1990 Worldwide Developers Conference

Michael Slater

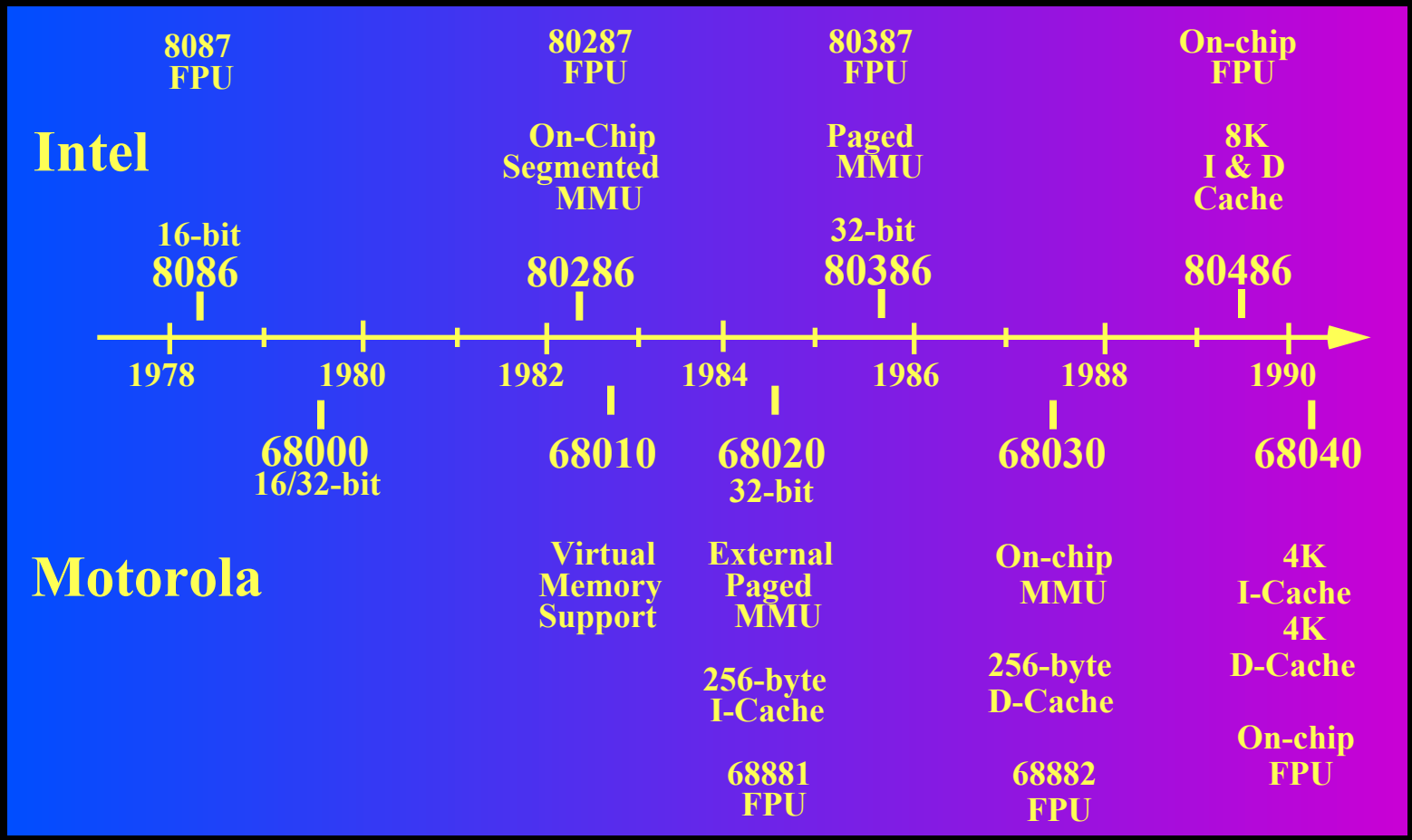
Editor and Publisher
Microprocessor Report

The Past, Present, and Future of the 68000 Family

Outline

- Origins of the 68000
- Evolution of the 68000
- Future of the 68000

Intel and Motorola Timeline



A Strong Beginning: the 68000

- A 16-bit implementation of a 32-bit architecture
- No attempt at compatibility with 6800 software
- “Regular” (orthogonal) architecture
- One oddity: separate address and data registers

The Heart of the Macintosh

- Used in Macintosh 128K, 512K, Plus, SE
- Used in Apple LaserWriter, LaserWriter IINT, SC
- Also used in early UNIX workstations, many embedded applications

Sixteen or Thirty-Two Bit?

