OS Initialization Review: From the Electron to the Boot

Prof. Antônio Augusto Fröhlich

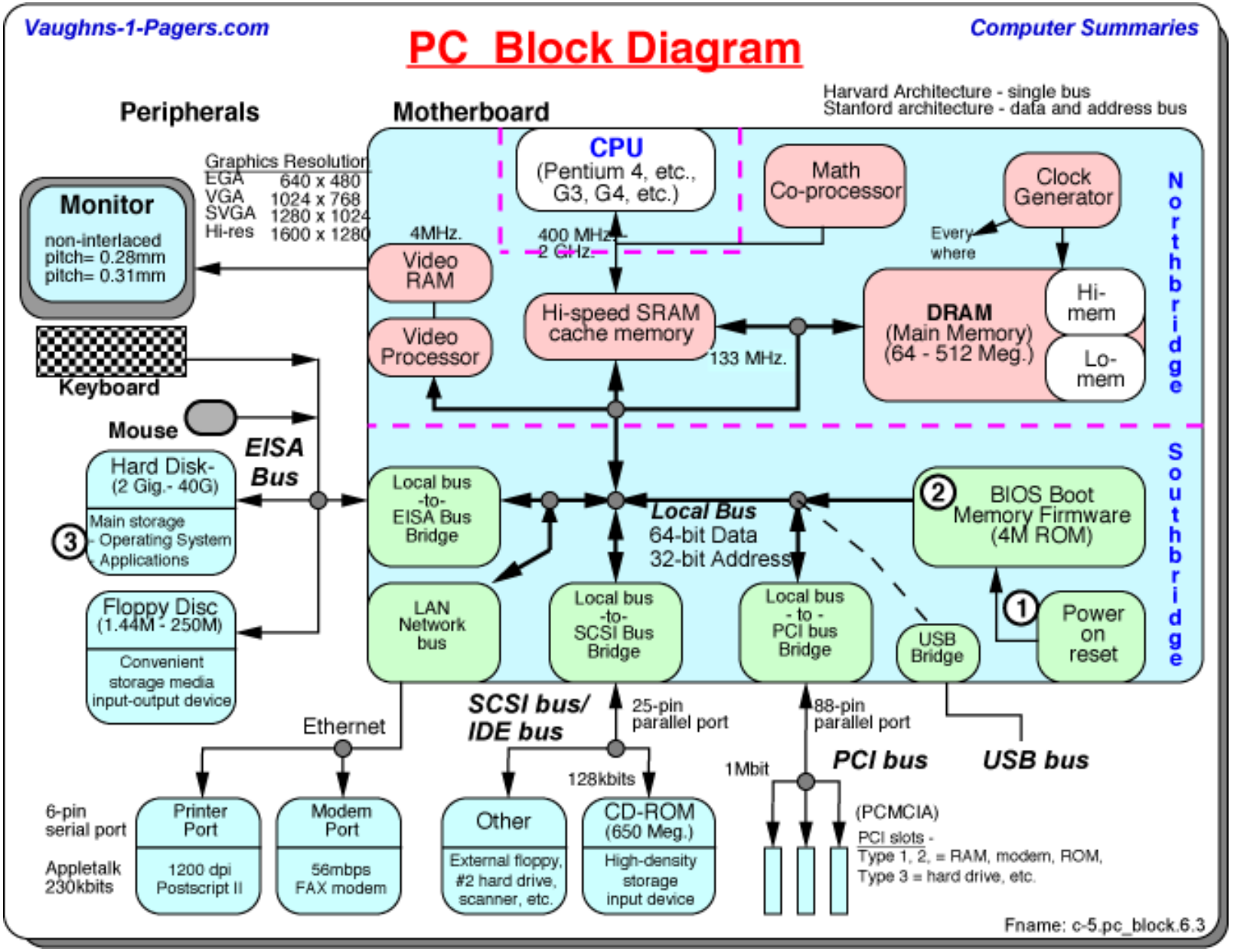
UFSC / LISHA

https://lisha.ufsc.br/~guto

What is an OS at all?

- Habermann: “Everything between hardware and applications”
- OS handles abstraction levels
  - From electrons to Java and Python applications
- OS developer
  - knowledge of computer architecture
  - notions of electronic circuits
  - outstanding programming skills
- How will you learn this?
  - Deep diving into a PC operating system

