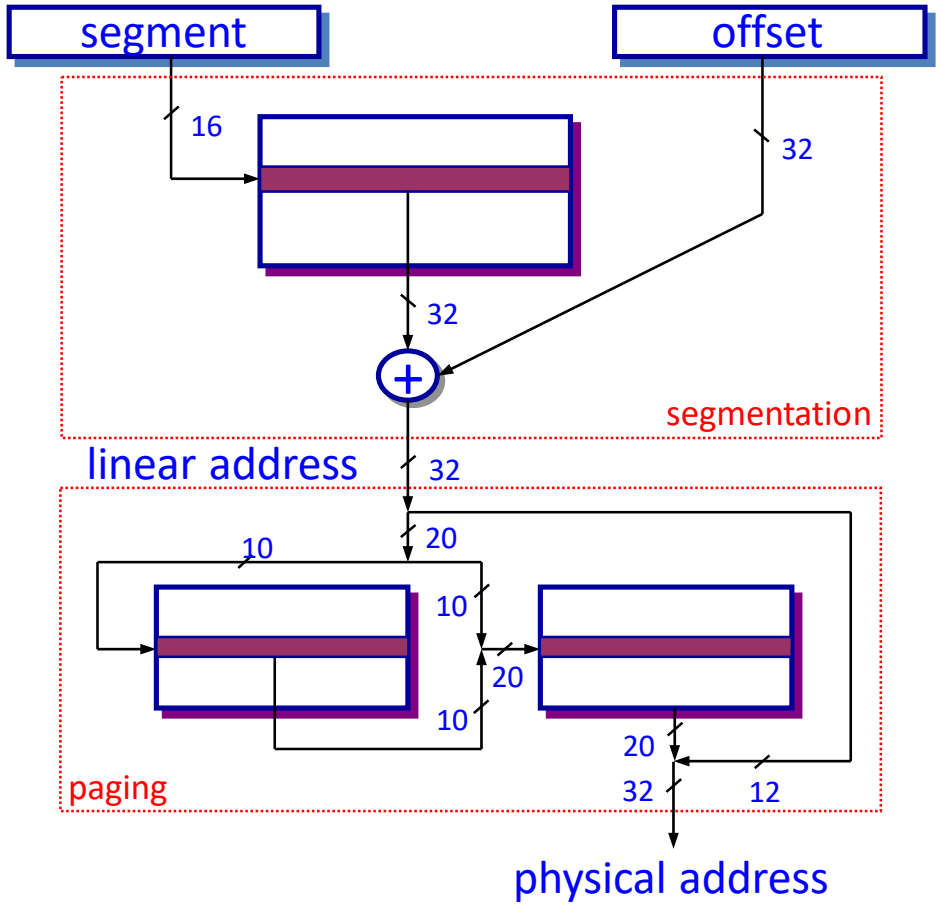
Memory Addressing (II)

- Protected Mode (80286)



Memory Addressing (III)

- Protected Mode (80386, 80486, and Pentium)



Operand Types

- **Not** a Load/Store architecture

Source 1 = Destination	Source 2
Register	Register
Register	Immediate
Register	Memory
Memory	Register
Memory	Immediate

- Immediate values can be on 8, 16, or 32 bits

Addressing Modes (I)

- **Absolute**
- **Register indirect** → [reg]
  - 16-bit registers: BX, SI, DI
  - 32-bit registers: EAX, ECX, EDX, EBX, ESI, EDI
- **Displacement** → [reg + *displacement*]
  - 16-bit registers: BP, BX, SI, DI
  - 32-bit registers: EAX, ECX, EDX, EBX, ESI, EDI
  - Displacement on 8, 16, or 32 bits
- **Indexed** → [base reg + reg]
  - 16-bit registers: BX+SI, BX+DI, BP+SI, BP+DI

Addressing Modes (II)

- **Indexed with displacement**
  → [base reg + reg + *displacement*]
  - Same registers as in mode indexed
- **Scaled indexed** → [base reg + 2^{scale} x reg]
  - Only in 32-bit mode
  - Scale is 0, 1, 2, or 3
  - Index register can be any of the basic registers (except ESP)
  - Base register can be any of the basic registers
- **Scaled indexed with displacement**
  → [base reg + 2^{scale} x reg + *displacement*]

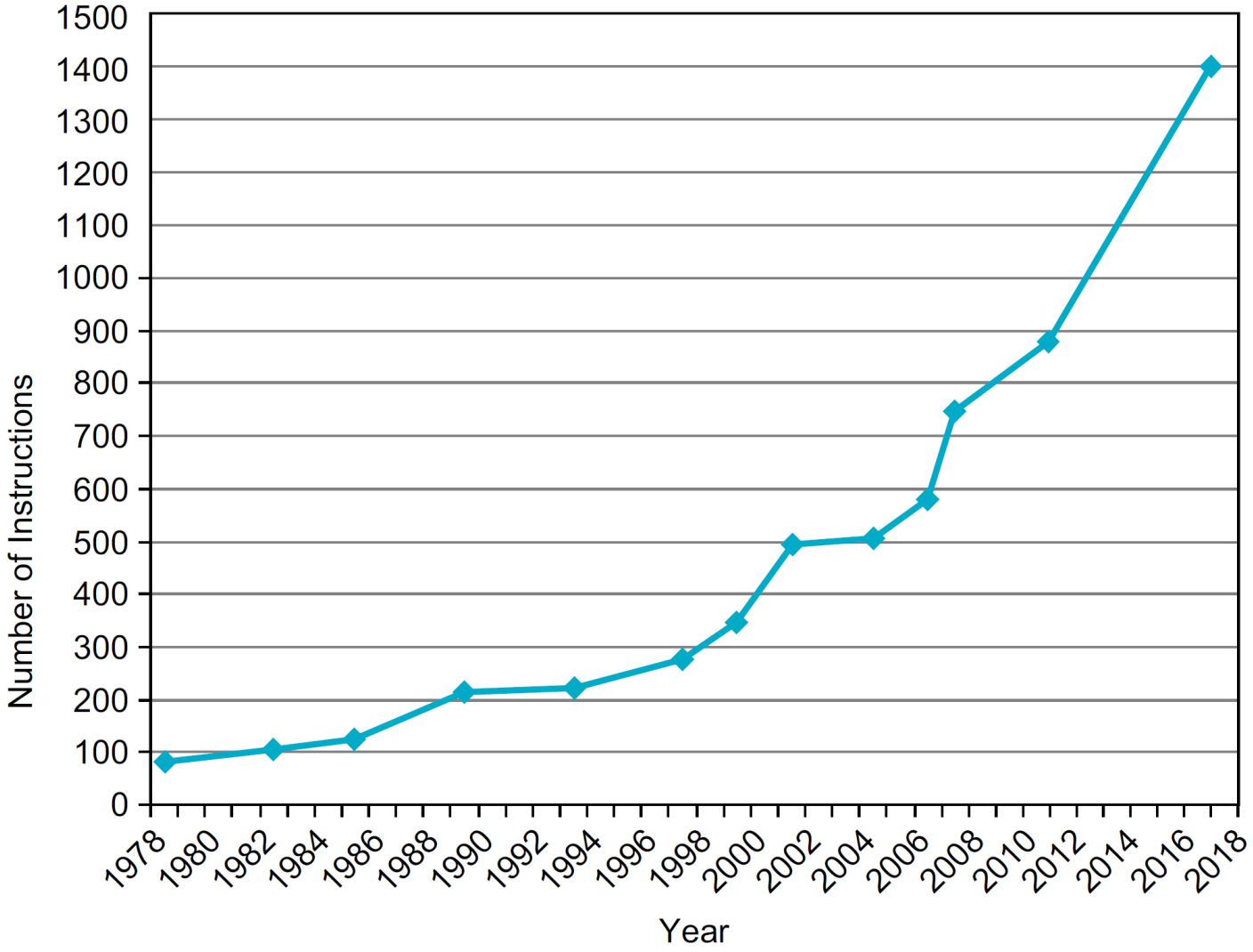
Address Segment (Legacy)

- For every indirect addressing (e.g., [reg]) the appropriate segment would be needed to complete the address
- Default:
  - References to instructions (IP) use CS (code segment register)
  - References to stack (BP or SP) use SS (stack segment register)
  - All other references use DS (data segment register)
- A **one-byte instruction prefix** can modify the default

A Growing ISA

- **80386** (1985): 32-bit registers, flat memory model, paging, and virtual memory
- **80486** (1989): x87 floating-point coprocessor instructions integrated
- **MMX** (1997): SIMD instructions for multimedia (8-bit, 16-bit, 32-bit integer)
- **SSE** (1999): 128-bit SIMD for floating-point, integer, and memory operations
- **SSE2** (2001): Double-precision floating-point and integer operations
- **SSE3**/**SSSE3** (2004-2006): Minor enhancements (e.g., horizontal add/subtract)
- **SSE4** (2007-2008): Adds text processing (SSE4.2) and improved vector processing
- **AVX** (2011): 256-bit SIMD, non-destructive 3-operand form, VEX prefix
- **AVX2** (2013): Adds integer operations, gathers, and FMA (Fused Multiply-Add)
- **AVX-512** (2016): 512-bit SIMD, mask registers, expanded precision operations
- **x86-64** (2003, **AMD**): 64-bit, more registers (R8–R15), RIP-relative addressing