- 16-bit data bus, 24-bit address bus
- 16-bit ALU, with microcode to perform 32-bit operations
- 32-bit address and data registers
- Let’s call it 16/32

A Small Step: the 68010

- Needed to support virtual memory; not used by Apple
- External MMU asserts BERR on page fault
- Internal state stored on stack
- Processor continues after page fault handler by restoring state from stack

An Even Smaller Step: the 68012

- Never heard of it? Don’t worry; not used by Apple
- Provides 31 address lines
- Rumored to have been developed for HP
- No longer in data book; effectively obsoleted by 68020

The Leap to 32 Bits: the 68020

- Full 32-bit ALU, so 32-bit operations are as fast as 16-bit operations
- Barrel shifter speeds multibit shifts
- 32-bit address and data buses
- Coprocessor interface
- On-chip 256-byte instruction cache
- Used in Macintosh II, LaserWriter IINTX

68020 Performance

- Basic bus cycle reduced from 4 clocks to 3 clocks
- Overlapped instruction fetch and execute
- Many instructions execute in fewer clock cycles
- Cache reduces effect of memory wait states

68020 Instruction Set Extensions

- New addressing modes
  - Scaled index
  - 16- and 32-bit displacements for address register indirect with index
  - Memory indirect, pre- or post-indexed
  - Base register

Instruction Set Extensions *(cont.)*

- Bit field instructions
  - Test bit field
  - Set bit field to 1’s or 0’s
  - Complement bit field
  - Move bit field to or from data register
  - Scan for first bit set

Instruction Set Extensions *(cont.)*

- PACK and UNPK for BCD data
- Compare and swap
  - CAS and CAS2 for multiprocessor interlocks
- Breakpoint instructions
- Return and deallocate parameters
- Move to control register, or to specified address space

Instruction Set Extensions *(cont.)*

- CALLM and RETM: orphan instructions
- Coprocessor instructions
- CMP2 and CHK2:
  bounds-checking instructions
- Conditional trap with parameter

Enhanced Instructions

- 32 x 32 multiply
- 64/32 and 32/32 divide
- 16- and 32-bit constants for many instructions

Compatibility

- 68020 instruction set is proper superset of 68000 instruction set, so all 68000 code should run on 68020
- 68020 code, however, will not run on 68000 if any instruction set extensions are used
- So, to optimize for Macintosh II, must have different version for 68000-based Macintoshes

Misaligned Data Support

- 68000 will trap if 16-bit value accessed at odd address
- 68020 will allow such an access, performing two bus cycles if needed
- Thus, some programs may run on Macintosh II but not on 68000-based Macintoshes, even if they don’t use any new instructions

Memory Management for the 68000 Family

- For the 68000/010: no standard
- For the 68020: 68851 paged memory management unit (PMMU)
- PMMU is optional in Macintosh II; standard system has simpler address mapper chip in PMMU socket

An Incremental Step: the 68030

- Combines 68020 and subset of PMMU
- Adds 256-byte data cache
- Adds new 2-clock special bus cycle
- Adds burst transfers for cache line fills
- Used in Mac IIx, IIci, IIcx, IIfx, SE/30

Importance of the 68030

- Unlike 68020, no major architectural change
- Adding PMMU on-chip makes it a standard part of the system
- Having PMMU on-chip also allows address translation delay to be hidden
- Available at higher clock rates (Macintosh IIfx)

The Big Leap: the 68040

- Combines 68030 and FPU
- Increases cache size to 4K bytes each
- Pipelined processor for dramatic reduction in clocks per instruction
- About 3 x 68030 speed at same clock rate for integer code

Instruction Set Extensions

- Only one new user instruction: MOVE16, which moves 16 bytes using burst transfer
- Additional supervisor-mode instructions for cache and MMU control
- Creeping instruction set complexity stops!

68040 Floating Point Unit

- Only basic operations supported directly in hardware
- Transcendental functions supported with “softwar envelope”
  - Transparent to application program, as long as operating system provides emulation via traps

68040 Caches

- Provide write-back and copy modes
- Sophisticated design minimizes processor stalls due to cache a overhead
- Bus supports burst writes as well as reads
- Bus snooping for cache coherency

Catching up to RISC?

- 68040 does achieve performance levels comparable to today’s mid-range RISC systems, but...
- RISC systems have been shipping at that performance level for over 1 year, while 68040 is not yet in production and systems won’t ship until later this year

What’s Next?

- The 68050, of course
- Bigger caches
- Even faster floating point
- More parallelism

The Big Test: Superscalar

- Decode, dispatch, and execute more than one instruction per clock cycle from a sequential instruction stream
- Without superscalar implementation, peak is 1 instruction per clock
- With superscalar implementation, peak is probably around 3 instructions per clock with existing software

So Will the 68050 be Superscalar?

- You'll have to ask Motorola
- RISC processors have a much easier time
  - Fixed-length instructions
  - Simple instruction formats

The RISC-CISC Gap

- RISC will stay in the lead
- Gap may widen as superscalar designs become more important and more effort goes into RISC designs
- For mainstream business computer users, continuit of software is more important that maximum performance

The power to be your best