Advanced x86:
BIOS and System Management Mode Internals
Memory Map

Xeno Kovah && Corey Kallenberg

LegbaCore, LLC

All materials are licensed under a Creative Commons “Share Alike” license.

http://creativecommons.org/licenses/by-sa/3.0/

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

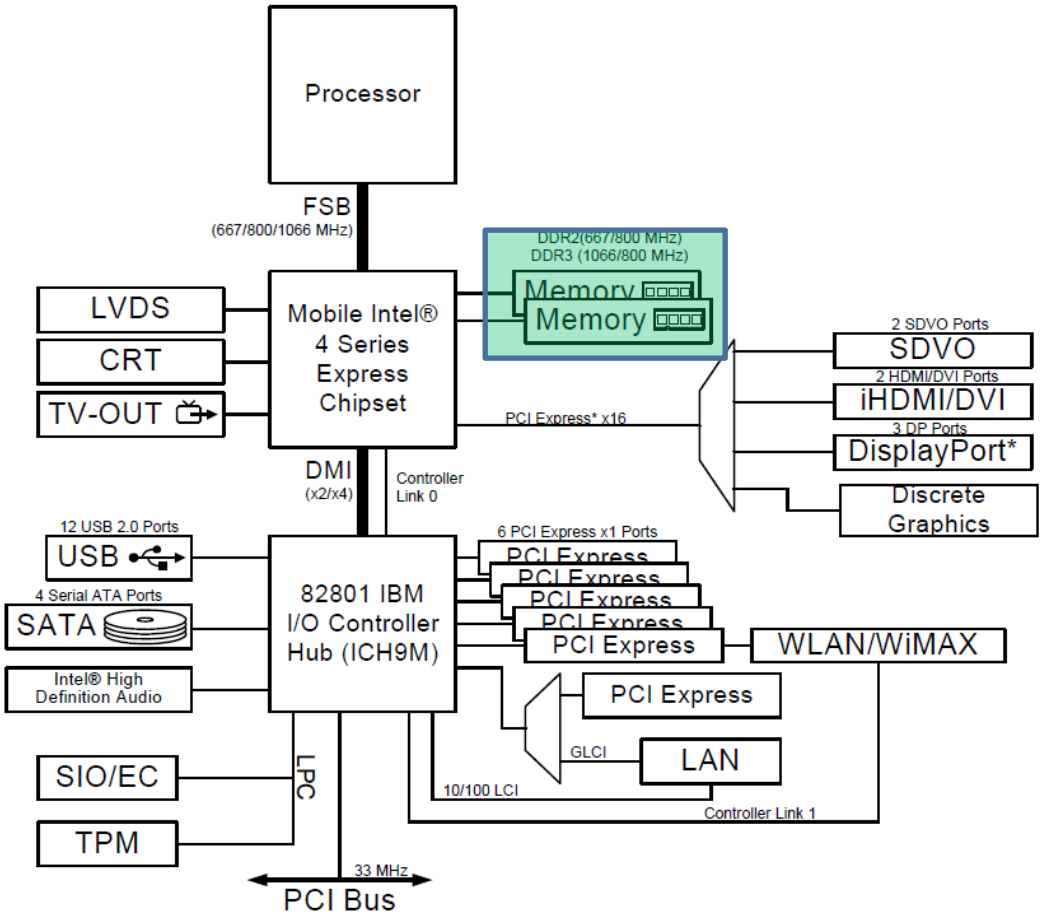
Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

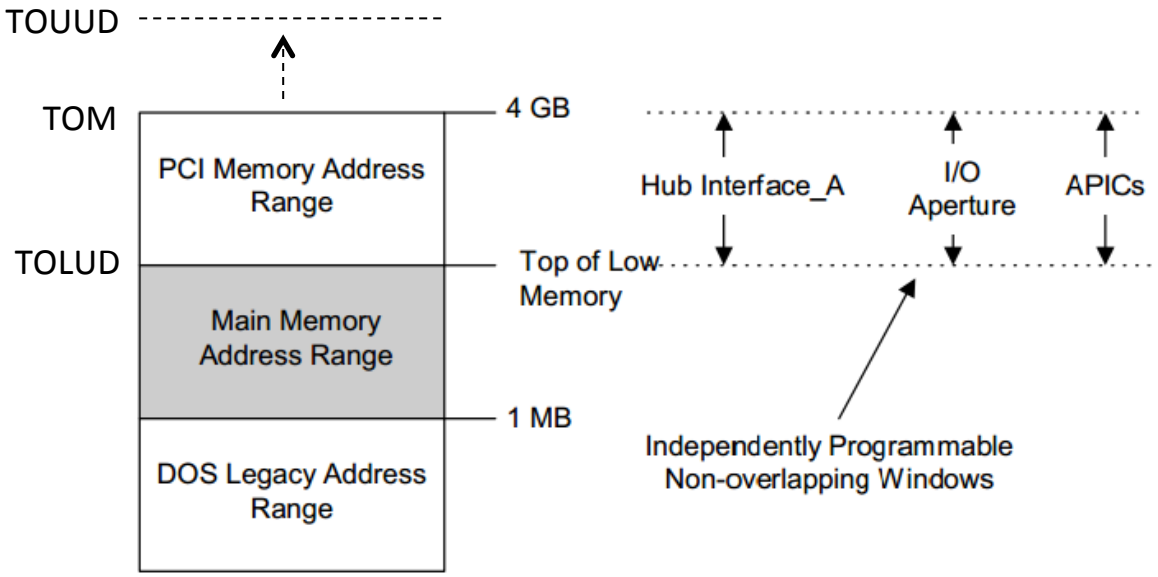
Attribution condition: You must indicate that derivative work
"Is derived from John Butterworth & Xeno Kovah’s ’Advanced Intel x86: BIOS and SMM’ class posted at http://opensecuritytraining.info/IntroBIOS.html”