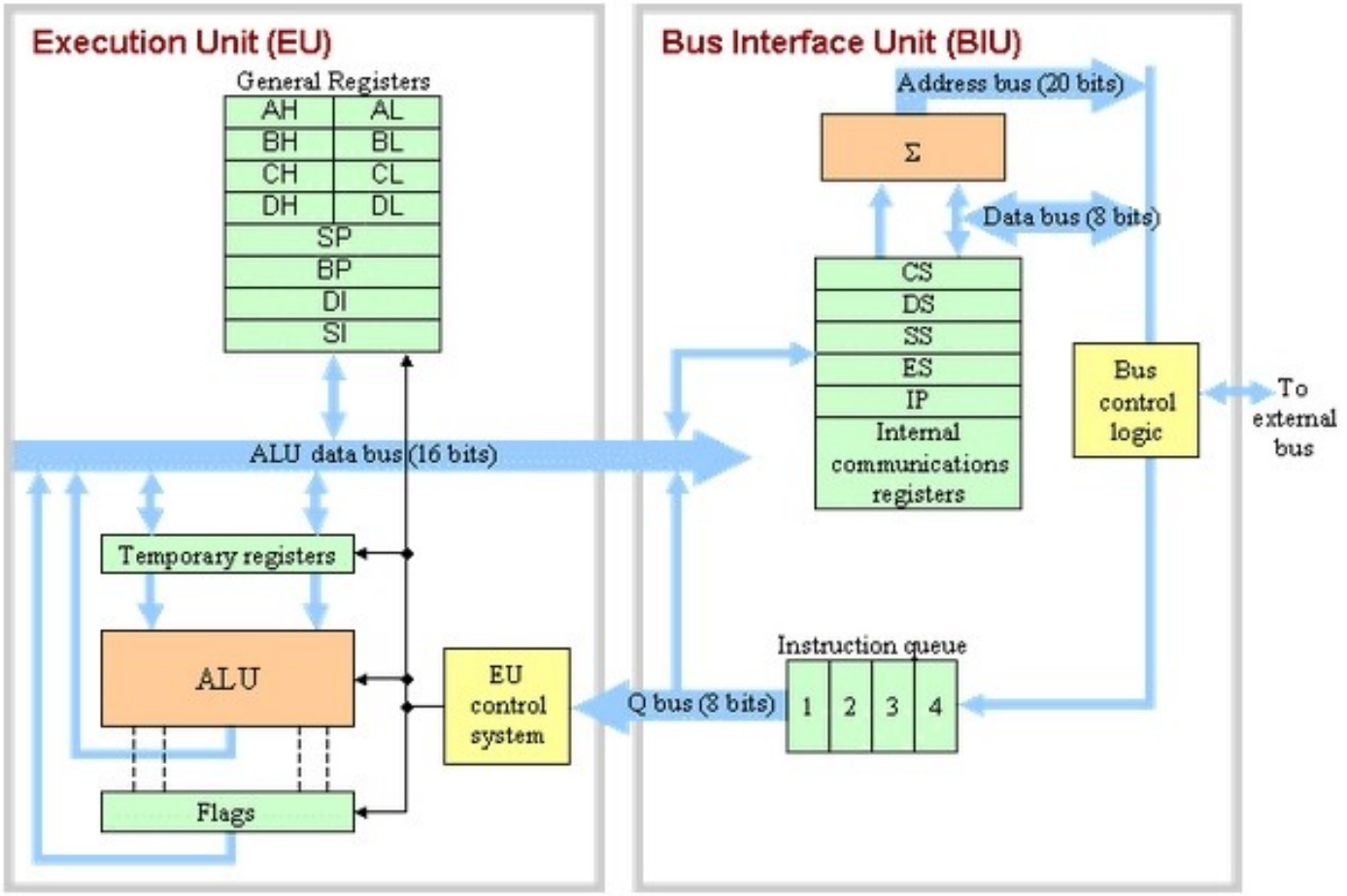
The PC


Vaughns-1-Pagers.com
PC Block Diagram
Computer Summaries
Harvard Architecture - single bus
Stanford architecture - data and address bus
Peripherals
Motherboard
Graphics Resolution
EGA 640 x 480
VGA 1024 x 768
SVGA 1280 x 1024
Hi-res 1600 x 1280
Monitor
non-interlaced
pitch= 0.28mm
pitch= 0.31mm
Keyboard
Mouse
EISA Bus
Hard Disk-
(2 Gig.- 40G)
Main storage
- Operating System
Applications
Floppy Disc
(1.44M - 250M)
Convenient
storage media
input-output device
Ethernet
6-pin
serial port
Appletalk
230kbits
Printer Port
1200 dpi
Postscript II
Modem Port
56mbps
FAX modem
CPU
(Pentium 4, etc.,
G3, G4, etc.)
4MHz.
400 MHz -
2 GHz.
Math
Co-processor
Clock
Generator
Every
where
Video
RAM
Video
Processor
Hi-speed SRAM
cache memory
133 MHz.
DRAM
(Main Memory)
(64 - 512 Meg.)
Hi-
mem
Lo-
mem
Northbridge
Local bus
-to-
EISA Bus
Bridge
Local Bus
64-bit Data
32-bit Address
BIOS Boot
Memory Firmware
(4M ROM)
LAN
Network
bus
Local bus
-to-
SCSI Bus
Bridge
Local bus
- to -
PCI bus
Bridge
USB
Bridge
Power
on
reset
Southbridge
SCSI bus/
IDE bus
25-pin
parallel port
88-pin
parallel port
1Mbit
PCI bus
USB bus
128kbits
Other
External floppy,
#2 hard drive,
scanner, etc.
CD-ROM
(650 Meg.)
High-density
storage
input device
(PCMCIA)
PCI slots -
Type 1, 2, = RAM, modem, ROM,
Type 3 = hard drive, etc.
Fname: c-5.pc_block.6.3

Starting the Machine

- When the PC's Power button is pushed
  - Start the SMPS (switched-mode power supply)
    - Integrity depends on precise voltage regulation
    - SMPS stabilizes and then
  - CPU comes to a cold-start
    - It needs to be initialized (booted)
    - Most common architecture today: Intel's x86
    - Why?
      - IBM: x86 and DOS to replace Motorola's 68k and CPM16