Number of x86 Instructions over Time



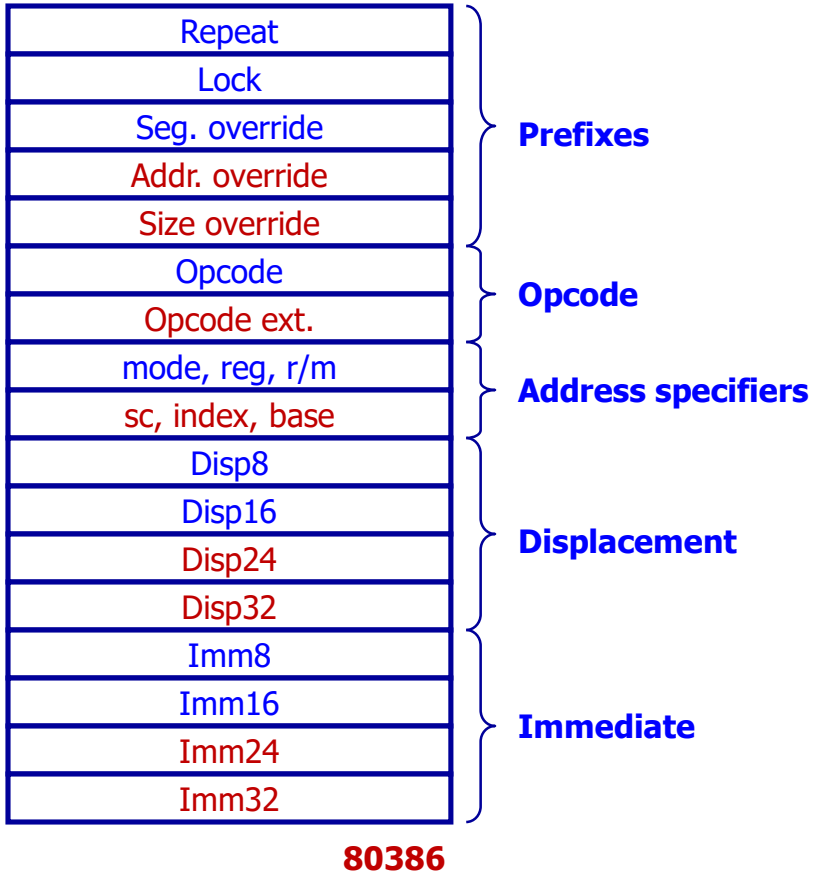
Source: COD RISC-V, © Morgan Kauffman 2021

Instruction Examples

JE addr	if equal(CC) then IP ← addr (IP-128 ≤ addr < IP+128)
JMP addr	IP ← addr
CALL addr,seg	SP ← SP-2; Mem[SS:SP] ← IP+5
	SP ← SP-2; Mem[SS:SP] ← CS
	IP ← addr; CS ← seg
MOVW BX,[DI+45]	BX ← Mem[DS:DI+45]
PUSH SI	SP ← SP-2
	Mem[SS:SP] ← SI
POP DI	DI ← Mem[SS:SP]
	SP ← SP+2
ADD AX,#6765	AX ← AX+6765
TEST DX,#42	set CC flags with (DX and 42)
MOVSB	Mem[ES:DI] ← Mem[DS:SI]
	DI ← DI+1
	SI ← SI+1

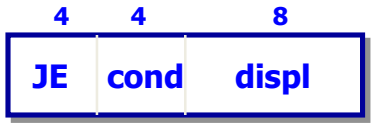
Instruction Encoding

- One instruction coded on 1 to 17 bytes in original x86
- Several types of modifiers/prefixes
- Two combinations of constants of variable length
  - Immediate and Displacement
  - 8, 16, and 32-bit
- Opcode “lost” and only moderately orthogonal



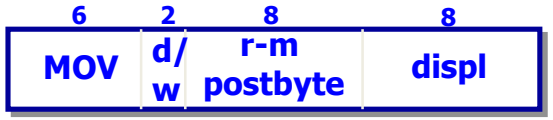
Examples of Instruction Encoding

JE PC+3657

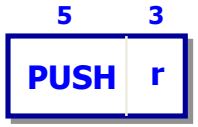


CALLF

MOV BX,[DI+45]

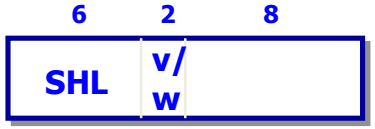


PUSH SI

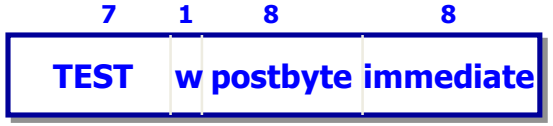


ADD AX,#6765

SHL BX,1



TEST DX,#42



Improved Encoding and Generality

- Legacy **MUL** instruction
  - Encoded as `0xF7 E3`

```
MOV  RAX, 5          ; Load first operand into RAX (implicit source)
MOV  RBX, 10         ; Load second operand into RBX
MUL  RBX             ; Multiply RAX * RBX -> result in RDX:RAX
```

  - Multiplier = result, implicit; multiplicand specified in the second byte
- Modern **MULX** instruction
  - Encoded as `0xF2 0F 38 F6`

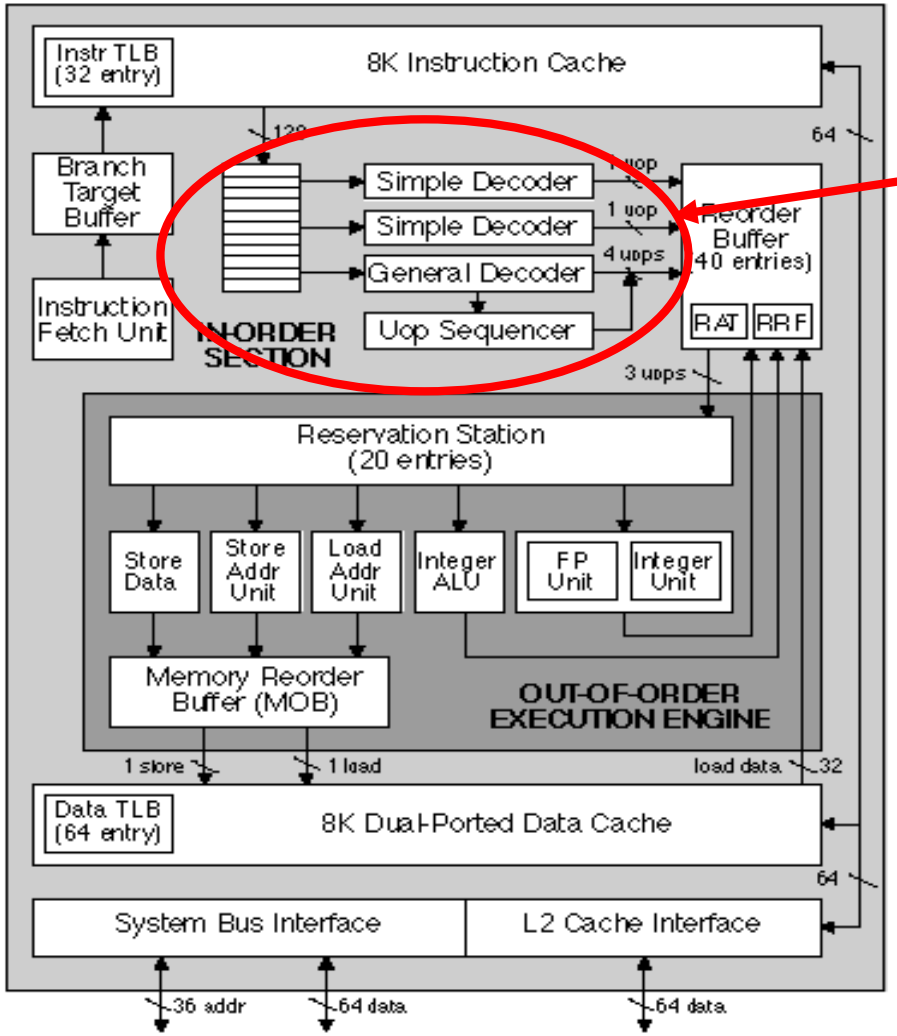

```
MOV  RAX, 5          ; Load first operand (explicitly RAX, required)
MOV  RBX, 10         ; Load second operand into RBX
MULX RDX, RCX, RBX   ; Multiply RBX * RAX -> RCX = low, RDX = high
```

  - Multiplier implicit; multiplicand and result arbitrary

1995: PentiumPro (P6)
A Superscalar x86 CISC?