Memory Map



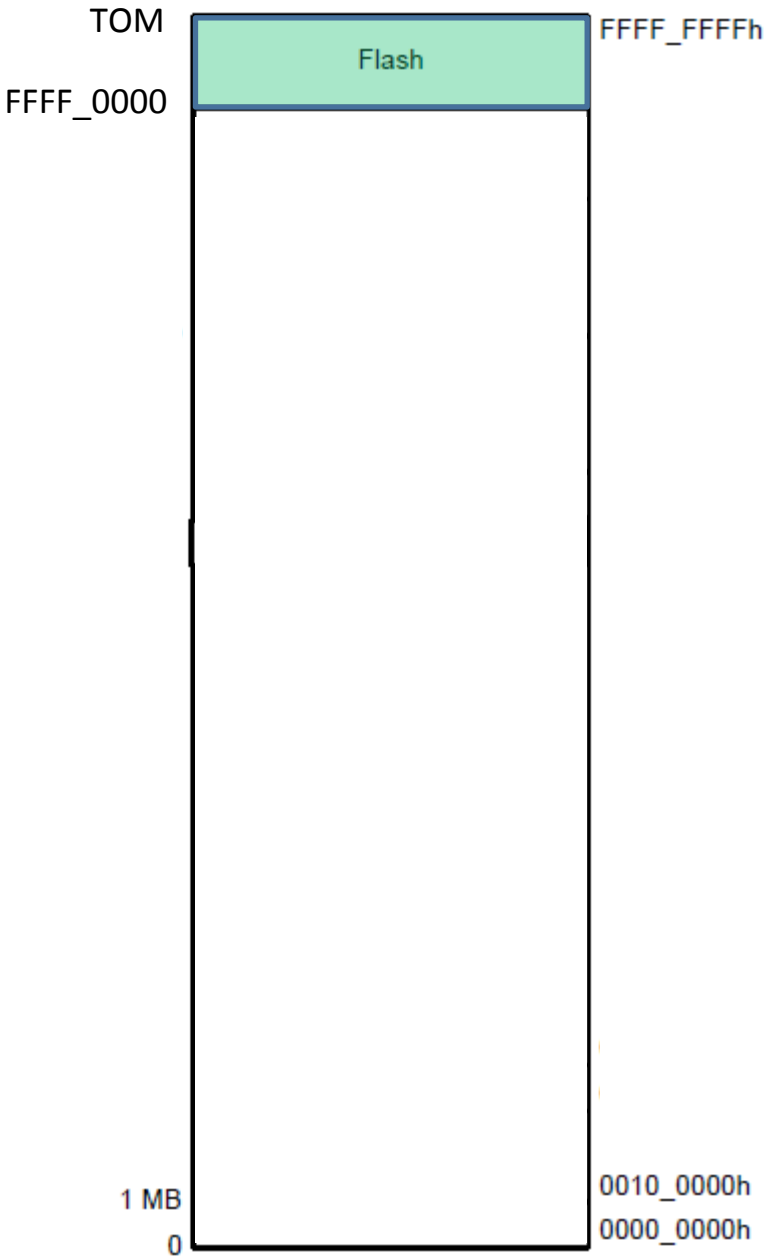
- One of the primary responsibilities of the BIOS is to program the memory map
- Many devices, in order to be useful, require their interfaces be extended to memory
- Also this is how the BIOS can ensure information about the way it set up the system is passed to the operating system at the time of handoff

