0-to-hero

07/10 Mentors <> 07/10 Sessions

Me

– – –

Kernel developer(Windows/Linux/Own)/Microarchitecture geek (Intel)/Security Researcher (Lenovo, Microsoft, Mysql, Python etc.)/
Software Engineer V Architect/Hardware Engineer (FPGA, ASIC, Firmware researcher - BIOS, UEFI, coreboot)/Contributor/Virtualization lover (VTx)/Intel lover (SGX, VTx, TXT, MMX, VNNI)/Performance tuning ninja/System Architect.

http://linkedin.com/in/emin-ghuliev-461a22129/

Job / Education

– – –

Self-taught ninja

Work experience:

- Mindcon
- CERT Government
- APA Holding
- Lambercy
- E-GOV
- EnsignInfoSecurity (as soon as)

Software Architect/System Engineering

– – –

Algorithms (Bloom filter (space efficient), Tomasulo, Branch prediction, Speculative execution, Instruction pipelining)

Optimization (Cache types - WB, WT/Replacement algorithms - LRU,Random replacement, LIFO, FIFO - Compiler based optimization - Loop unrolling, Dead code elimination, Code coverage, Register allocation, Machine code optimization, Prefetching)

SWOT (strength, weakness, opportunities, threats) analysis.

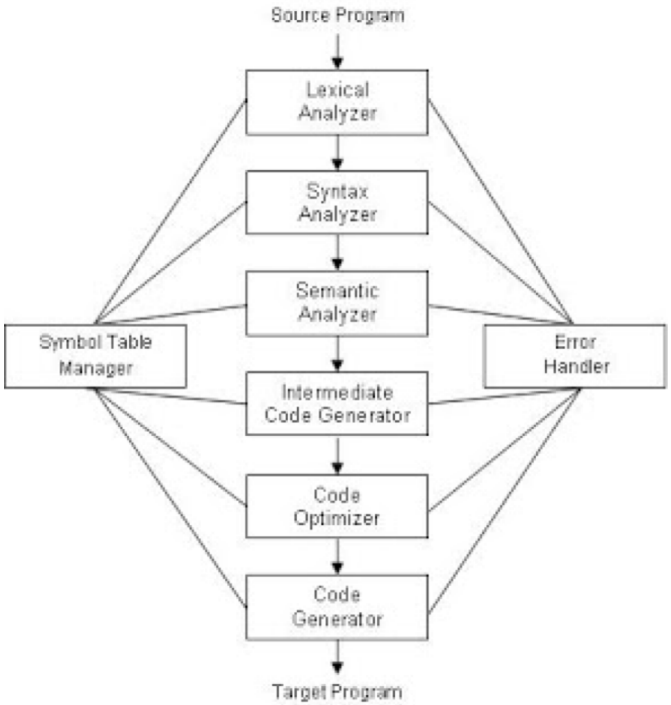
Distributed and parallel computing - protocols (IPC, RPC, Shared memory), algorithms (consistent hashing, fork-join model, pipeline, concurrency, parallelism)

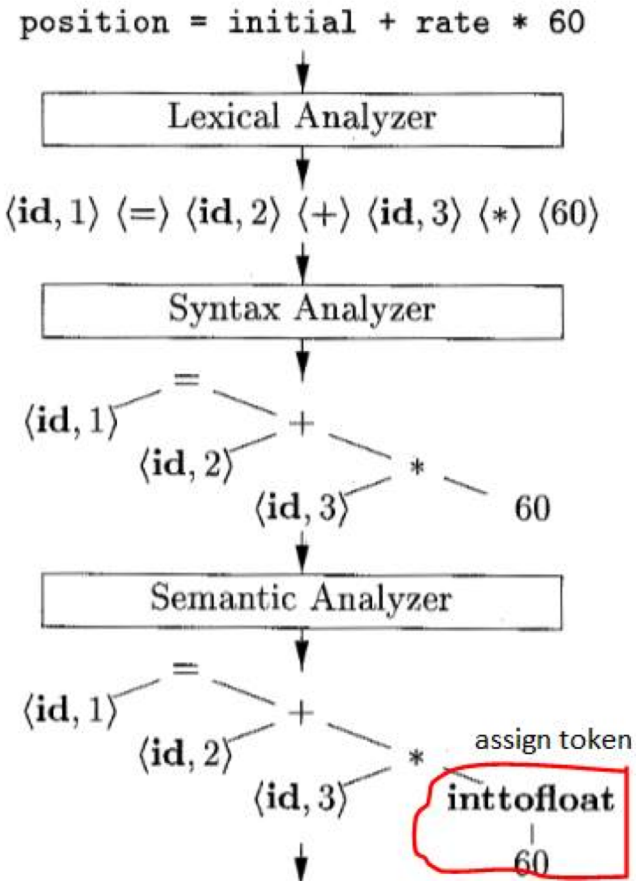
Parallelism, concurrency (reactor pattern) - Instruction-level, task-level, data level parallelism (Vector processor - SIMD, SIMT)

