Intel 80x86 Timeline (selected)

- 1978: 8086, 16-bit processor
- 1982: 80186/80286, 16-bit (MMU, fast * /)
- 1985: 80386, AMD Am386, 32-bit ISA (IA-32)
- 1989: 80486
  - RISC-like pipelining, integrated x87 FPU
- 1993: Pentium
- 1995: Pentium Pro
  - u-op translation, OoO reg rename, speculative execution, in package L2
- 2003: Core, AMD Athlon64, AMD Opteron (64-bit ISA)
  - AMD64, which Intel adopts to x86-64
- 2004: Pentium 4 Prescott
  - deep pipeline, >4GHz,
- 2006: Core 2 (lower power, multicore)
- 2008+: Core i3/i5/i7 (Sandy Bridge, Ivy Bridge, Haswell, Broadwell, …)
Important Trends

- **Moore’s Law**
  - Number of transistors will double about every ~18 months
  - Scientific or Economic?
  - This is about to end!

- **Denard Scaling**
  - Power density of transistors remains constant across process generations
  - Both Voltage and Current scale down
  - This has ended!
Implications of “End of Scaling”

- **Moore’s Law:**
  - If scaling stops, no more transistors for “free”
  - Need to increase area or be more creative in architecture
    - Exciting! (from research view)

- **Denard Scaling**
  - Designs limited by power and heat
    - Transistors compute by generating heat! (one view)
  - Can’t dissipate heat, so we can’t clock faster, or potentially activate entire chip at once! (Dark Silicon)
Modern x86-64 Implementations

- x86-64 (x86) is an Instruction Set Architecture (ISA)
- ISA is very different (!) than Microarchitecture/Organization
- x86 often thought of as complex, but “simple” implementations exist
  - Comparable to certain ARM cores
- uOps (micro-operations)
  - RISC-like instructions that are executed by the processor
uOps

- x86 instructions are variable length (1-17 bytes)
  - Decoding is more complex than fixed-length
  - Some instructions not implemented directly in hardware!
- Internally, x86 implementations are RISC machines
- Example:
  - ADD (%rdx), %rax
  - LOAD (%rdx), %preg; ADD %preg, %rax
- Simple/Common x86 operations will be a single uOp
  - Hardware is optimized for these
General OoO Execution

Fetch → Decode → Rename → Dispatch → Buffer of instructions → Issue → Reg-read → Execute → Writeback → Commit
Intel Haswell

- 22nm process
- 4th Generation Core (iX) Architecture
- x86-64 + AVX, SSE, FMA, TSX
  - SIMD extensions
  - Hardware Transactional Memory (oops!)
- 2-8 cores, 8+ in server chips
- 2-20 MB L3 cache
- OoO, dynamically scheduled, speculative execution
Haswell

56 μop Decode Queue

4 μops

192 Entry Reorder Buffer (ROB)

168 Integer Registers

168 AVX Registers

48 Entry Branch Order Buffer

72 Entry Load Buffer

42 Entry Store Buffer

60 Entry Unified Scheduler
Partners

- You **must** work in pairs for this assignment (one group of 3)
- If you want input on partners, fill out catalyst survey entering name of preferred partners (Teddy & I don’t count)