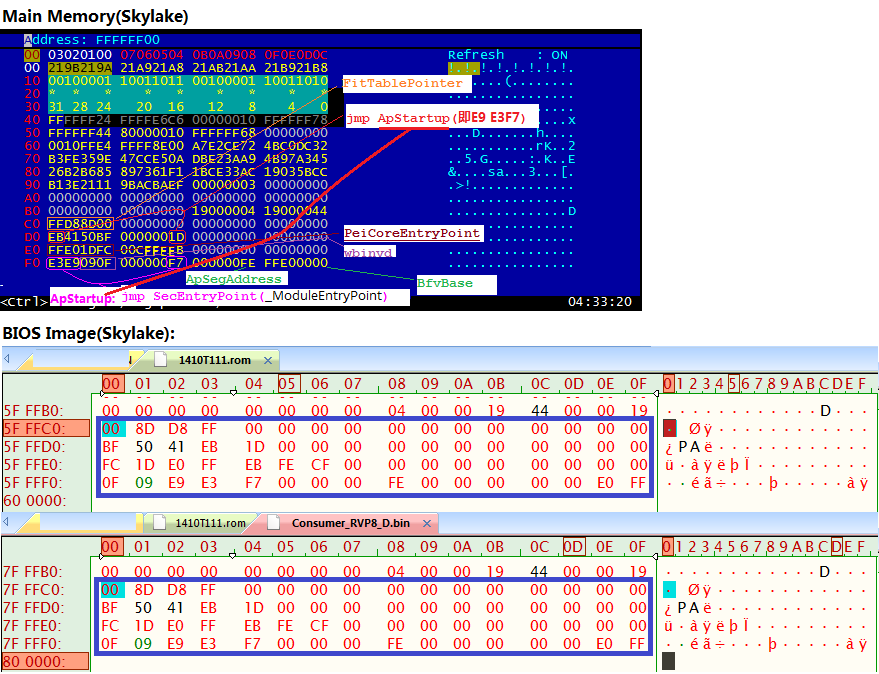
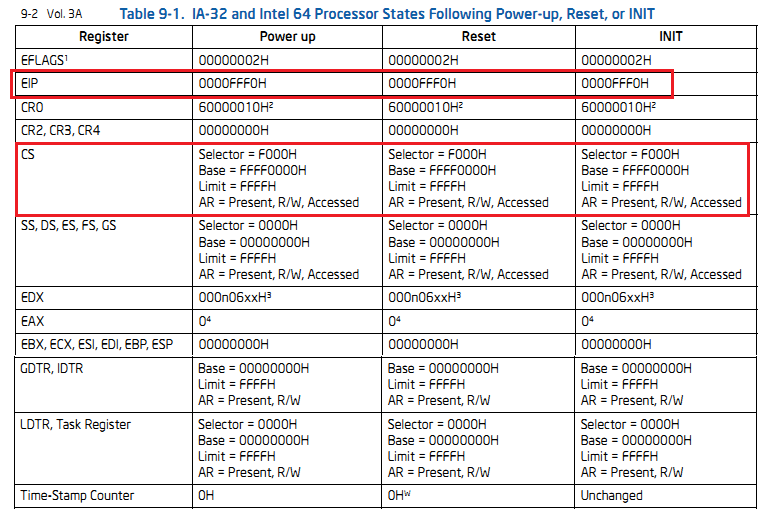
**根据BIOS Image分析SecPhase**

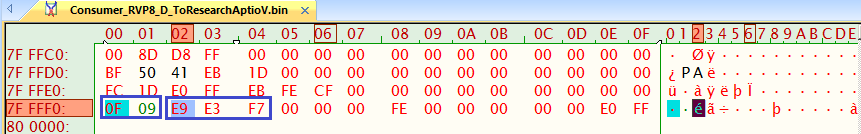
1. 说明
   1. 基于Skylake PCH110 BIOS（公版RVP8 SKL-S Q0/S0 D0/D1 DDR4）
   2. BIOS size是8MB（布局：Descriptor Region、GbE、ME、BIOS），其中Skylake BIOS的细分布局如下图