Intel 8088 Architecture



-- Static registers (groups of D Flip-Flops) used to hold or transfer binary data

-- Logic gate circuits designed to perform arithmetic or logical functions

-- Logic gate circuits designed to provide internal control to processor

-- Internal data busses used to pass information between components

Source: CSedukit.com

“Modern” x86 Architecture

- Limitations
  - No register banks
  - No internal bus
    - cx => IO
    - bx => ULA
  - Backward compatibility
- Will live with that
  - x86 is super-scalar
  - Microcode
    - Communication to the internal architecture
    - Speculative, out-of-order execution
    - Non-determinism

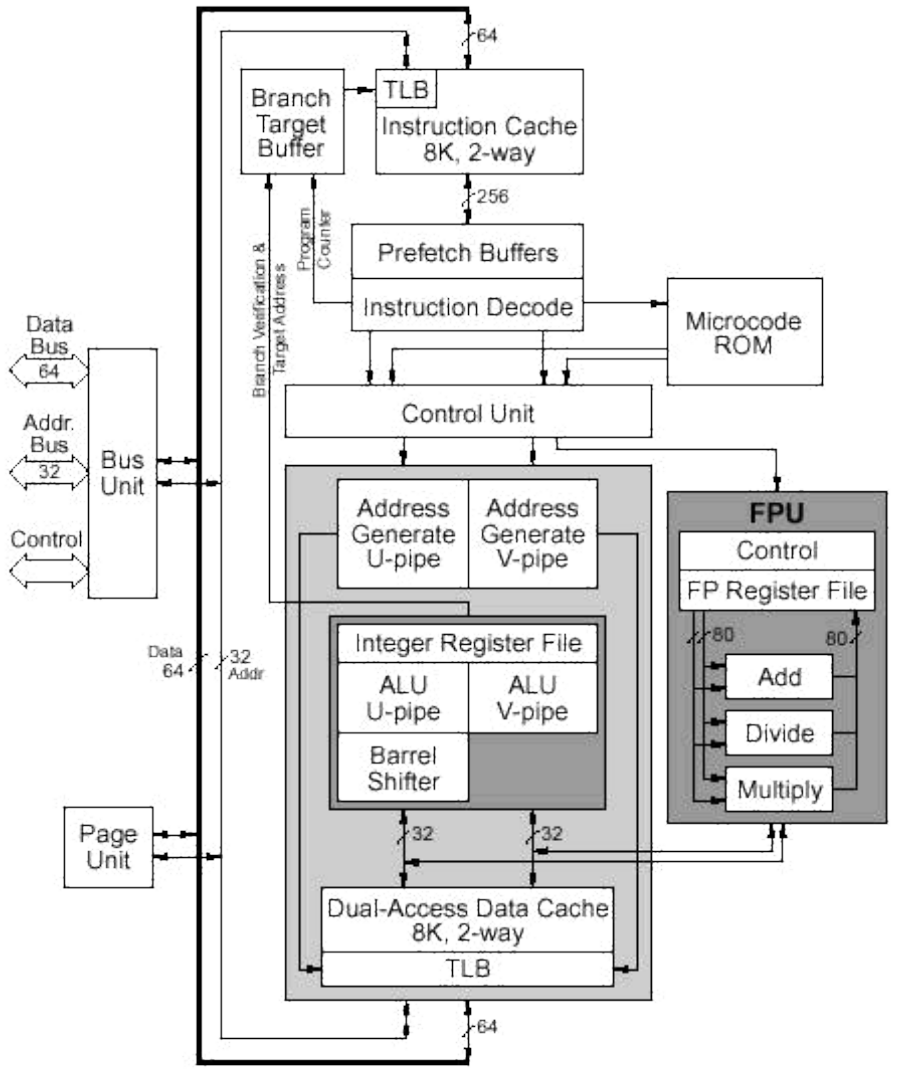
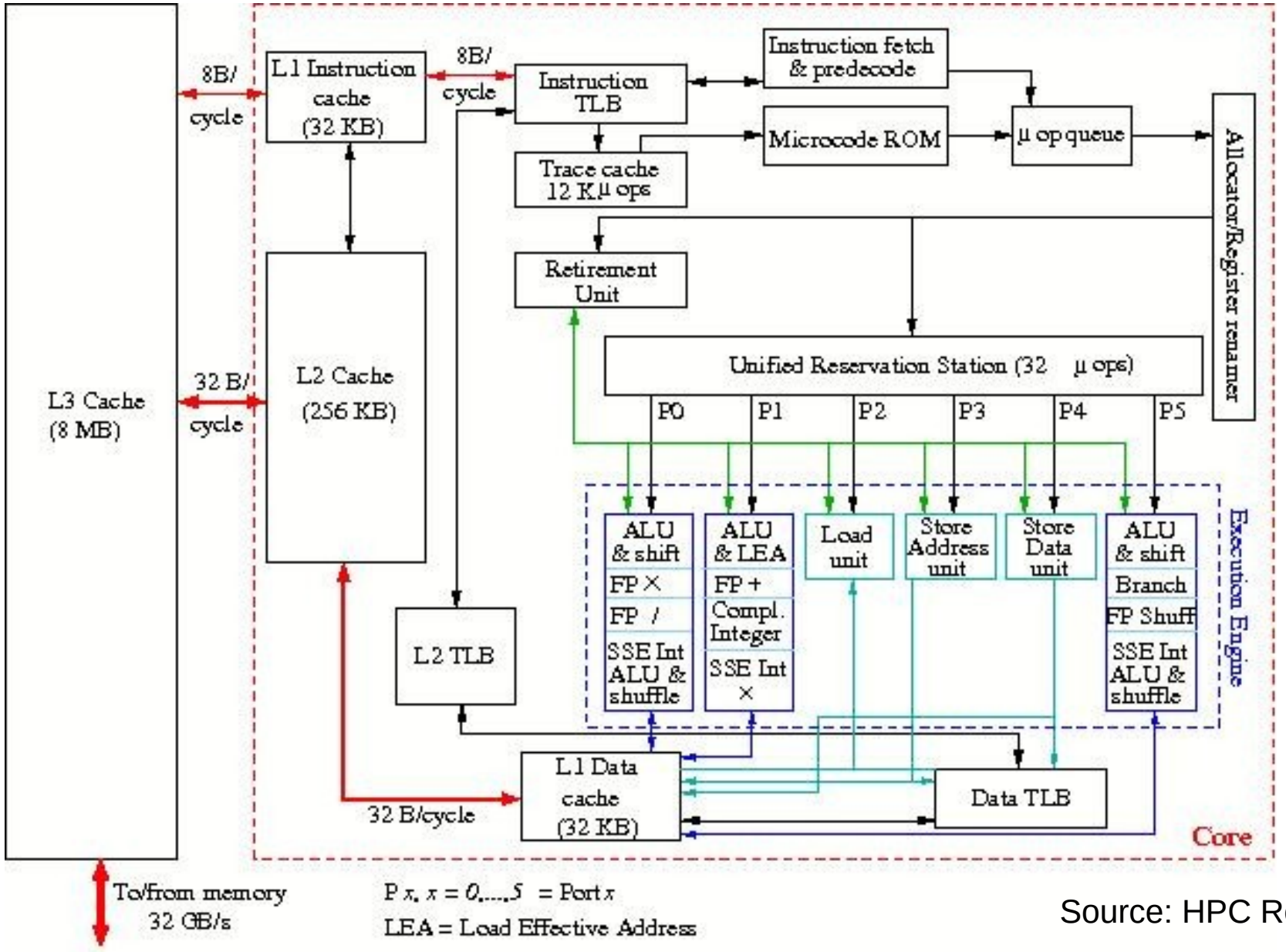