Virtualization (VTx/binary translation or emulation) - Memory, Device, CPU

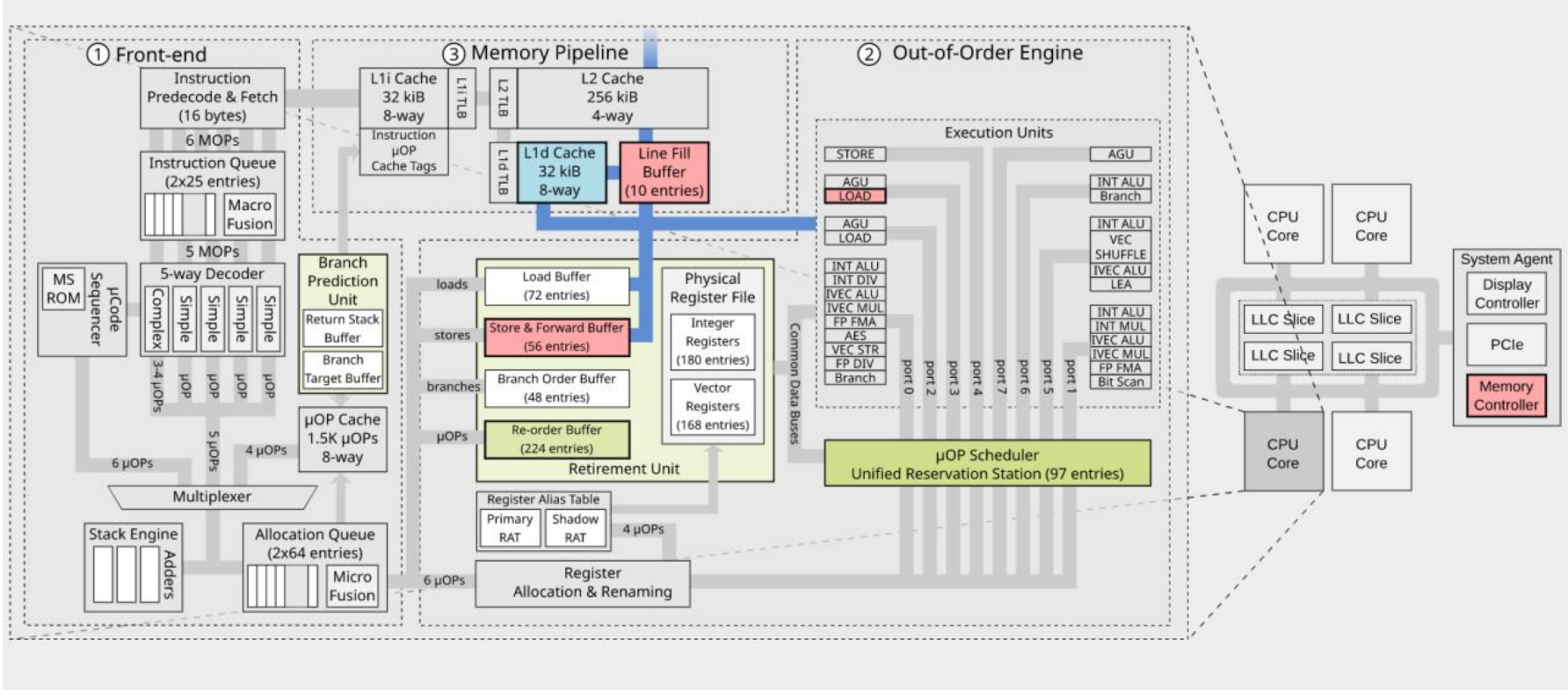
Compiler/Interpreter principles - (Lexical, token, semantic), Interpreter based VM and bytecodes