- How to adapt the superscalar ideas to fit such an irregular architecture?
  - Complexity of decoding is huge
  - Parallel decoding of instructions is tough due to an encoding strongly variable in lenght
  - Instructions mix memory operations with computations
  - Too few registers

PentiumPro Microarchitecture: Out-of-order CISC Execution



Decoder from x86 to P6 uops: A "CISC-to-RISC converter"…

Classic RISC out-of-order engine

Source: Microprocessor Report, © Cahners 2001

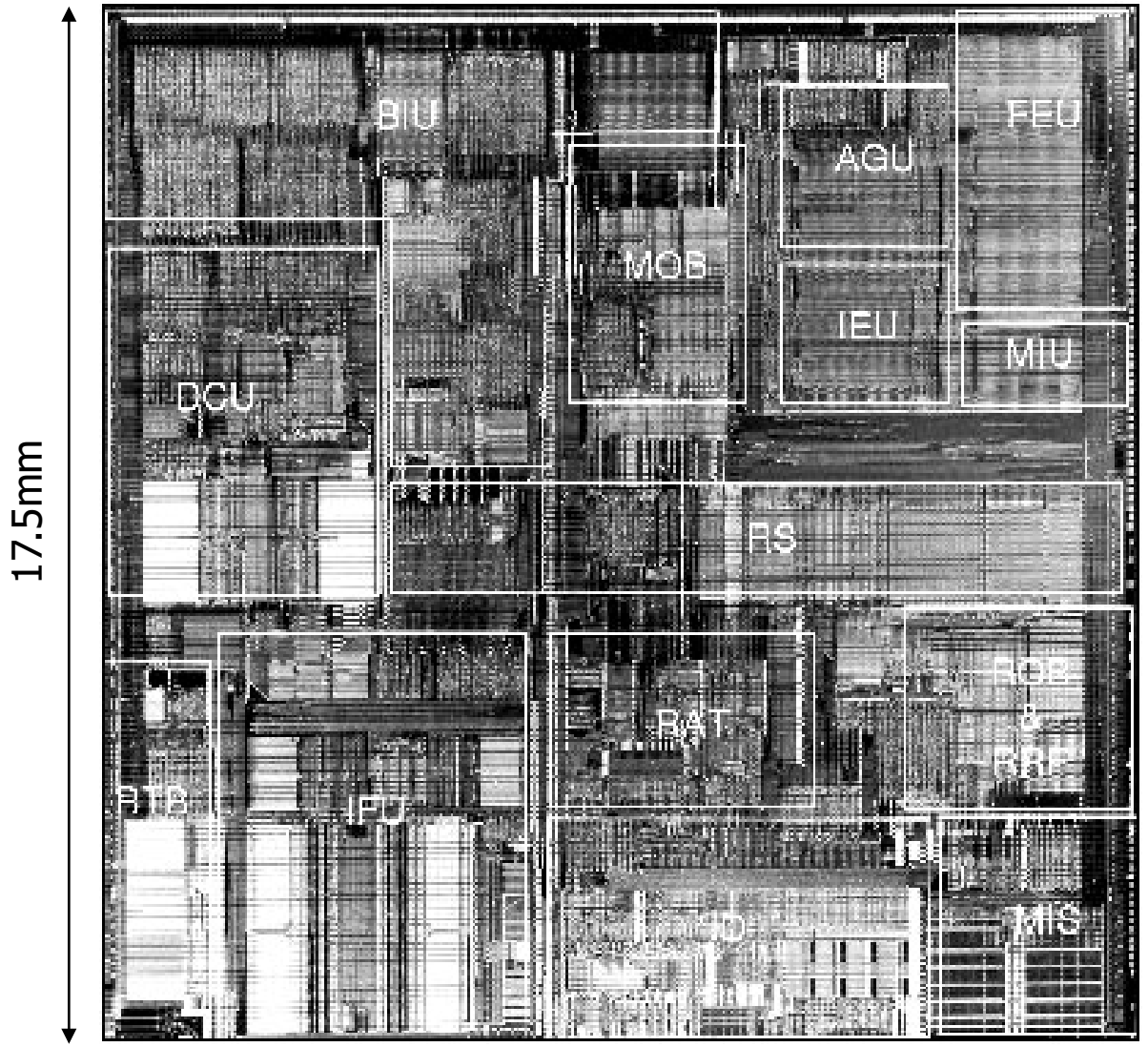
PentiumPro In-Order Section

- Converts every x86 instruction into one or more internal RISC-like 118-bit instructions (micro-operations or uops); on average 1 instruction = 1.5-2.0 uops
- Three decoders and a sequencer work in parallel to perform the conversion
  - Two highest priority simple decoders intercept register-register operations (1 instr. → 1 uop)
  - A low priority general decoder handles all other basic operations (1 instr. → 4 uops)
  - A sequencer is used by the general decoder for very complex operations (1 instr → several groups of 4 uops)
- Reorder Buffer (ROB) implements renaming and commits uops in program order to the Real Register File (RRF)

PentiumPro Out-of-order Section

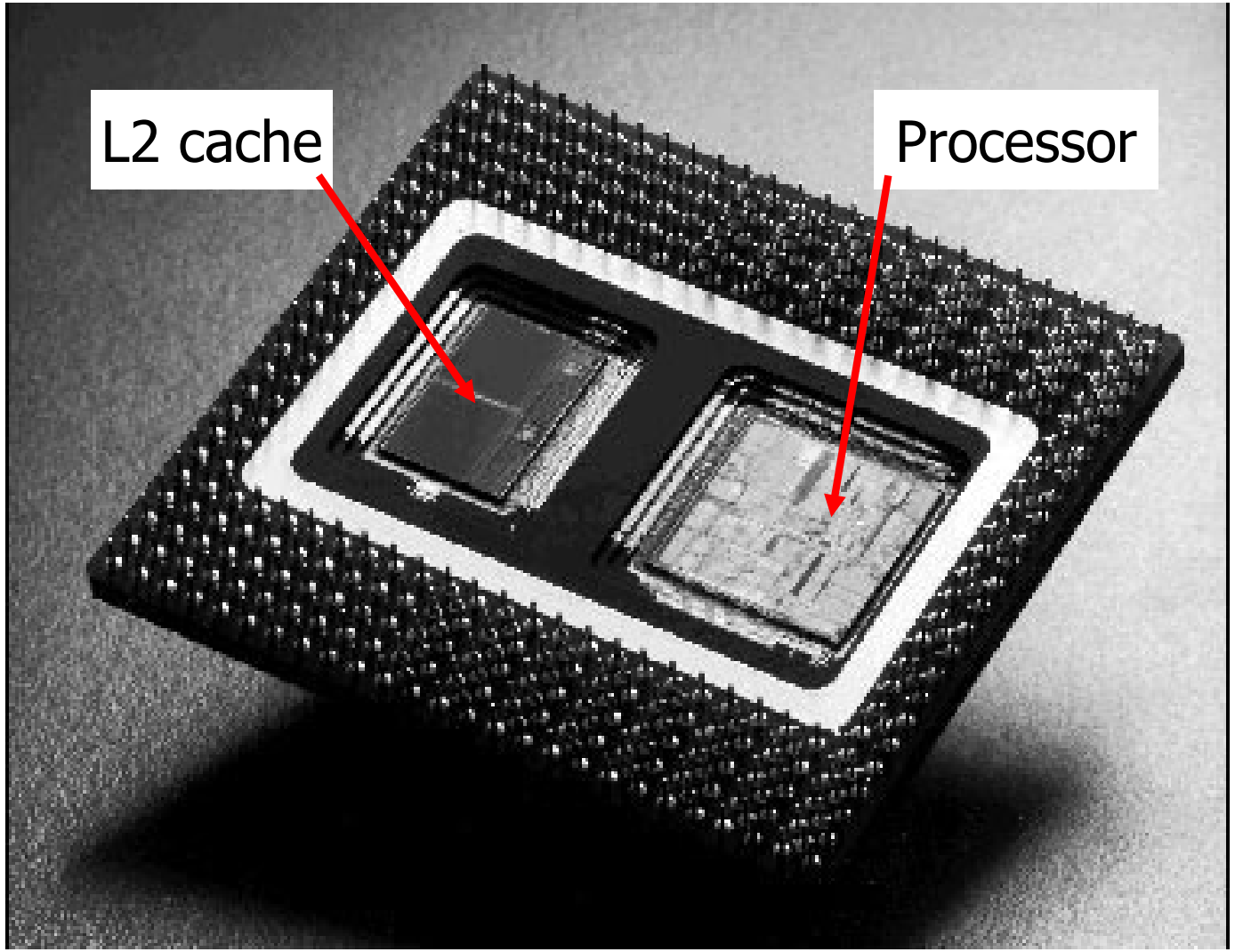
- Superscalar very similar to the general model studied in previous lessons
- Up to 20 uops wait in the Reservation Stations until the operands are all available
- A maximum of 5 uops can be issued per cycle: a generic calculation (int or FP), a simple integer (no shift, mul, nor div), a load, a store address, and a store data
- A Memory Reorder Buffer (MOB) reorders memory accesses and waits for D-cache availability