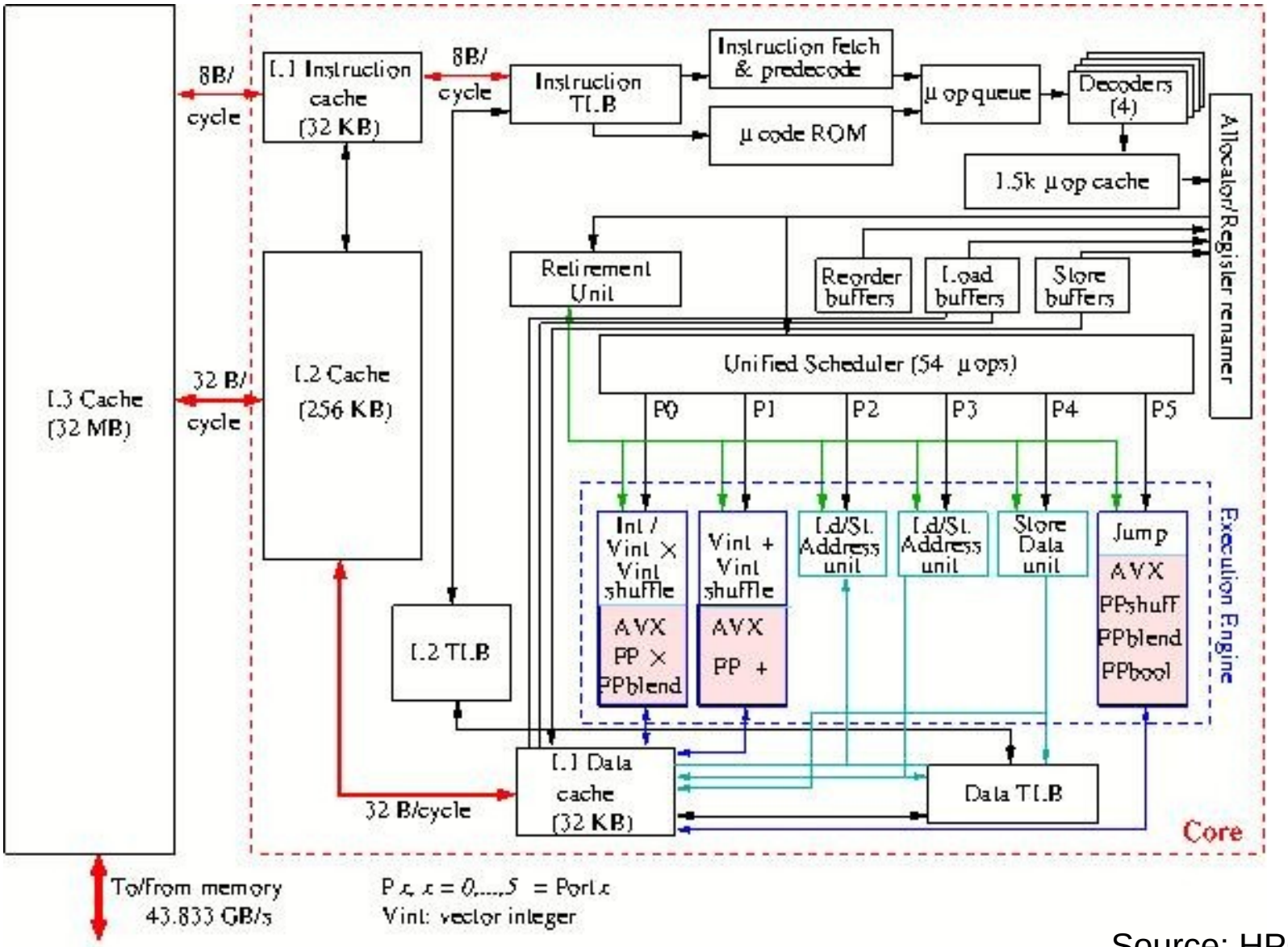
Figure 1. Pentium block diagram.