Compiler principles

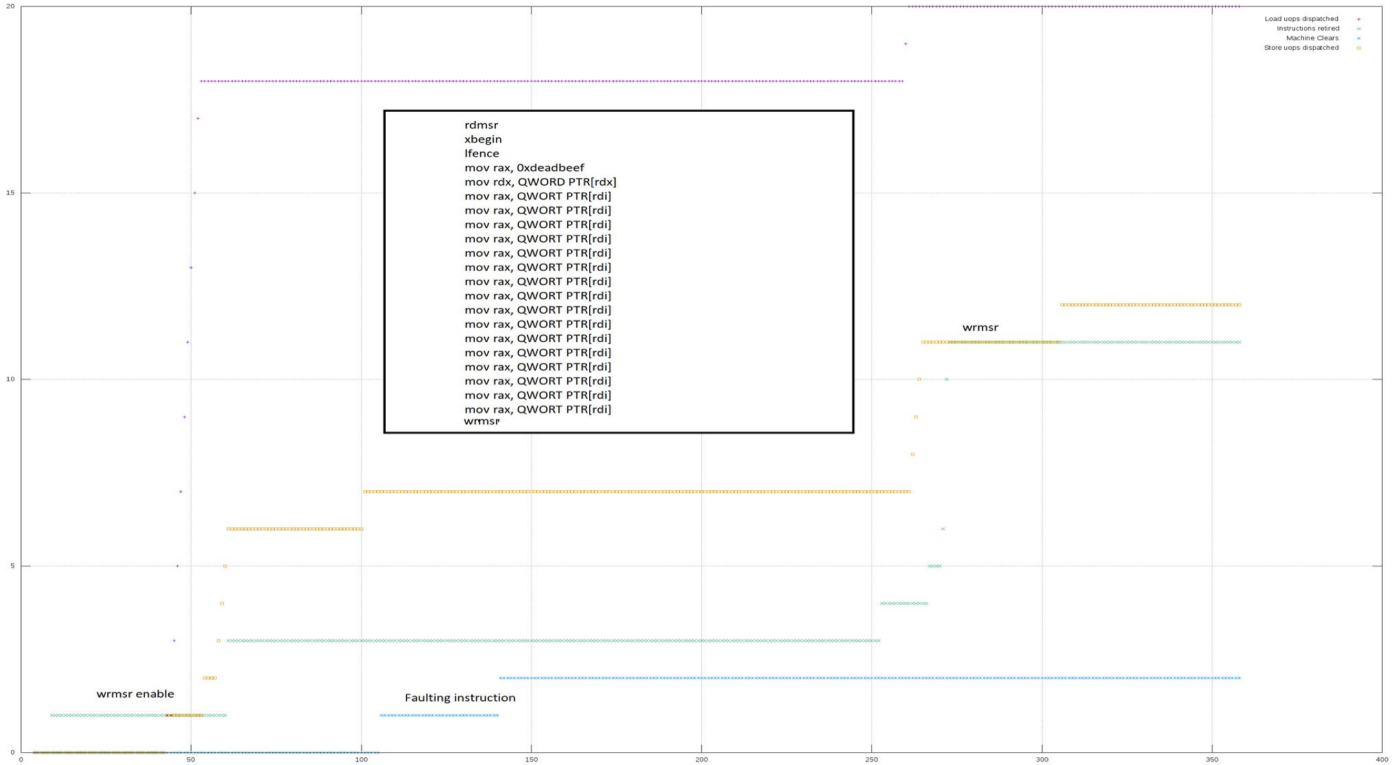




Modern CPU

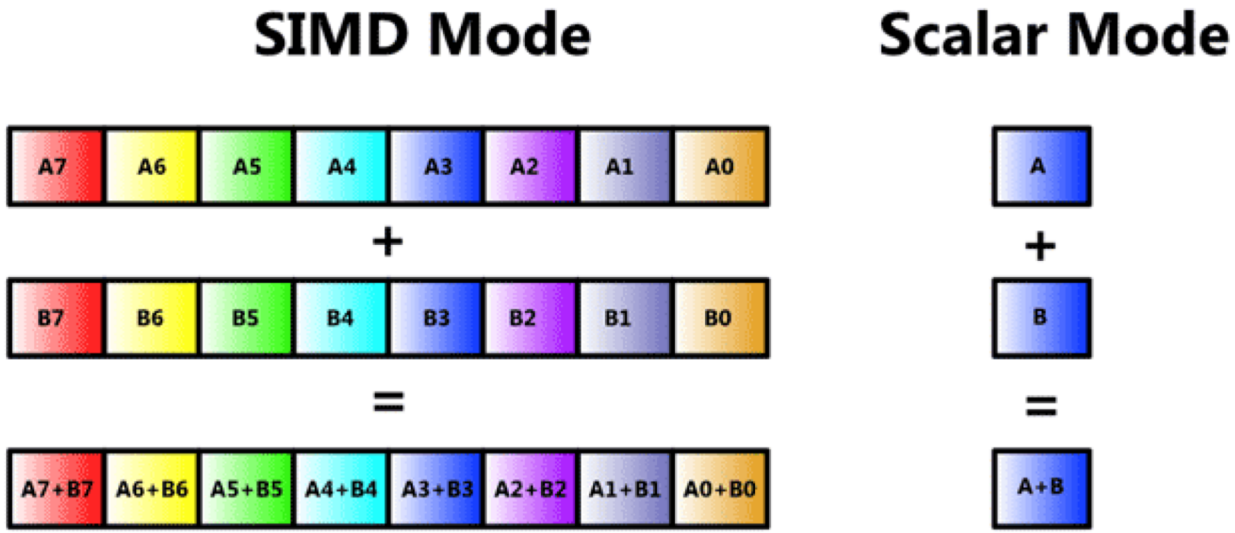

① Front-end
Instruction Predecode & Fetch (16 bytes)
6 MOPs
Instruction Queue (2x25 entries)
Macro Fusion
5 MOPs
MS ROM
μCode Sequencer
5-way Decoder
Complex
Simple
Simple
Simple
Simple
3-4 μOPs
μOP
μOP
μOP
μOP
Branch Prediction Unit
Return Stack Buffer
Branch Target Buffer
μOP Cache 1.5K μOPs 8-way
5 μOPs
4 μOPs
6 μOPs
Multiplexer
Stack Engine
Adders
Allocation Queue (2x64 entries)
Micro Fusion
③ Memory Pipeline
L1i Cache 32 kiB 8-way
L1 TLB
Instruction μOP Cache Tags
L2 TLB
L2 Cache 256 kiB 4-way
L1d TLB
L1d Cache 32 kiB 8-way
Line Fill Buffer (10 entries)
loads
Load Buffer (72 entries)
stores
Store & Forward Buffer (56 entries)
branches
Branch Order Buffer (48 entries)
μOPs
Re-order Buffer (224 entries)
Retirement Unit
Physical Register File
Integer Registers (180 entries)
Vector Registers (168 entries)
Register Alias Table
Primary RAT
Shadow RAT
4 μOPs
6 μOPs
Register Allocation & Renaming
② Out-of-Order Engine
Execution Units
STORE
AGU
LOAD
AGU
LOAD
INT ALU
INT DIV
IVEC ALU
IVEC MUL
FP FMA
AES
VEC STR
FP DIV
Branch
AGU
INT ALU
Branch
INT ALU
VEC SHUFFLE
IVEC ALU
LEA
INT ALU
INT MUL
IVEC ALU
IVEC MUL
FP FMA
Bit Scan
port 0
port 2
port 3
port 4
port 7
port 6
port 5
port 1
Common Data Buses
μOP Scheduler
Unified Reservation Station (97 entries)
CPU Core
CPU Core
CPU Core
CPU Core
LLC Slice
LLC Slice
LLC Slice
LLC Slice
System Agent