PentiumPro Die
300mm² 0.5µm 4ML BiCMOS



Source: Microprocessor Report, © Cahners 2001

PentiumPro Package

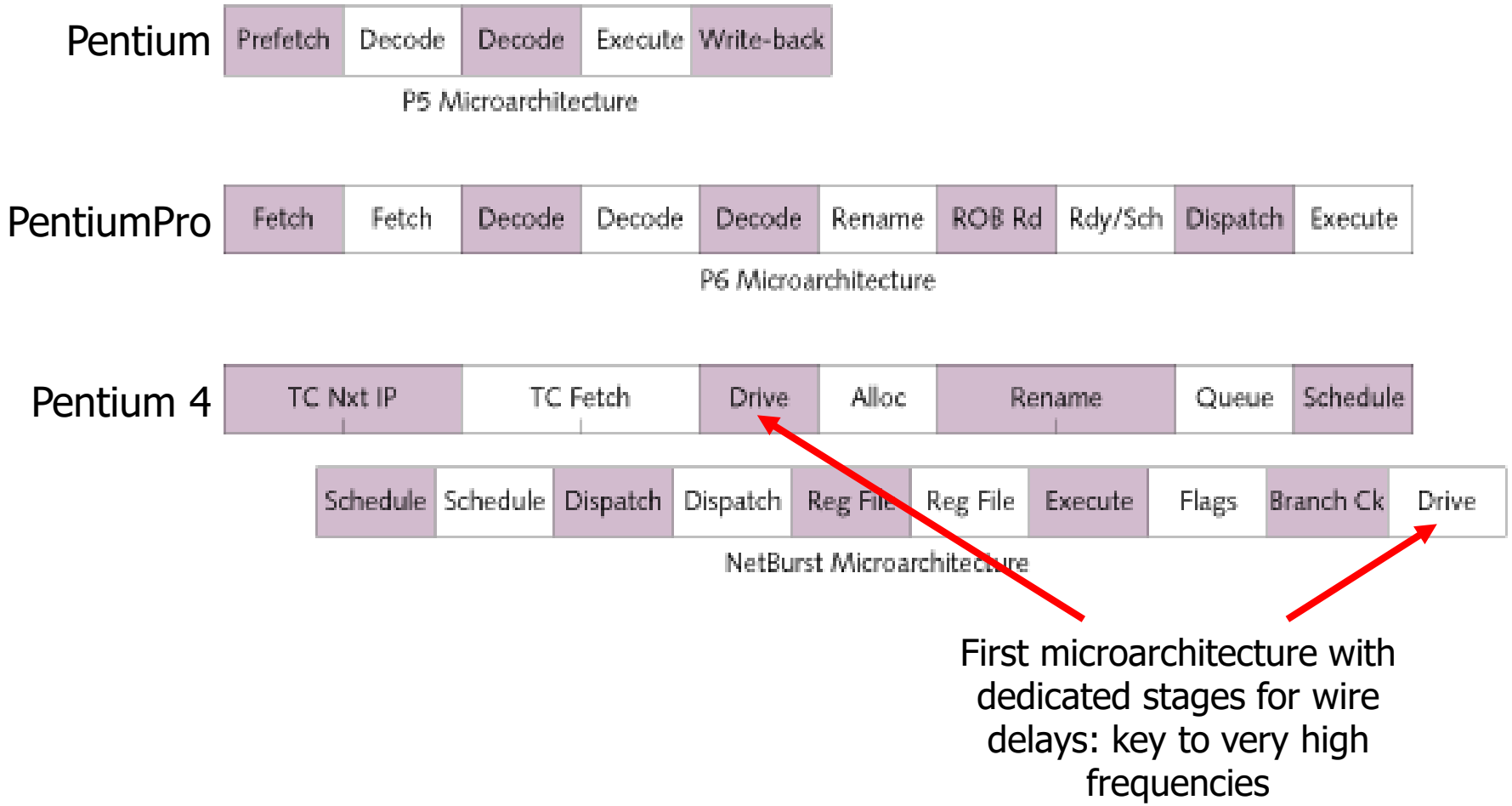


Source: Microprocessor Report, © Cahners 2001

2000: Pentium 4
Processors for the Multi-GHz Era

- Isn’t Pentium dead in favour of IA-64 and Itanium? Clearly not…
- How to modify Pentium III to achieve way less than 1ns of cycle time?
  - Pipeline expansion? (Pentium III: 10 stages)
  - Wire propagation time becomes very tangible compared to computation
  - uop decoding very heavy

Evolution of Pentium Pipeline: From 5 to 20 Stages!

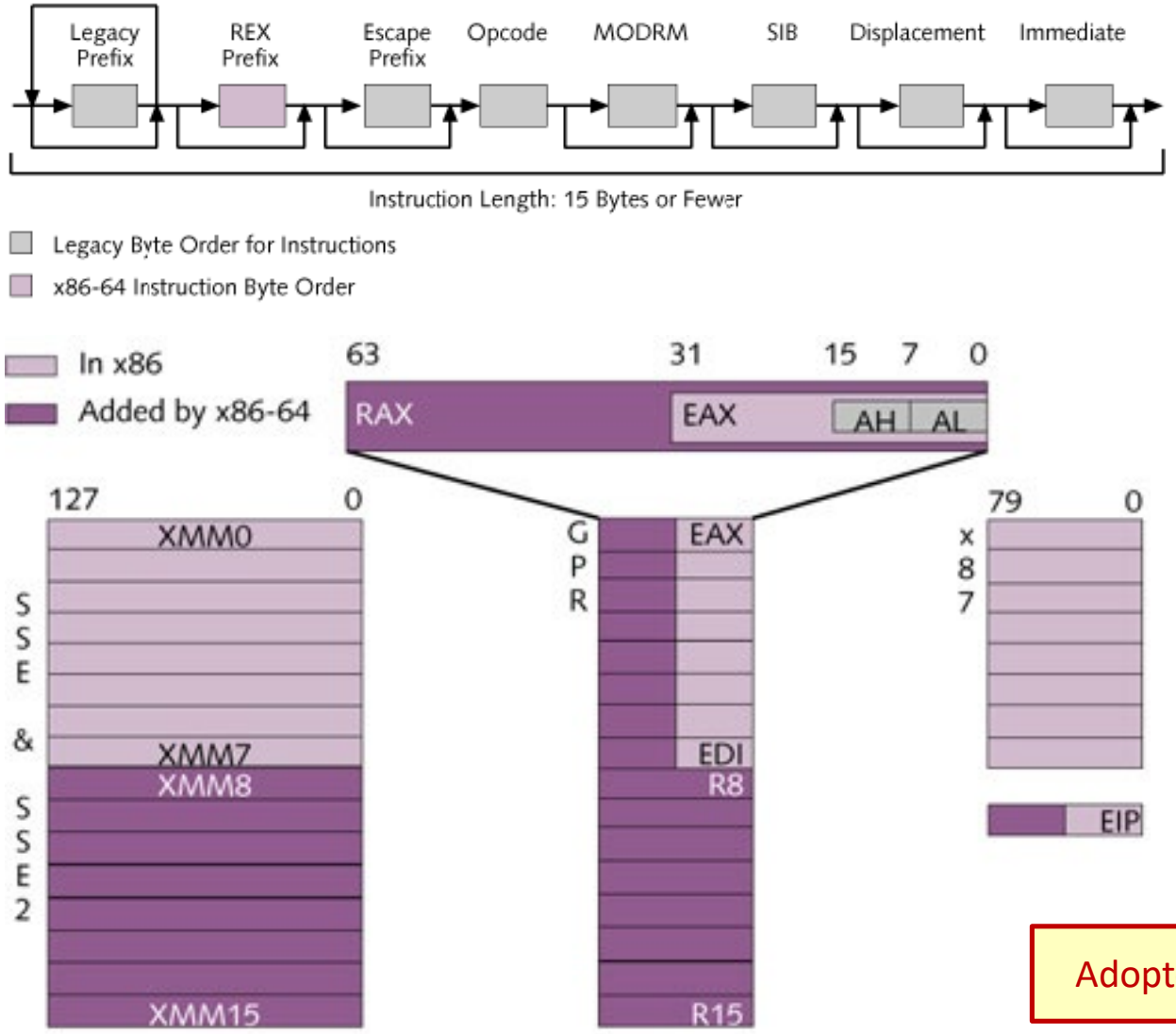


Source: Microprocessor Report, © Cahners 2001

Some Pentium 4 Characteristics

- **Trace caches** to memorize approx. 12,000 recent uops—sort of L0 cache to avoid x86 instruction decoding from the main loop
- Approx. 126 uops can be in-flight at one time (3 times more than Pentium III)
- **Data speculation** to execute a potentially dependent load before a store: if the dependence was real, the load is squashed and replayed
- P4 ALUs can perform simple operations in half clock-cycle, to sustain throughput
  - Two dependent operations can be scheduled in the same cycle