Nehalem



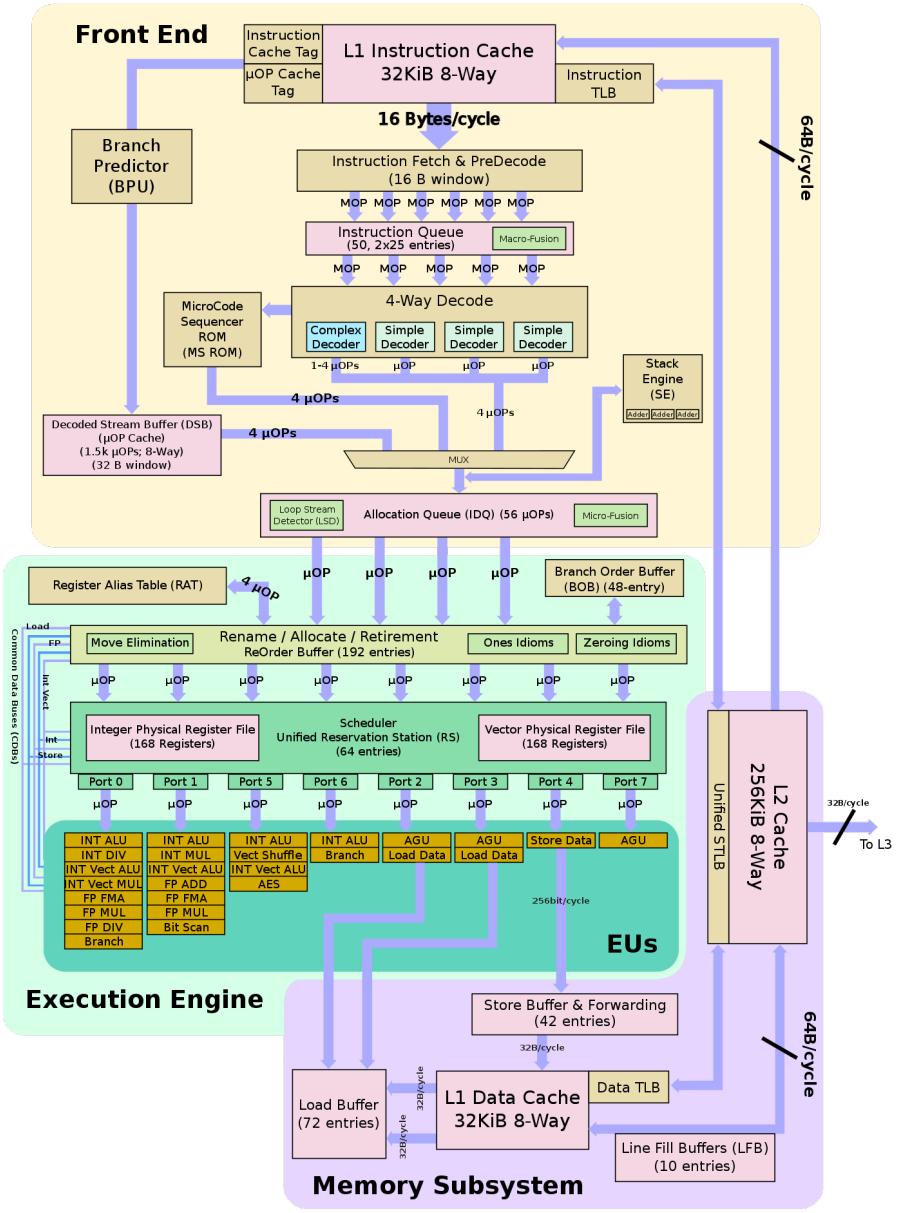
Source: HPC Research

Sandy Bridge



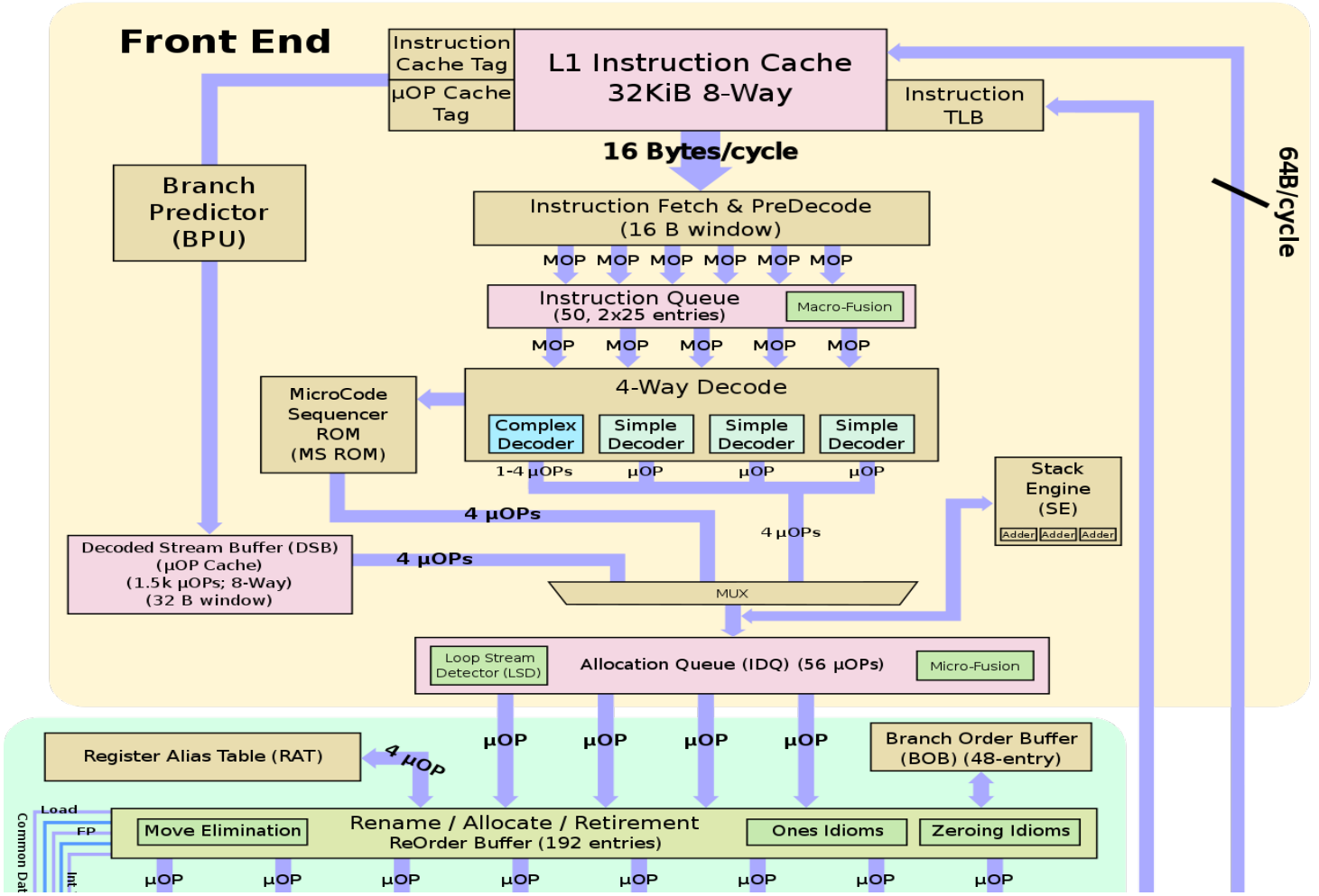
Source: HPC Research

Broadwell



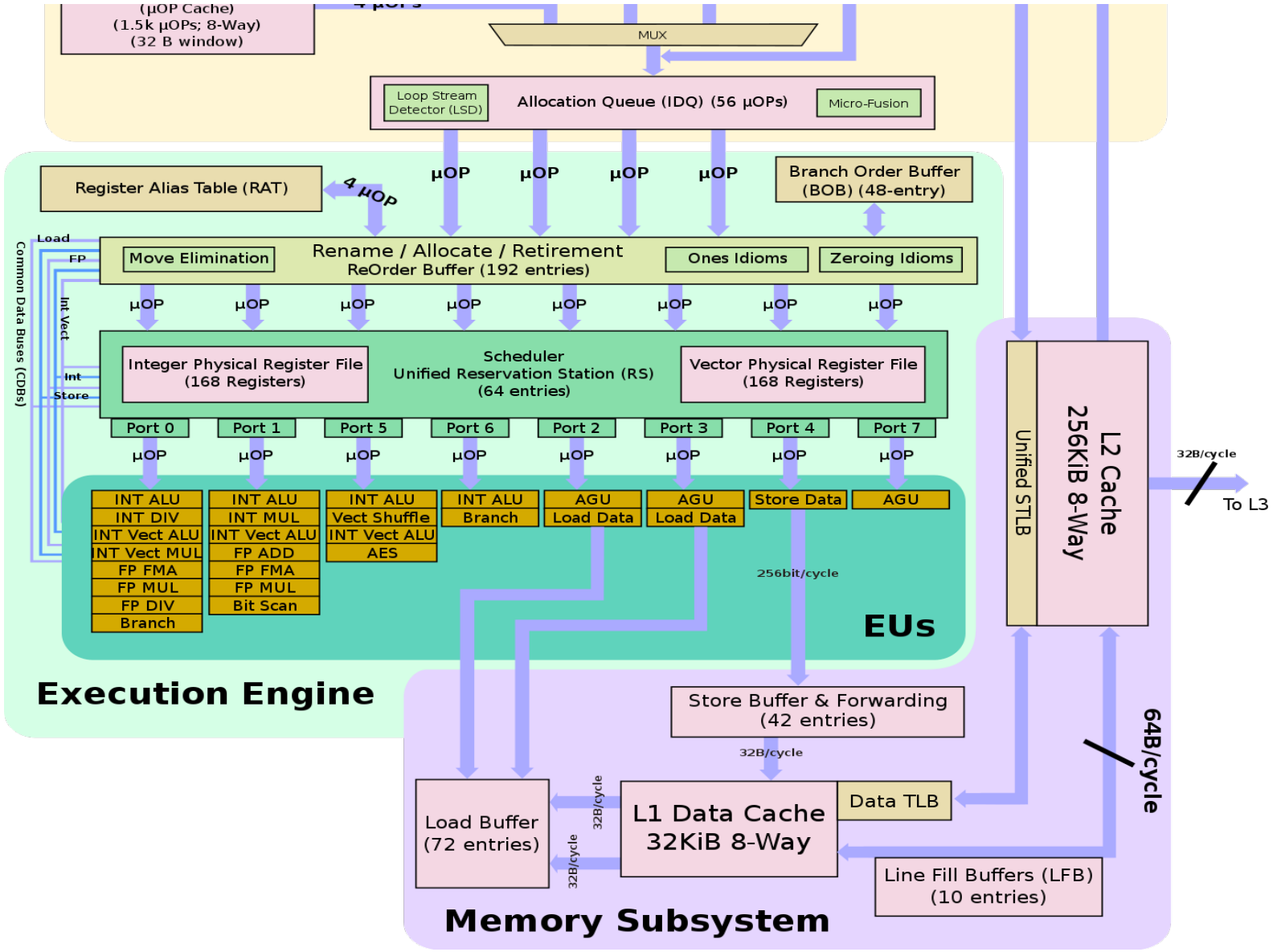
Source: Wikichip

Broadwell



Source: Wikichip

Broadwell


(µOP Cache)
(1.5k µOPs; 8-Way)
(32 B window)
MUX
Loop Stream Detector (LSD)
Allocation Queue (IDQ) (56 µOPs)
Micro-Fusion
Register Alias Table (RAT)
4 µOP
µOP
µOP
µOP
µOP
Branch Order Buffer (BOB) (48-entry)
Load
FP
Int Vect
Int
Store
Common Data Buses (CDBs)
Move Elimination
Rename / Allocate / Retirement
ReOrder Buffer (192 entries)
Ones Idioms
Zeroing Idioms
Integer Physical Register File (168 Registers)
Scheduler
Unified Reservation Station (RS)
(64 entries)
