How do modern processors work ?

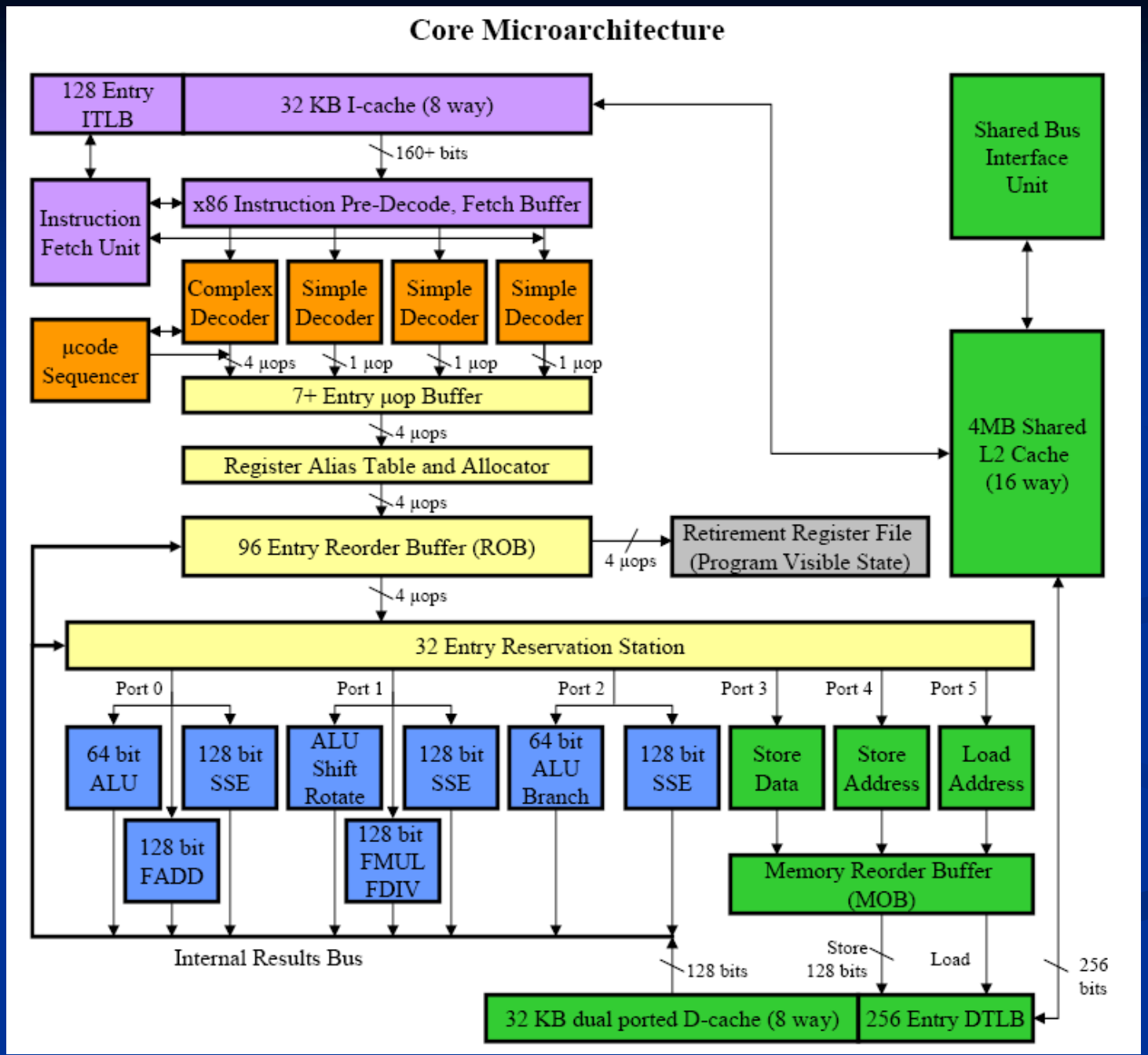
Under the hood of the

Intel Core 2 Duo

vasko.anton@gmail.com

Description

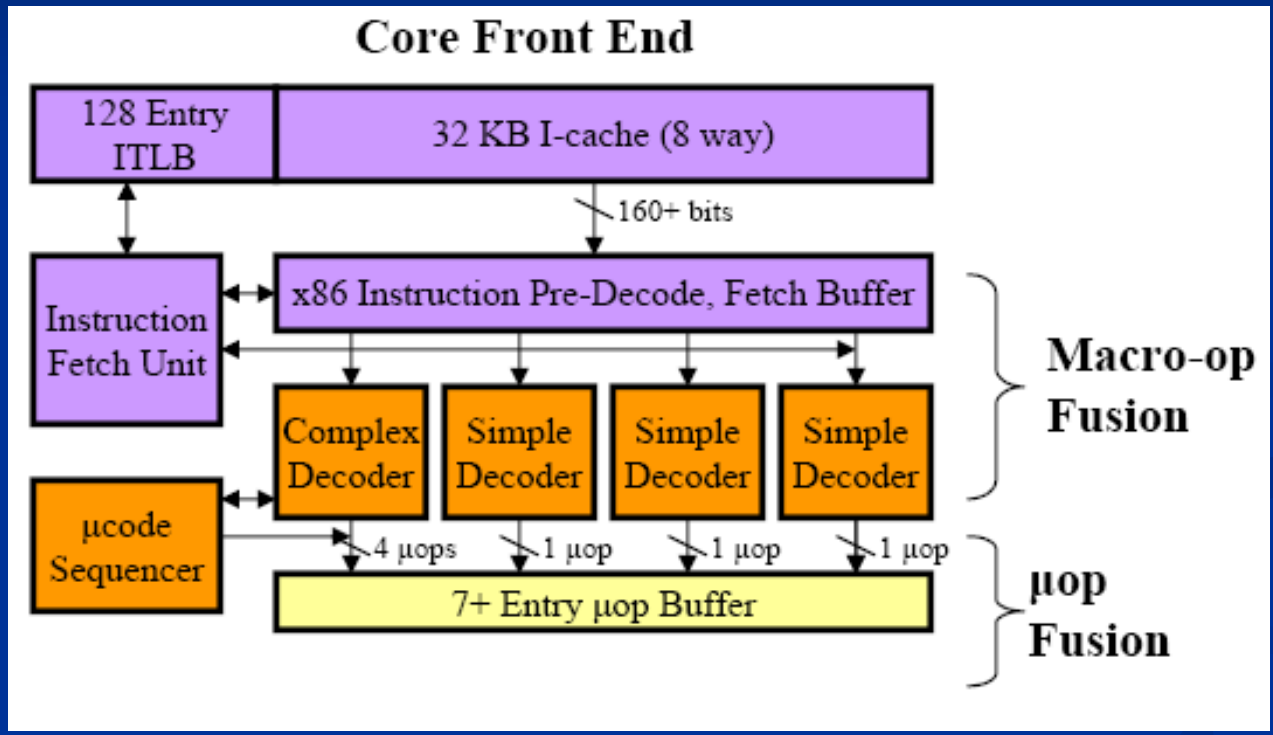
- dual core
- 64-bit (EMT64 = AMD64)
- 4-issue
- out-of-order
- 32 KB L1 instruction cache
- 2 x 32 KB dual ported L1 data cache
- shared 4MB L2 cache
- clock about 3GHz