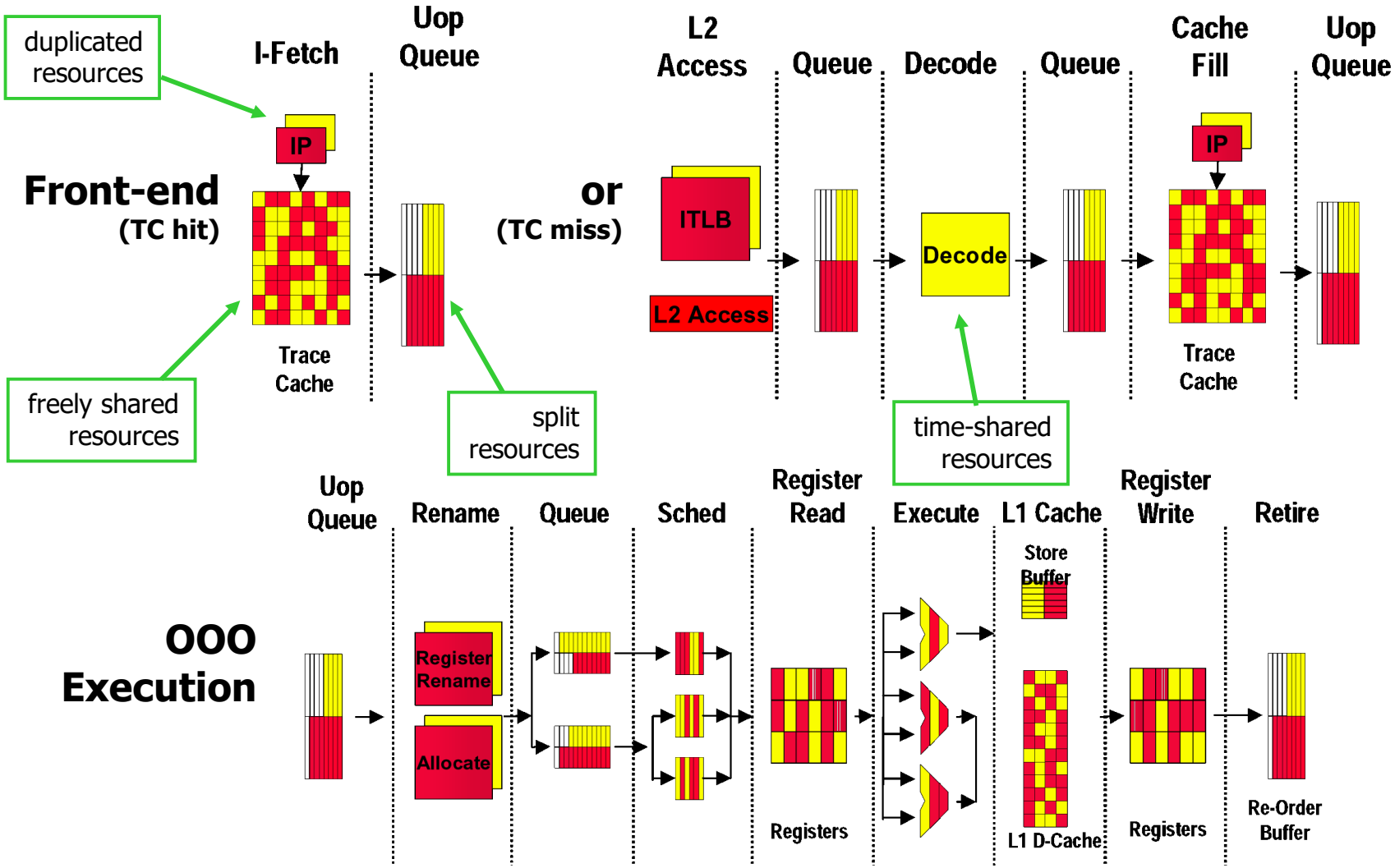
AMD Hammer x86-64: Extension of x86 to 64 bits



Adopted in 2004 by Intel

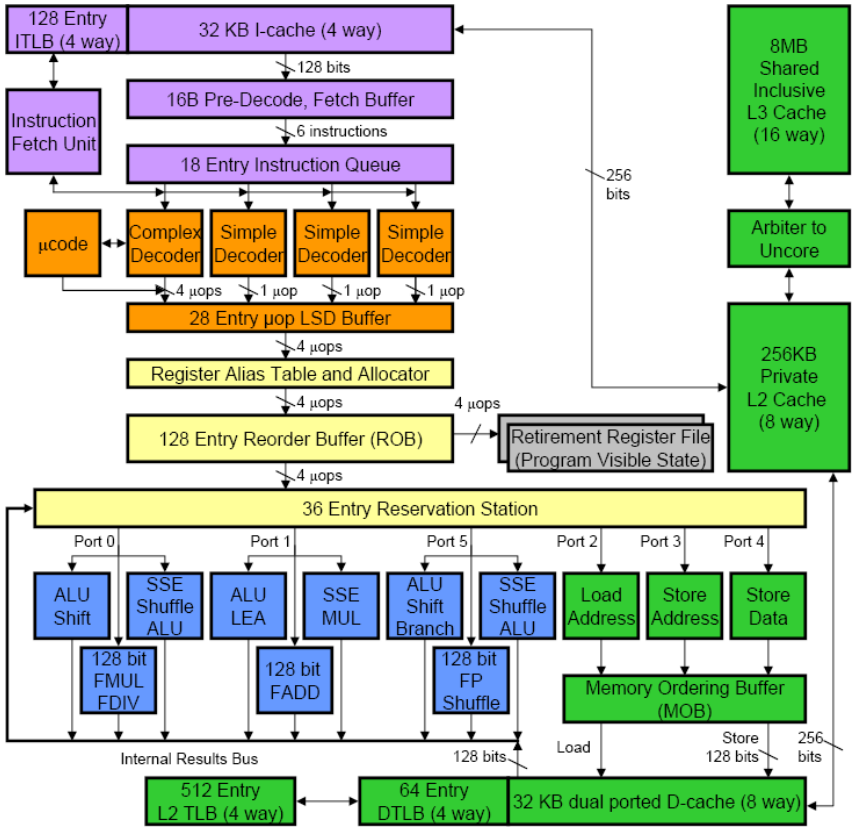
Source: Microprocessor Report, © Cahners 2001