Vector Physical Register File (168 Registers)
Port 0
Port 1
Port 5
Port 6
Port 2
Port 3
Port 4
Port 7
INT ALU
INT DIV
INT Vect ALU
INT Vect MUL
FP FMA
FP MUL
FP DIV
Branch
INT ALU
INT MUL
INT Vect ALU
FP ADD
FP FMA
FP MUL
Bit Scan
INT ALU
Vect Shuffle
INT Vect ALU
AES
INT ALU
Branch
AGU
Load Data
AGU
Load Data
Store Data
AGU
256bit/cycle
EUs
Execution Engine
Unified STLB
L2 Cache
256KiB 8-Way
32B/cycle
To L3
Store Buffer & Forwarding
(42 entries)
32B/cycle
64B/cycle
Load Buffer
(72 entries)
32B/cycle
32B/cycle
L1 Data Cache
32KiB 8-Way
Data TLB
Line Fill Buffers (LFB)
(10 entries)
Memory Subsystem

BIST – Built-In Self Test

- Microcode (firmware) ran at boot
  - Check the integrity of functional units
  - Turns on stable units
    - division bug on the Pentium I processor
    - handling of failures in production process
- BIST generates report on hardware status
  - At the BIOS, 1048 (or more) bits
  - Some things may be turned on and off by micro-fuses
- After the BIST
  - Processor ready to call first software instruction

Booting “For Real”

- Intel's Basic Architecture Manual
  - Section 9.4.1 – First Instruction Executed
  - `0xfffffff0` (`0xf.fff0`)
  - In Real Mode (8086)
  - It is something like “jmp #BIOS_ADDR”
  - Why at the “top” (1MB)?
    - 20 bit physical bus
  - Why 16-bytes below (instead of 16 bits)?
    - Segmentation for 8086 → shift by 4
- It allows different sizes of BIOS memories
- Flexibility for system developers