Core Microarchitecture
128 Entry ITLB
32 KB I-cache (8 way)
160+ bits
Instruction Fetch Unit
x86 Instruction Pre-Decode, Fetch Buffer
Complex Decoder
Simple Decoder
Simple Decoder
Simple Decoder
µcode Sequencer
4 µops
1 µop
1 µop
1 µop
7+ Entry µop Buffer
4 µops
Register Alias Table and Allocator
4 µops
96 Entry Reorder Buffer (ROB)
4 µops
Retirement Register File (Program Visible State)
4 µops
32 Entry Reservation Station
Port 0
Port 1
Port 2
Port 3
Port 4
Port 5
64 bit ALU
128 bit SSE
ALU Shift Rotate
128 bit SSE
64 bit ALU Branch
128 bit SSE
Store Data
Store Address
Load Address
128 bit FADD
128 bit FMUL FDIV
Memory Reorder Buffer (MOB)
Internal Results Bus
128 bits
Store 128 bits
Load
256 bits
32 KB dual ported D-cache (8 way)
256 Entry DTLB
Shared Bus Interface Unit
4MB Shared L2 Cache (16 way)

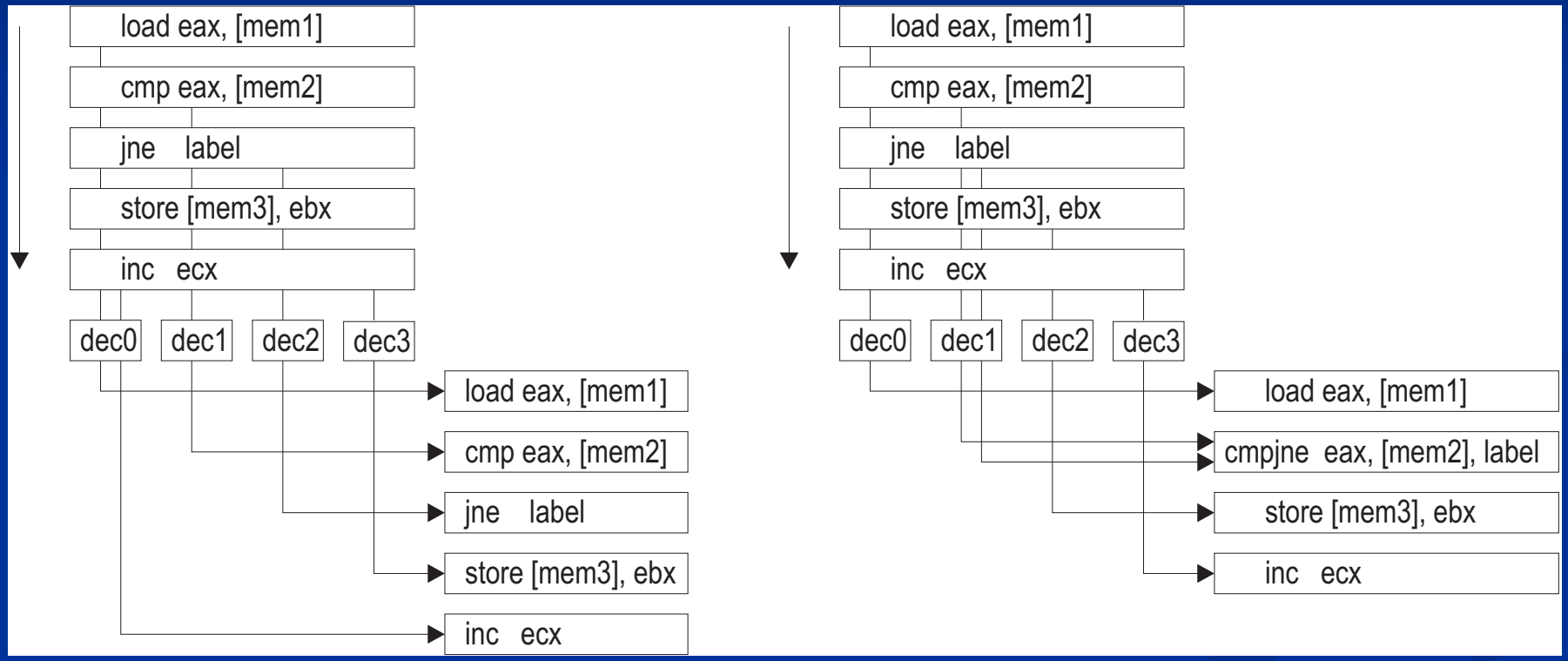
Sub-blocks

- Front end
- Out-of-order Engine
- Memory Subsystem

The Front End



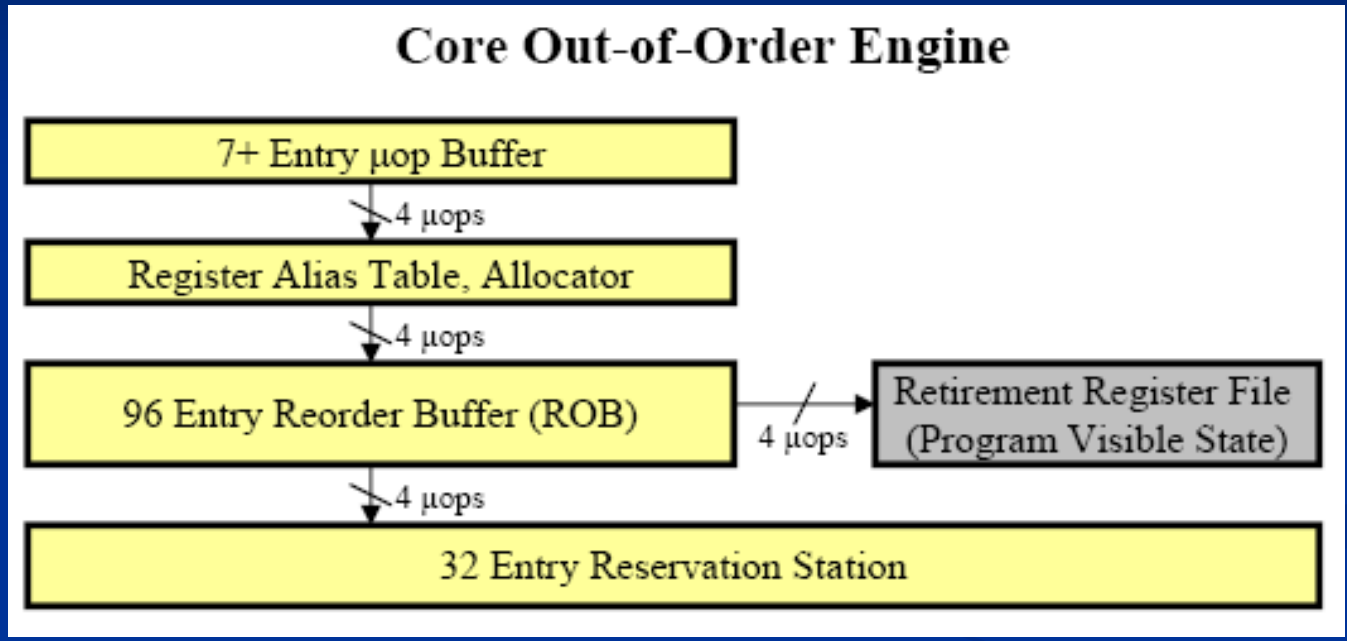
Macro-op Fusion



µ-ops Fusion

- if some instructions requires two operations => 2 µops
- sometimes they can be fused into 1 µop
- 1 record through the pipeline, split before execution units/memory
- Macro-ops fusion for different class of instruction