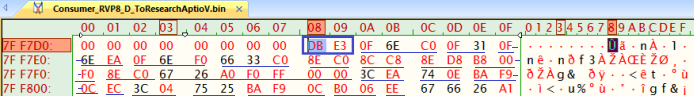
1. Sec执行流程
   1. Processor Reset Vector

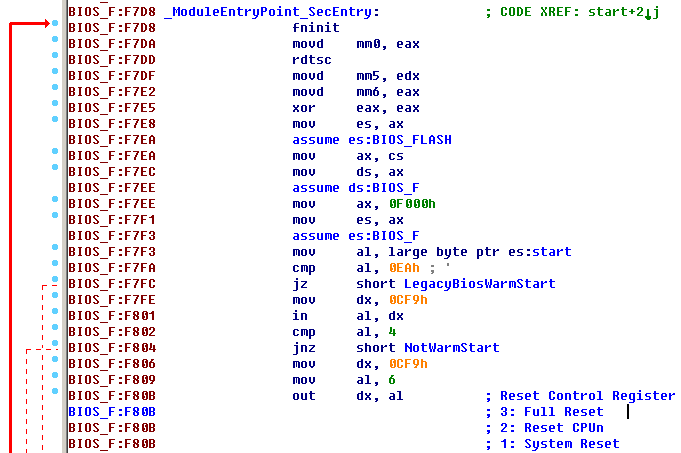


* + 1. 上电后（电源、时序等正常），寄存器EIP=**0x0000 fff0**、CS Selector= **0xf000**（CS Base=**0xffff 0000**），即在实模式下从**0xf000<<4 | 0xfff0** --换算成32位--> **0xffff fff0**（对应8MB BIOS固件地址是0x7f fff0，如下仅用BIOS固件地址分析）开始执行。[源文件：IA32FamilyCpuPkg\SecCore\Ia32\ResetVec.asm16](../../../IA32FamilyCpuPkg/SecCore/Ia32/ResetVec.asm16)

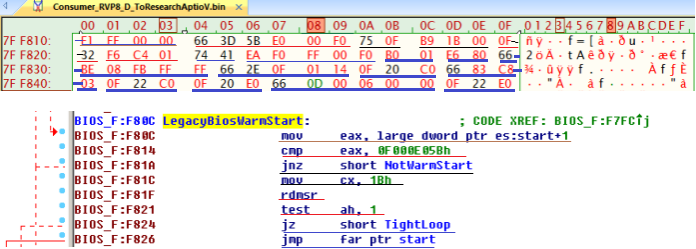


* + 1. 第1条指令：0f 09，即wbinvd（writeback cache and invalid cache）。
    2. 第2条指令：e9 e3 f7，即jmp $+0xf7e3（**跳转到**0x7f fff2 + 0x3 + “带符号的0xf7e3” = 0x7f fff2 + 0x00 0003 - 0x00 081d = **0x7f f7d8**）,跳转到SecEntryPoint。
  1. [SecEntryPoint（源文件：AmiSkylakeCpuPkg\Library\ProcessorStartup\Ia32\Flat32.asm）](../../../AmiSkylakeCpuPkg/Library/ProcessorStartup/Ia32/Flat32.asm)