Simultaneous Multi-Threading: Xeon Pipeline

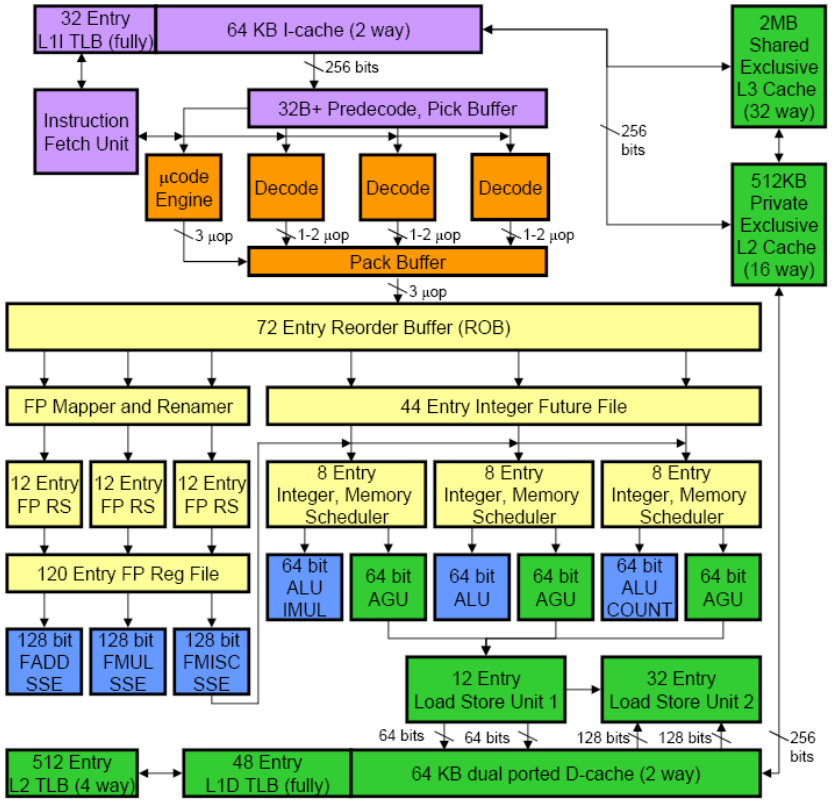


Source: Marr et al., © Intel 2002

Intel Nehalem and AMD Barcelona: Now Oldish Microarchitectures



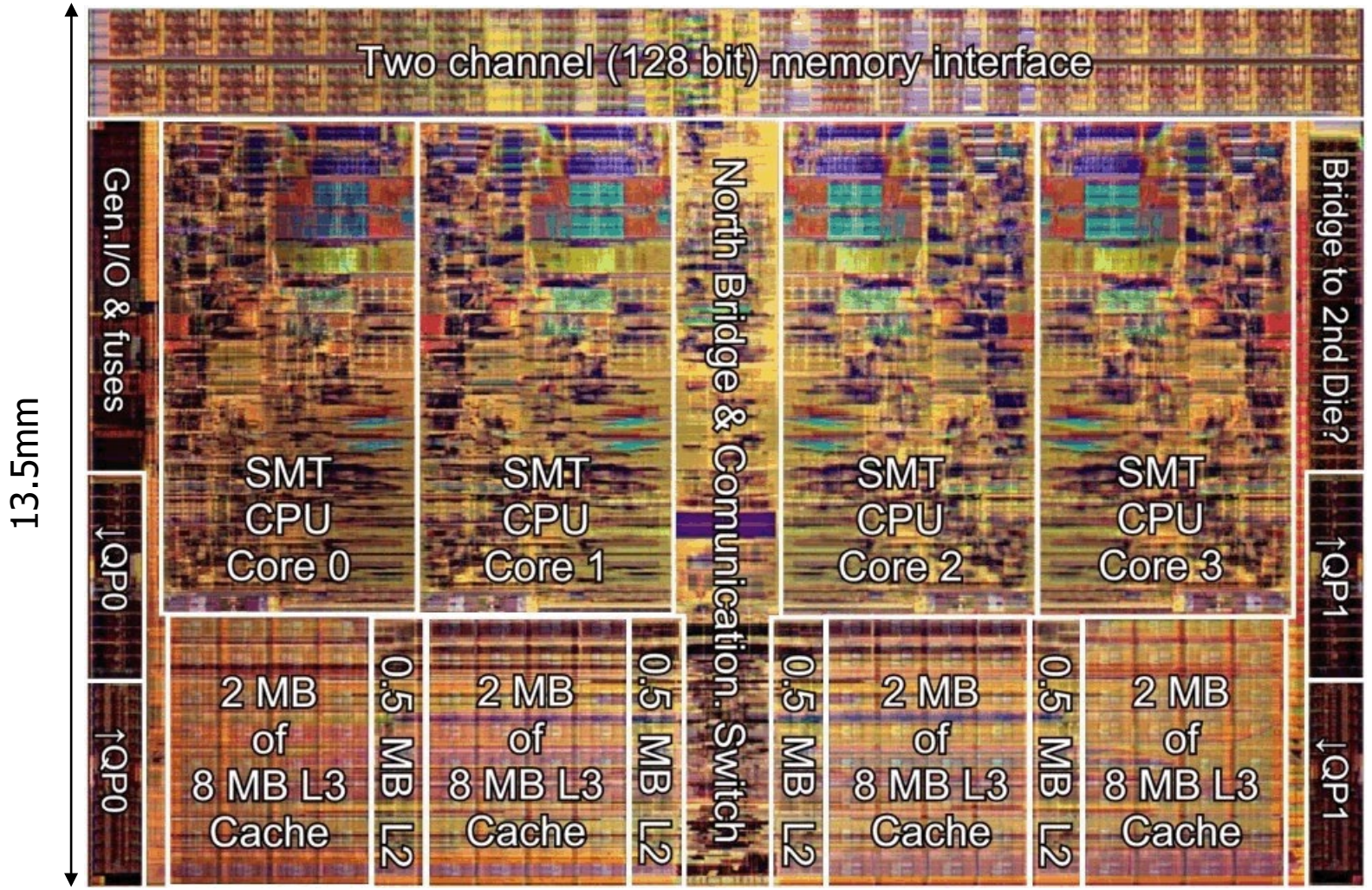
Intel Nehalem



AMD Barcelona

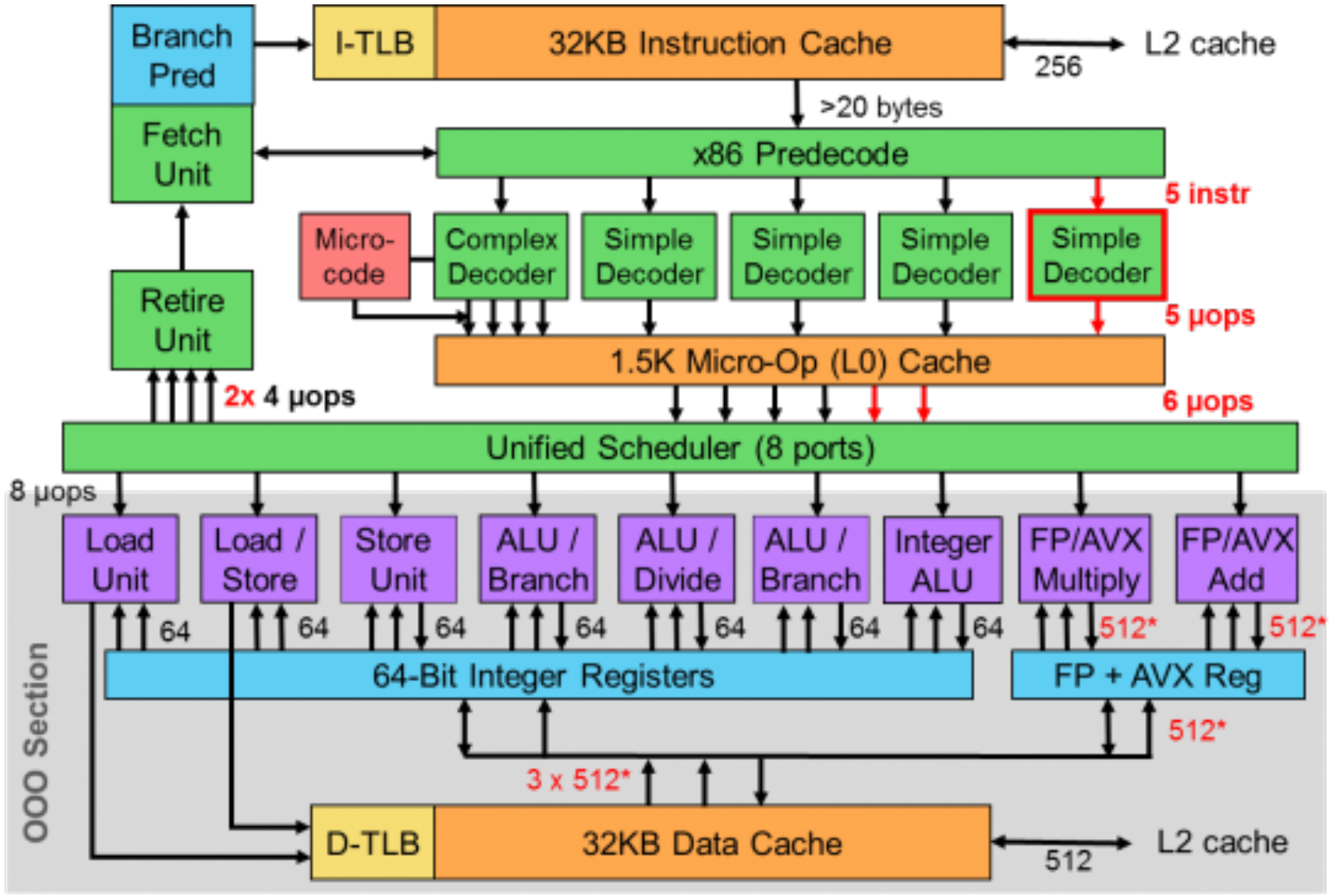
Source: http://realworldtech.com/, © RWT 2008