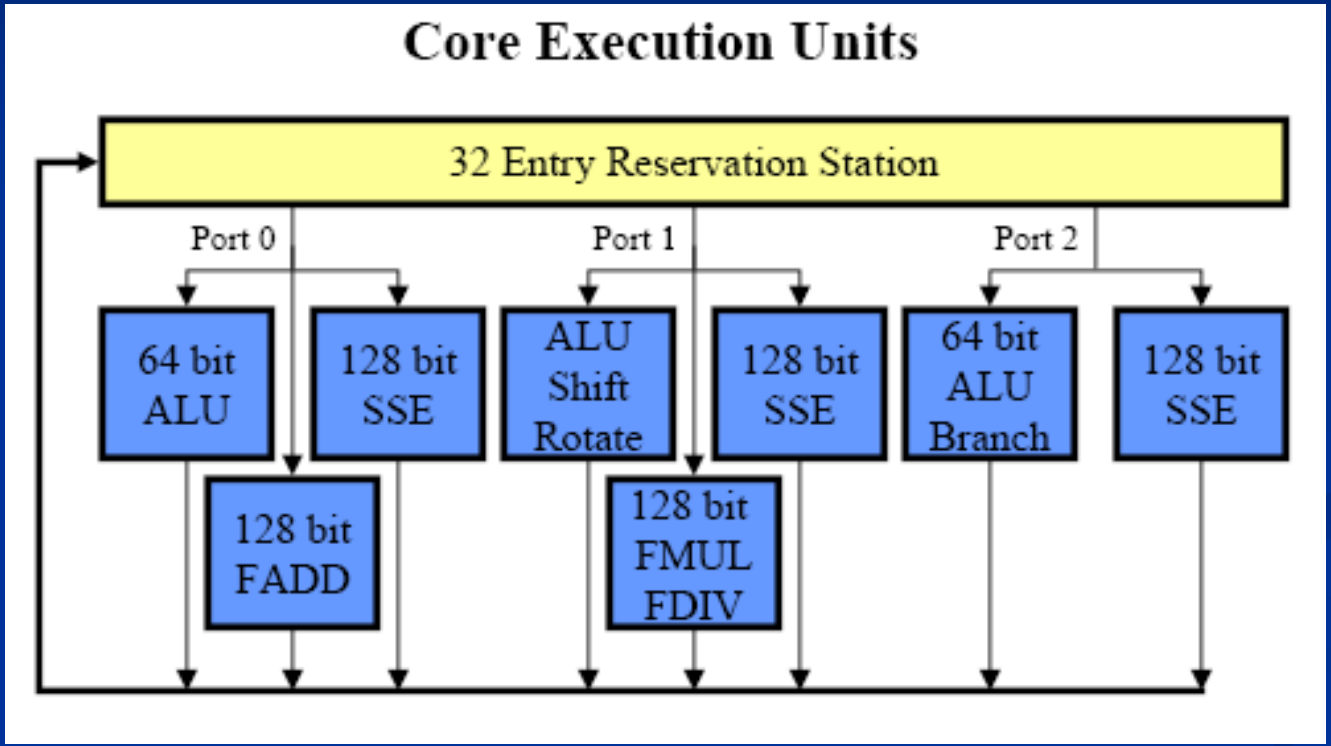
Out-of-order Engine



- False dependency:

