

Registers to configure memory remap (must be locked) : LOCK (bit 0) = 1  
 >> to define using of DRAM (not just SMRAM)

- [+] PCI0.0.0\_BDSM = 64 bits - Base of Graphics Stolen Memory
- [+] PCI0.0.0\_BGSM = 64 bits - Base of GTT Stolen Memory
- [+] PCI0.0.0\_DPR = 64 bits - DMA Protected Range
- [+] PCI0.0.0\_GGC = 64 bits - Graphics Control
- [+] PCI0.0.0\_MESEG\_MASK = 64 bits - Manageability Engine Limit Address Register
- [+] PCI0.0.0\_PAVPC = 64 bits - PAVP Configuration
- [+] PCI0.0.0\_REMAPBASE = 64 bits - Memory Remap Base Address
- [+] PCI0.0.0\_REMAPLIMIT = 64 bits - Memory Remap Limit Address
- [+] PCI0.0.0\_TOLUD = 64 bits - Top of Low Usable DRAM
- [+] PCI0.0.0\_TOM = 64 bits - Top of Memory
- [+] PCI0.0.0\_TOUUD = 64 bits - Top of Upper Usable DRAM
- [+] PCI0.0.0\_TSEGMB = 64 bits - TSEG Memory Base

## SMRAM Protection

SMRAMc or System Management  
RAM Controller Register (CPU)  
on 8 bits

bit 3 > G\_SMRAME = 1  
 >> to enable D\_\* bits

bit 4 > D\_LCK = 1  
 >> to forbidden access to SMRAM from ring 0  
 >> to set in read only mode for  
 several important bits of SMRAM registers  
 (D\_OPEN, G\_SMRARE, C\_BASE\_SEG, H\_SMRAM\_EN,  
 GMS, TOLUD, TOM, TSEG\_SG, TSEG\_EN)

D\_OPEN = 0  
 >> just to initialize SMM (by BIOS code)

ESMRAMC or Extended System  
Management RAM Control  
on 8 bits (0xab)

bit 0 > T\_EN or TSEG\_EN = 1  
 >> to enable TSEG (protection against  
 DMA attacks)

SMRR or System Management  
Range Registers (CPU) :  
IA32\_SMRR\_PHYSBASE  
IA32\_SMRR\_PHYSMASK

Range to protect with SMRR  
 >> to prevent modification  
 of SMRAM properties from ring 0  
 i.e: injection SMI code into memory cache

TSEGMB or TSEG Memory  
Base Register (CPU)  
on 32 bits (0xabcdeffgh)

>> to prevent DMA attack

bits 31:20 > TSEGMB <=> SMRAM  
 >> TSEG range covers entire SMRAM

bit 0 > LOCK = 1  
 >> TSEG range is locked