Display Controller
PCIe
Memory Controller

Instruction cycle



Machine Learning/Computer Vision

– – –



OS Kernel

— — —

- BIOS/UEFI
- Architectures - (Bootloader, BIOS data structures, Multiprocessing, Interrupts, Exceptions, ACPI, Hypervisor, Timer, Memory/IO mapped, PCI(e), Syscalls, Context switching)
- Device drivers - (Network, USB, Storage - IDE/ATA, Keyboard)
- File system
- Process scheduler - Scheduler types:
    - Long term scheduler (which process are to be admitted to the ready queue - CPU or I/O bound)
    - Medium term scheduler (transfer to second storage from main memory)
    - Short term scheduler (schedule after interrupt)
  - Preemptive multitasking (Completely fair scheduler, Priority scheduling, Round-robin/time slicing)
- Memory Management (TLB, Paging, Segmentation)
- PXE (painful :-D)

Perf tuning/Troubleshooting

– – –

Code coverage - Cyclomatic complexity - Asymptotic notation - Big O

VM optimization

Memory management

Profiling - Metric

PMU/PT/LBR - Processor Feature

BCC

Linux perf

SystemTap

strace/dtrace/ltrace

Perf Tuning ninja - SWOT analysis

– – –

Handle 100k data in Python.

vagrant@ubuntu-xenial:/vagrant/Documents/lang/python$ python db.py
The function took 6.10998797417 seconds to complete.

vagrant@ubuntu-xenial:/vagrant/Documents/lang/python$ python eminus.py
The function took 0.0119049549103 seconds to complete

Performance Tuning Ninja 😎😎😎

Security Trends

SecOps

Reverse Engineering

Web App security