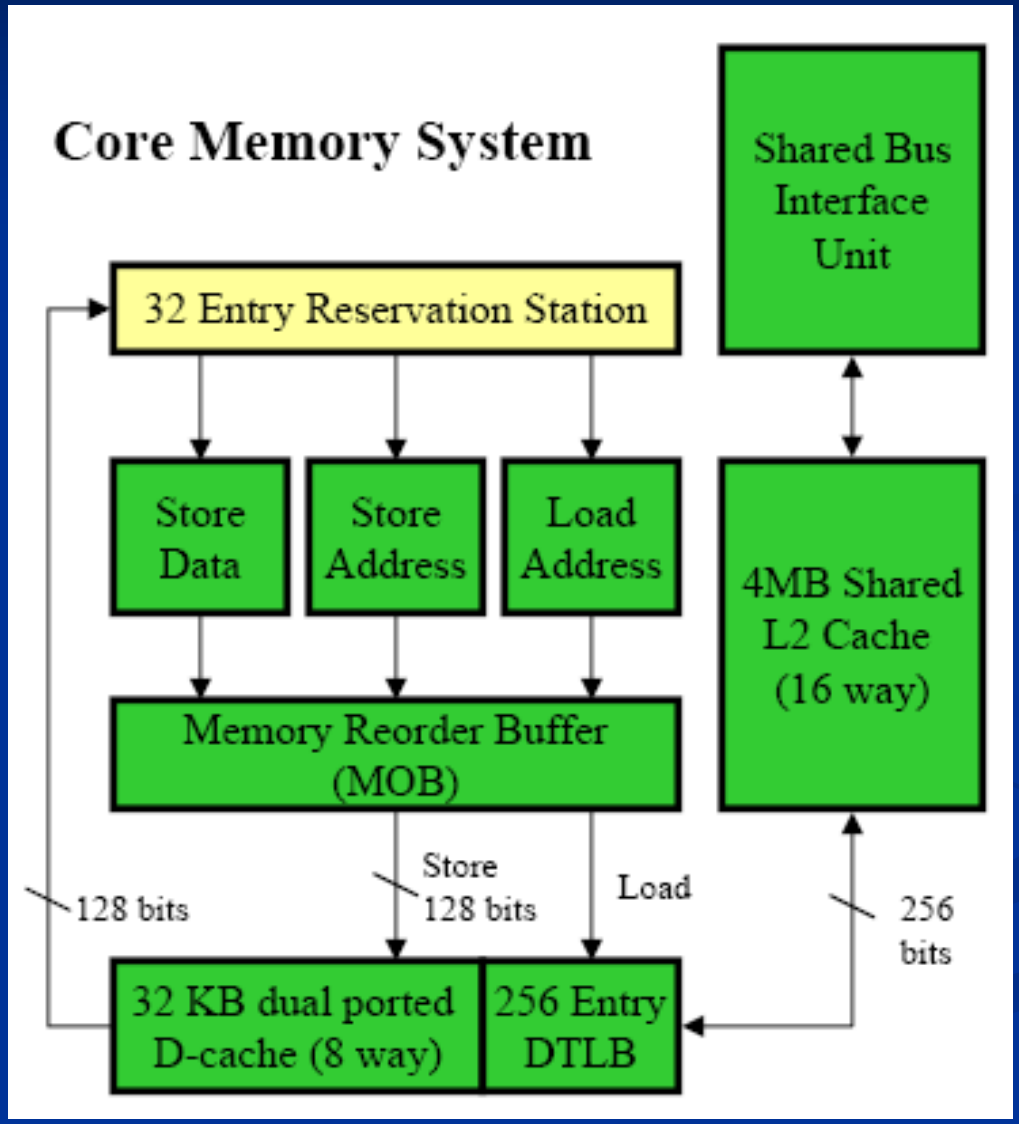
  mm0 = mm1 + mm2, mm2 = mm3 - mm1

Execution Units



- SSE instructions are executed in one cycle only!