4 “Basic” Ranges in System Memory



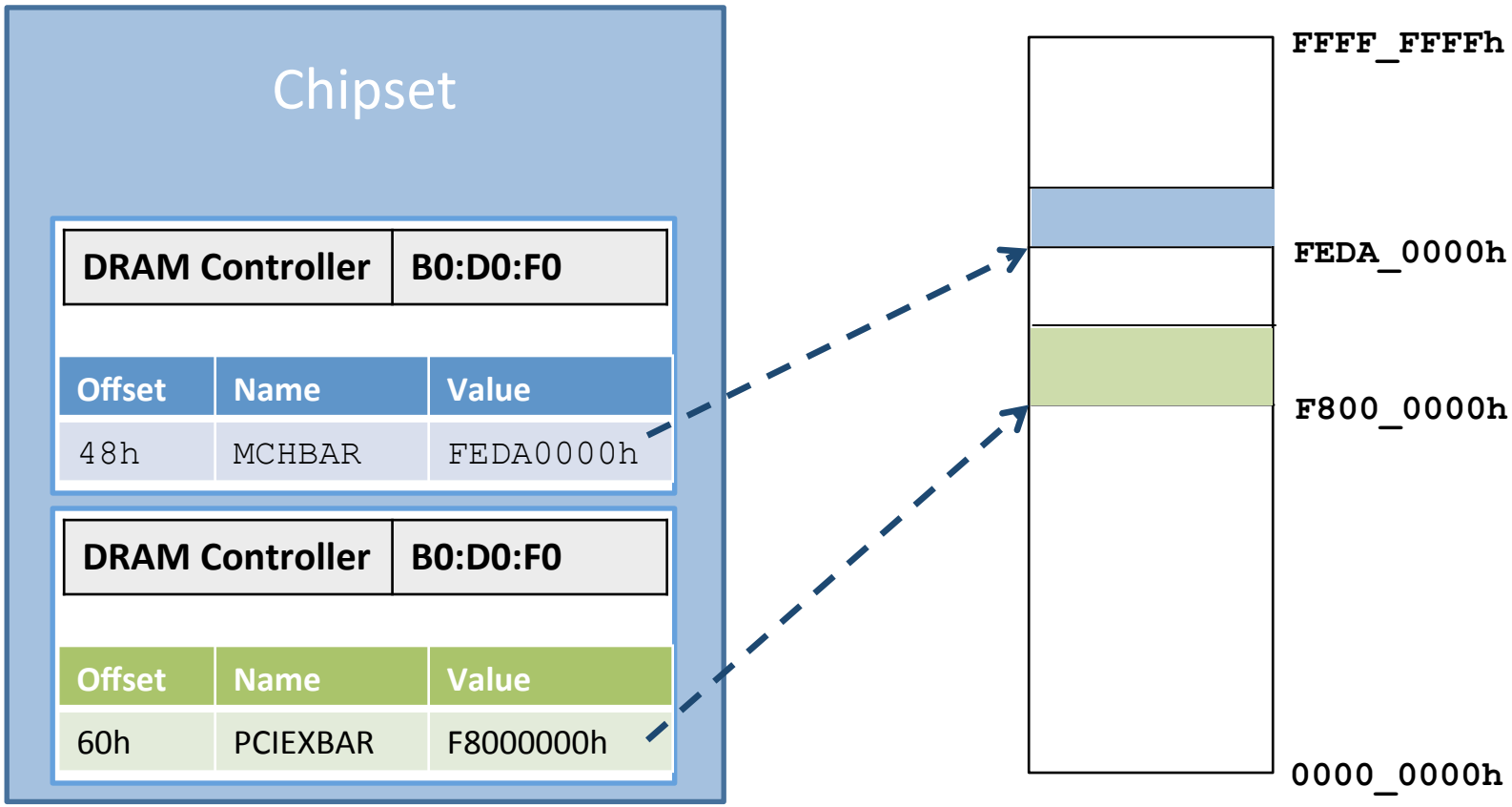
1. High Memory Range: Memory above 4GB (called Top of Upper Usable DRAM). Used for memory mapping and recoverable memory (system memory that overlaps with the PCI range)
   – TOM (Top of Upper Memory): size of physical memory
2. PCI Memory Address Range: Used for memory-mapped IO (TPM, APIC, Flash, PCI Express, devices on chipset, etc.)
3. Main Memory Address Range: Addressable memory from TOLUD (Top of Low Usable DRAM) down to 1 MB
4. Compatible Memory space: 1 MB and below

Memory Map



- But on startup the processor is only aware of one memory range as we've seen
  - Often called the Boot Block, it contains the entry vector and uncompressed BIOS code
- The system automatically maps the top 16 MB of memory to the flash bios
  - Non-negotiable, does not matter if your system has < 4 GB of memory, the system never actually accesses that memory. Rather, it is mapped to the flash device.
- The rest of system memory needs to be configured by the BIOS

Hardware Block Diagram



- On the Mobile 4-Series Chipset, the BIOS (executed by the CPU), configures the MCHBAR in the DRAM Controller
- FEDA_0000h (on an E6400 with 4GB RAM for example)
- MCHBAR is now added to the memory map
- So how does this actually occur?