Intel Nehalem Die
265mm² 731M transistors

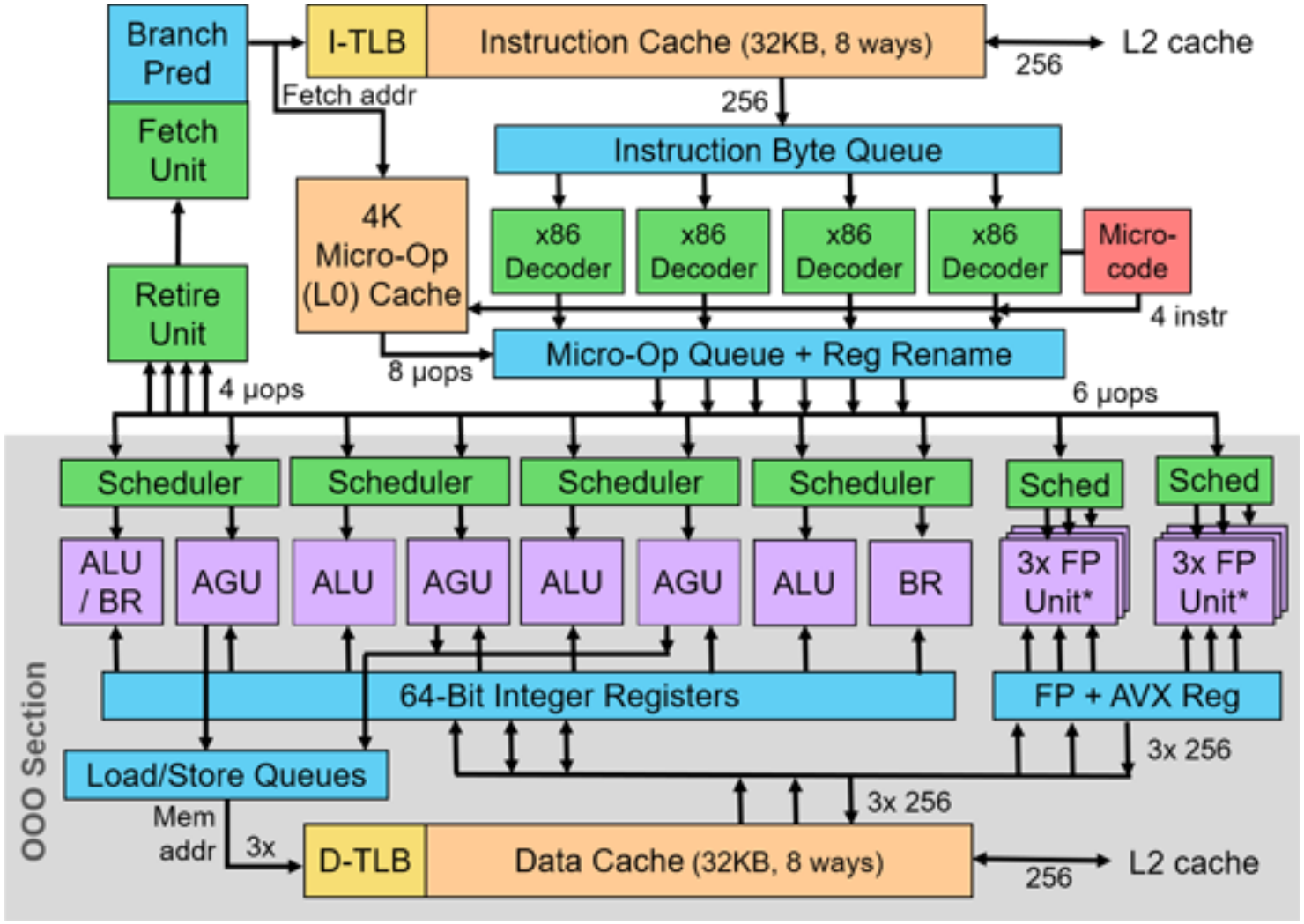


Source: http://www.chip-architect.com/

Intel Skylake Microarchitecture

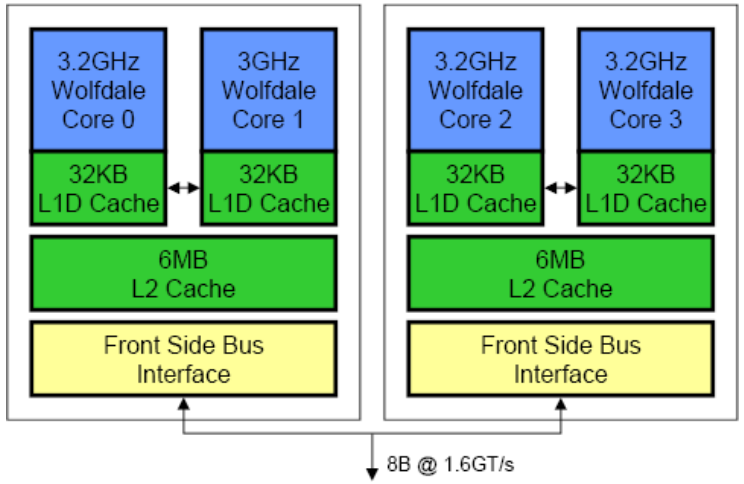


AMD Zen 3 Microarchitecture

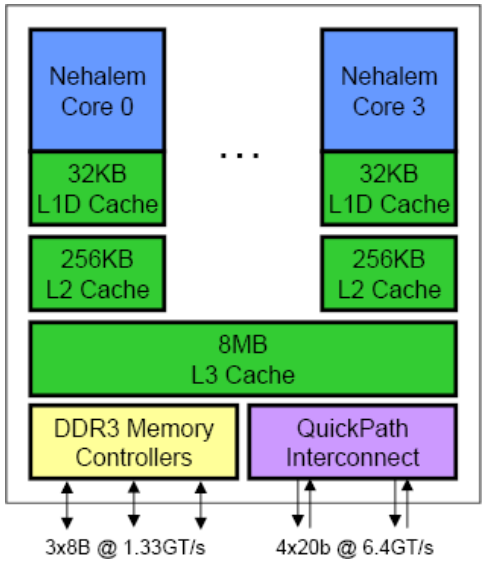


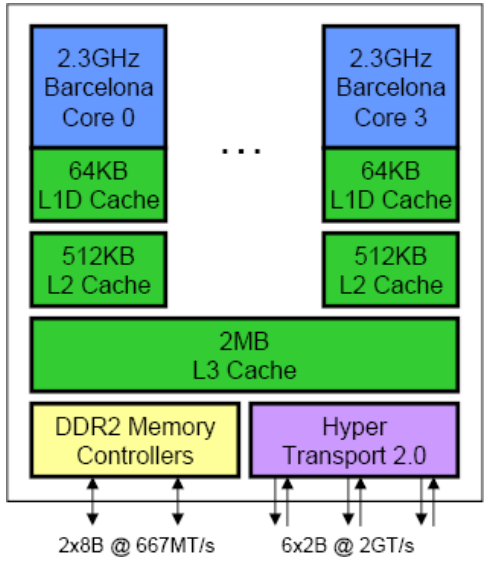
Where the Challenge is Today: Multicore and System Architecture

Intel Core



Intel Nehalem





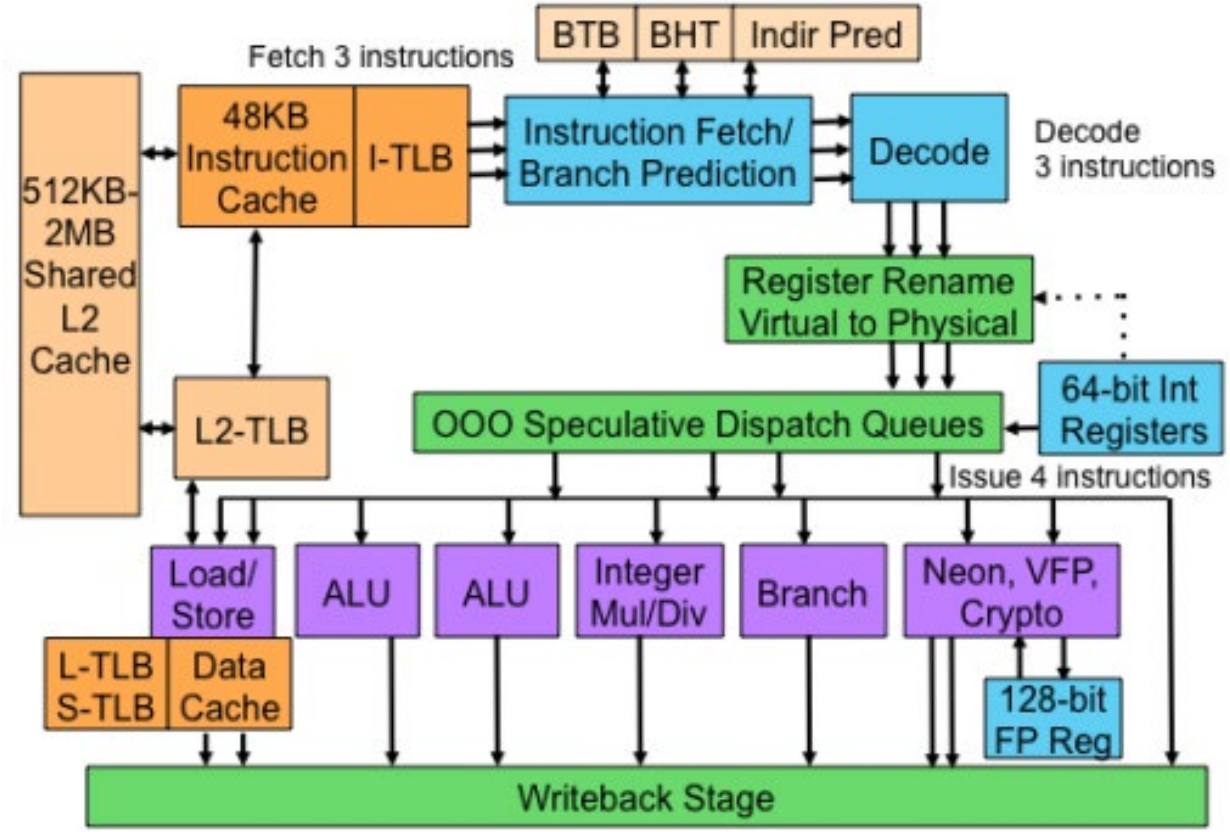
AMD Barcelona

Source: http://realworldtech.com/, © RWT 2008

Conclusions

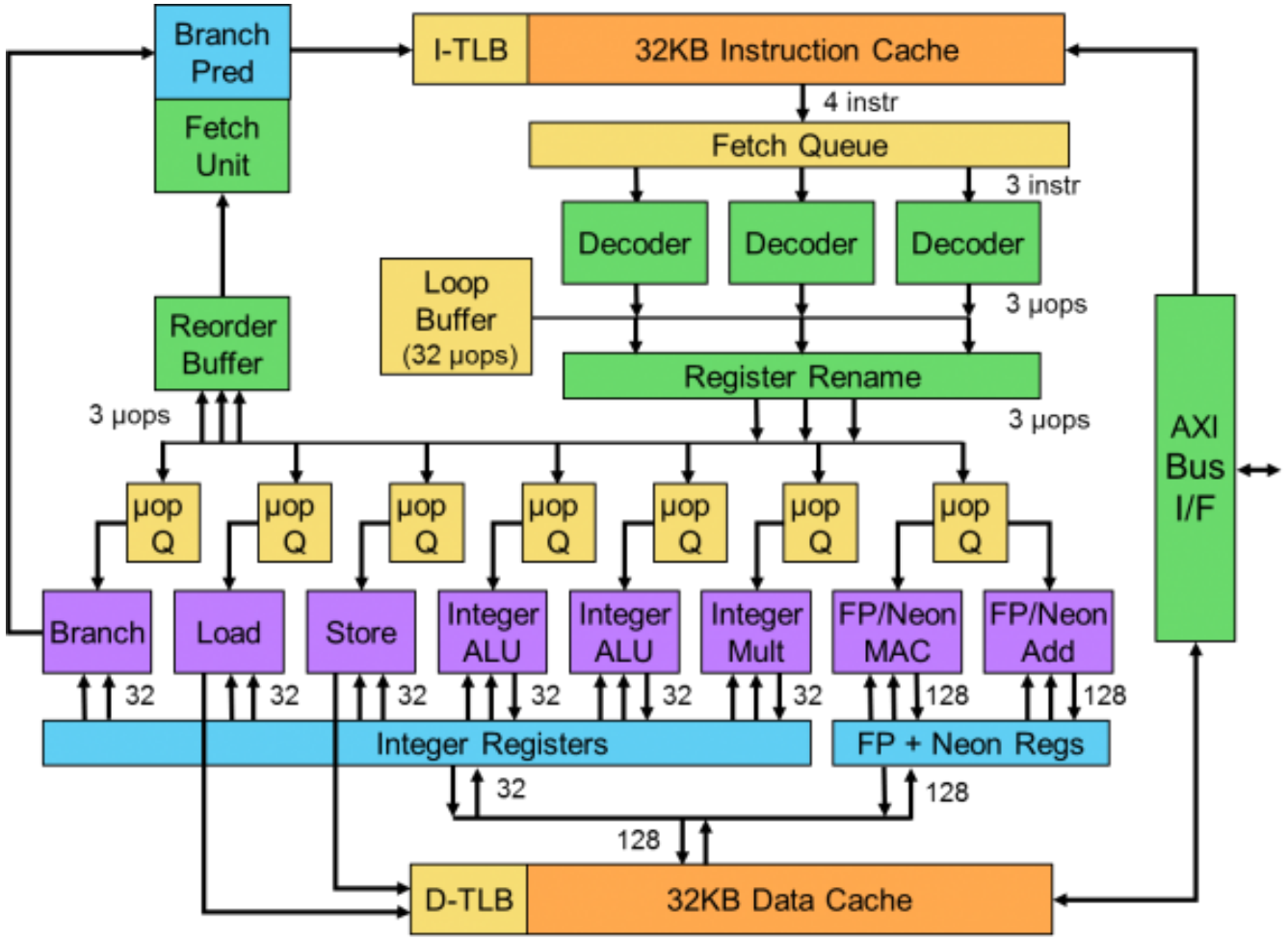
- x86 is the oldest important ISA around
- It is not absolutely fixed but a family of constantly evolving ISAs with many new add-ons (MMX, SSE, AVX, CMOV, etc.)
- Intel has managed to continue pushing the performance by adapting to its CISC nature the techniques developed to speed-up newer RISC processors
  - Similar work has been done by some competitors—notably by AMD with Athlon, for a few months the fastest x86 processor on the market
- It has been given for agonizing many times over, it is still well alive and kicking!...