Memory System

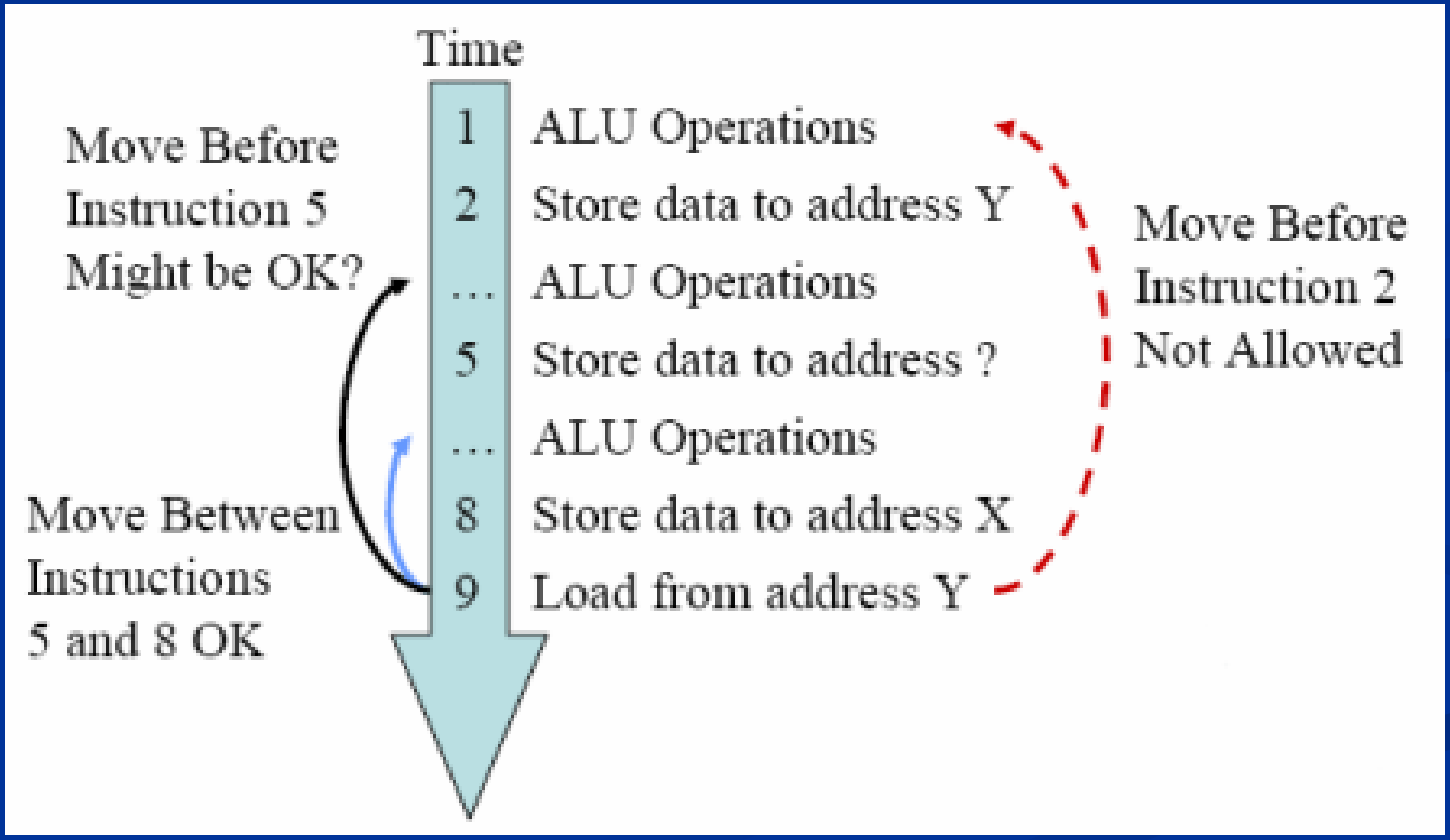


Features of Memory System

- Prefetching (software, hardware)
- Directly data transfer between cores' L1 data caches
- Shared L2 cache
- Memory Reorder Buffer (MOB)

Memory Aliasing Problem

- only out-of-order CPU



Old solution

1. All loads are delayed if a store is in-flight with an unknown address
2. Loads cannot proceed ahead of an aliased store data µop
3. A store cannot be moved in front of another store

- safe but pessimistic

New solution

- 97+% of loads and stores do not alias ([1])
- dynamic alias predictor
- loads can be speculative moved
- if bad prediction -> exception and flush of the pipeline (stall)

Conclusion

- Many features are already available in AMD processors
- SSE is becoming important (1 cycle)
- 32-bit CPUs are history – AMD64, EMT64
- Not speed, but parallelism:

1. Logical – SSE
2. Physical - dual core, quad core ...

Bibliography

1. http://www.realworldtech.com/includes/templates/articles.cfm?ArticleID=RWT030906143144

2. http://www.chip-architect.com/news/2003_09_21_Detailed_Architecture_of_AMDs_64bit_Core.html

3. http://www.svethardware.cz/art_doc-3B56A5C905E08771C125715E00793B42.html

Thank you for your attention !

vasko.anton@gmail.com