BIOS POST

- POST – Power-On Self Test
  - What comes first? BIOS or VGA?
    - Hooks for peripheral initialization
    - VGA comes first
  - Initializes legacy peripherals
    - Keyboard, serial, parallel ports, buses
    - South-bridge - ISA (timing legacy)
  - Initialize remaining things (new stuff)
  - Memory test - write-read-compare procedure
    - A few chips feature smart controllers (self-test)
    - For others run test until it fails (memory top reached)
  - POST report status to NVRAM (CMOS)
    - At internal RTC
    - No standard report – useless for generic OS

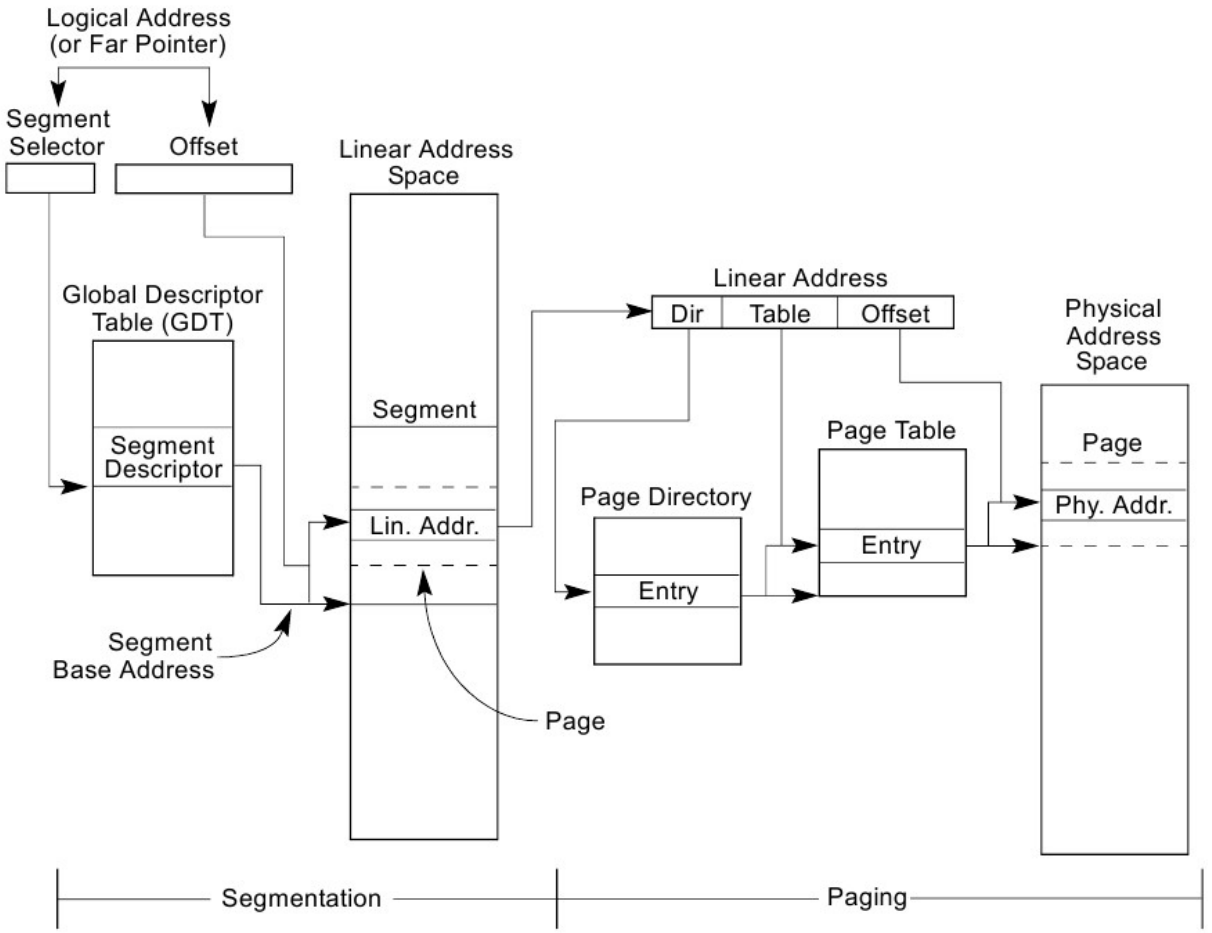
Initializing the machine

- After POST, BIOS initialization code
  - Run in Real Mode (8086)
  - BIOS is therefore 16-bits code
    - Useless for modern OS
    - Drivers re-implemented with 32-bits code by OS
- Why are BIOS still in use?
  - Hardware bugs “workarounds” at BIOS
- Final initialization hooks
  - Auxiliary boot
    - e.g., network (remote boot)
    - USB is not here, BIOS emulates it as a disc
  - If no auxiliary boot
    - Load the first sector of the first detected disk at `0x7c00`
      - the 512-bytes long MBR – Master Boot Record
      - containing the bootstrap (and eventually a partition table)

EPOS x86 Bootstrap

- 16-bit code
  - as86, ld86
- Load the OS from disc to RAM
- Enter Protected mode (32 or 64 bits)
- From this point on system is functional
  - May execute “generic”, 32 bits, compiled code
- Although lots of architecture-specific configuration still needs to be performed...

x86 Architecture Legacy Overview



- MMU (Memory Management Unit)
  - Paging X Segmentation
  - Internal X external fragmentation
  - Unfinished 8086 => (CS << 4) + offset