And ARM? (1/2)
Cortex-A57 Microarchitecture



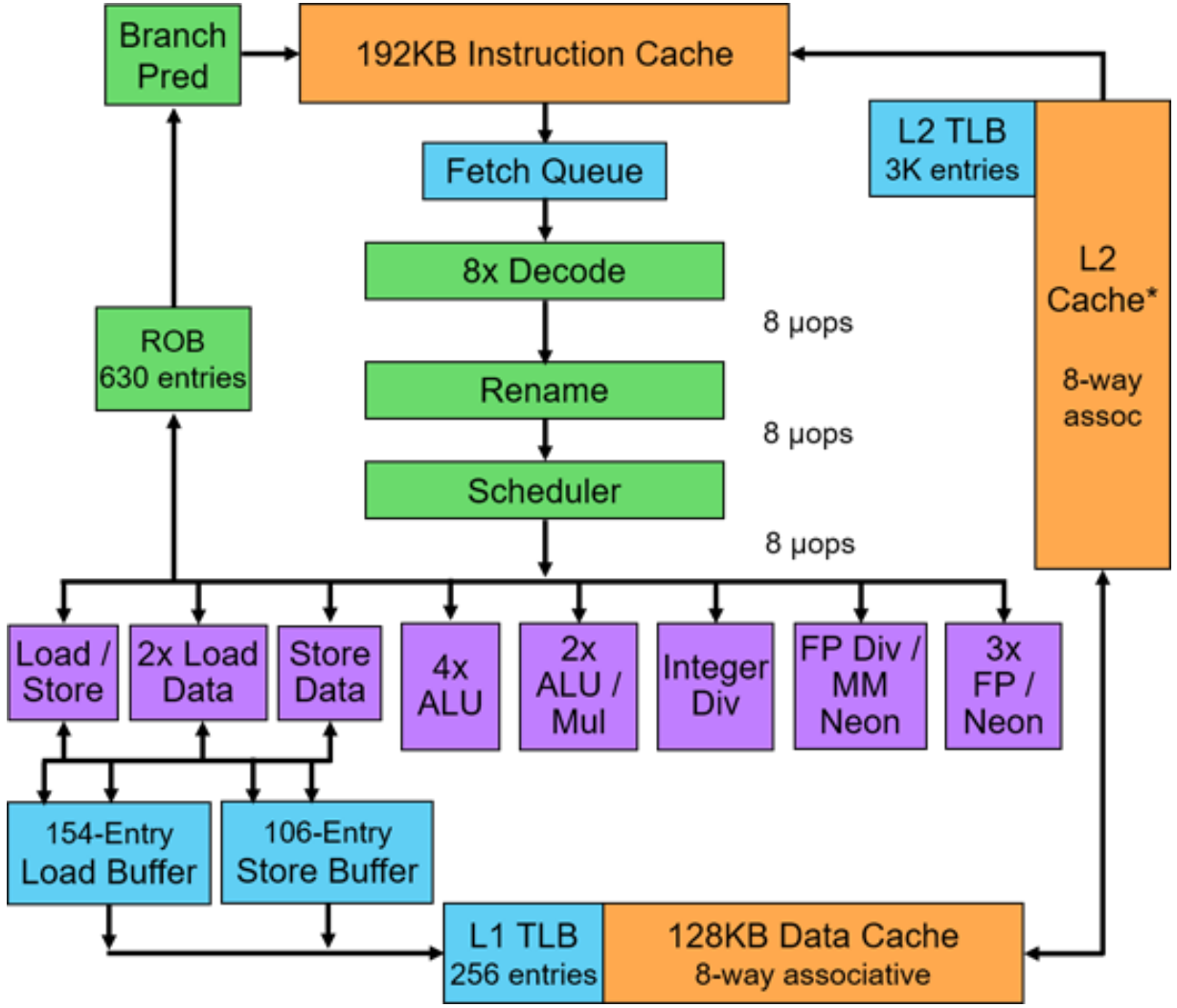
Source: Microprocessor Report, © The Linley Group 2012

And ARM? (2/2)
Cortex-A15 Microarchitecture

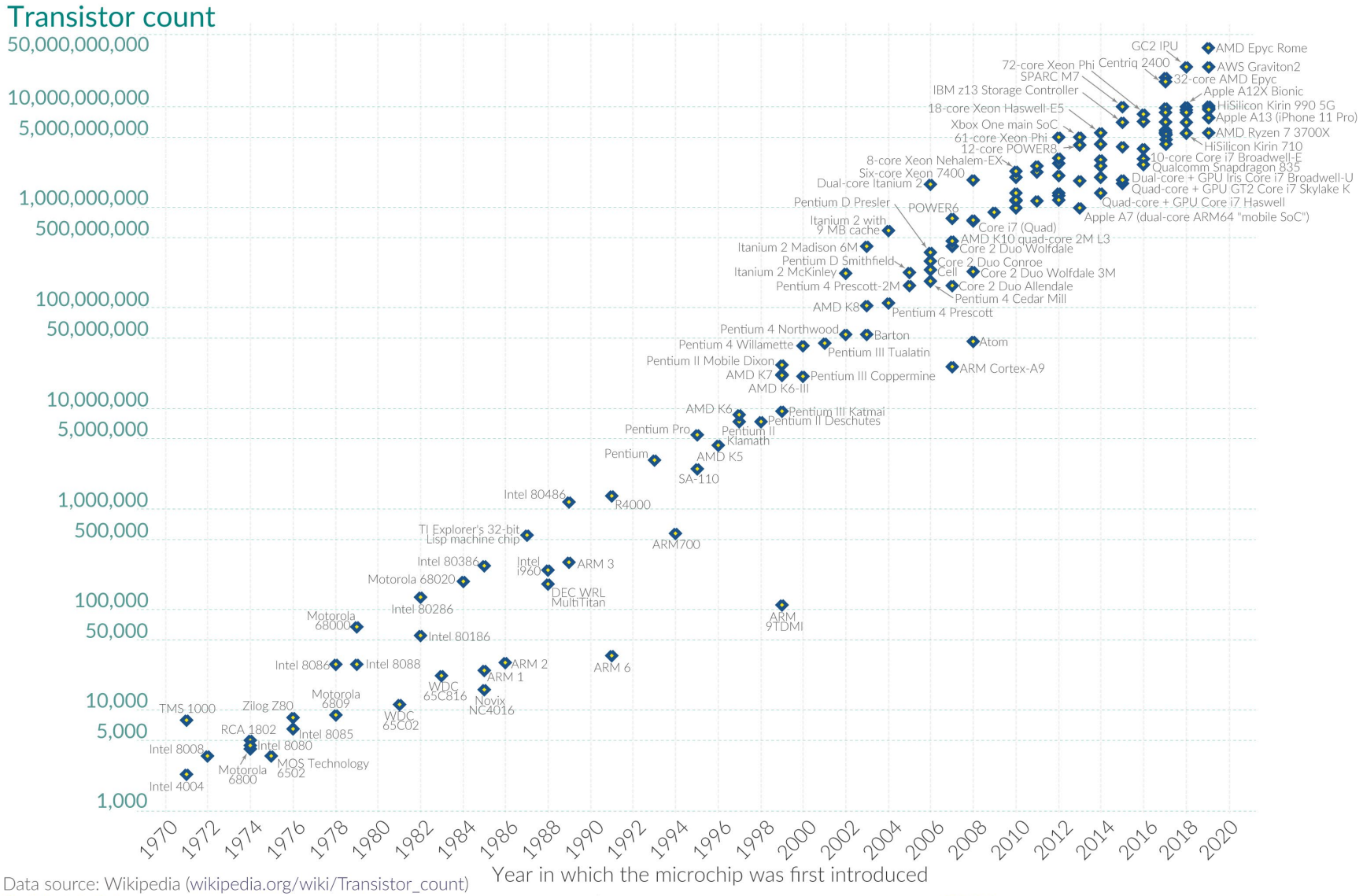


Source: Microprocessor Report, © The Linley Group 2013

Apple Firestorm ARM Microarchitecture



Transistor Count over Time



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

References

- Patterson & Hennessy, COD – RISC-V Edition
  - **Sections 2.17** to **2.19** (ARMv7, ARMv8, x86), **2.22**, and **3.7** (SSE, AVX)
  - **Section 4.12** (Core i7 and ARM Cortex-A53)
  - **Section 5.13** (ARM and Intel memory hierarchies)