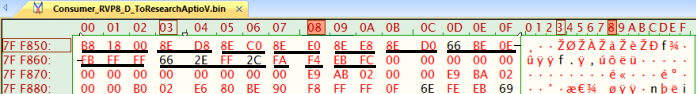


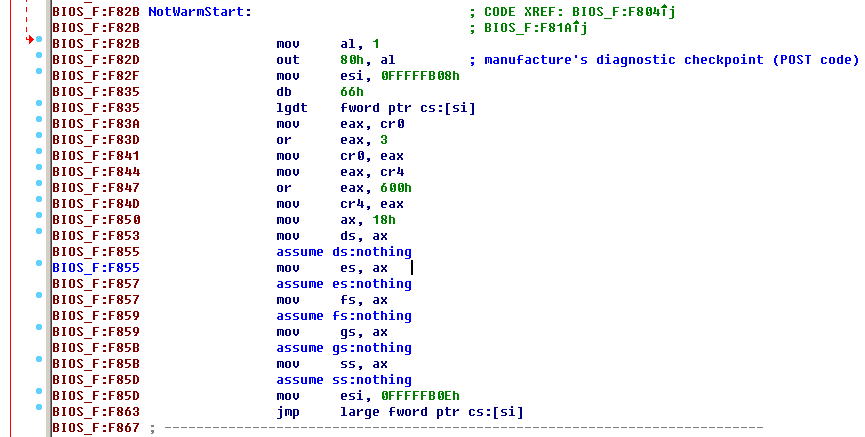


* + 1. SecPhase的入口是**\_ModuleEntryPoint PROC NEAR C PUBLIC**，在本BIOS Image中的地址是**0x7f f7d8**。
    2. **0x7f f7d8**：db e3，即fninit（clear any pending Floating point exceptions）
    3. **0x7f f7da**：0f 6e c0，即movd mm0, eax（Save time-stamp counter value）
    4. **0x7f f7dd**：0f 31 ，即rdtsc（load 64bit time-stamp counter to EDX:EAX）
    5. **0x7f f7df**：0f 6e ea，即movd mm5, edx（backup）
    6. **0x7f f7e2**：0f 6e f0，即movd mm6, eax（backup）
    7. **0x7f f7e5**：66 33 c0，即xor eax, eax （AMI\_OVERRIDE\_START >>> for detecting cpu only reset, if happened, change to chipset reset）
    8. **0x7f f7e8**：8e c0，即mov es, ax ; seg\_selector = 0,
    9. **0x7f f7ea**：8c c8，即mov ax, cs
    10. **0x7f f7ec**：8e d8，即mov ds, ax
    11. **0x7f f7ee**：b8 00 f0，即mov ax, MAGIC\_SEG ; MAGIC\_SEG EQU 0F000h
    12. **0x7f f7f1**：8e c0，即mov es, ax ; MAGIC\_ADDRESS\_IN\_SEG EQU 0FFF0h
    13. **0x7f f7f3**：67 26 a0 *f0 ff 00 00*，即mov al, BYTE PTR **es:**[*MAGIC\_ADDRESS\_IN\_SEG*]
    14. **0x7f f7fa**：3c ea，即cmp al, 0EAh
    15. **0x7f f7fc**：74 0e，即jz LegacyBiosWarmStart
    16. **0x7f f7fe**：ba f9 0c，即mov dx, 0CF9h
    17. **0x7f f801**：ec，即in al, dx
    18. **0x7f f802**：3c 04，即cmp al, 04h
    19. **0x7f f804**：75 25，即jnz NotWarmStart
    20. **0x7f f806**：ba f9 0c，即mov dx, 0CF9h
    21. **0x7f f809**：b0 06，即mov al, 06h
    22. **0x7f f80b**：ee，即out dx, al

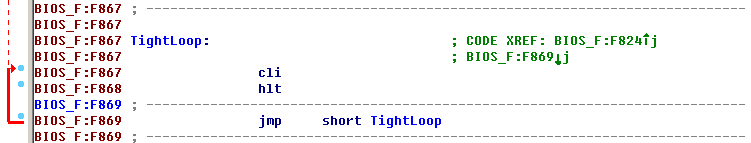


* + 1. **0x7f f80c**：67 66 26 a1 f1 ff 00 00，即mov eax, DWORD PTR **es:**[MAGIC\_ADDRESS\_IN\_SEG + 1]
    2. **0x7f f814**：66 3d 5b e0 00 f0，即cmp eax,0f000e05bh
    3. **0x7f f81a**：75 0f，即jne NotWarmStart
    4. **0x7f f81c**：b9 1b 00，即mov cx, MSR\_APIC\_BASE
    5. **0x7f f81f**：0f 32，即rdmsr
    6. **0x7f f821**：f6 c4 01，即test ah, 1(APIC\_BASE\_MSR.BIT8=1表明是BSP，否则是AP)
    7. **0x7f f824**：74 41，即jz TightLoop（若是AP，则跳转到死循环；若是BSP，则跳转到0xfff0）
    8. **0x7f f826**：，即DB 0EAh, DW MAGIC\_ADDRESS\_IN\_SEG, DW MAGIC\_SEG





* + 1. **0x7f f82b**：80 01，即mov al, 0x1
    2. **0x7f f82d**：e6 80，即out 80h, al
    3. **0x7f f82f**：66 be 08 fb ff ff，即 mov esi, OFFSET GdtDesc
    4. **0x7f f835**：66 2e f0 01 14，即DB 66h; lgdt fword ptr **cs:**[si]
    5. **0x7f f83a**：0f 20 c0，即mov eax, cr0
    6. **0x7f f83d**：66 83 c8 03，即or eax, 00000003h(PE bit (bit #0) & MP bit (bit #1))
    7. **0x7f f841**：0f 22 c0，即mov cr0, eax (Activate protected mode)
    8. **0x7f f844**：0f 20 e0，即mov eax, cr4
    9. **0x7f f847**：66 0d 00 06 00 00，即or eax, 00000600h (Set OSFXSR(bit #9, FXSAVE & FXRSTOR) & OSXMMEXCPT(bit #10,SIMD))
    10. **0x7f f84d**：0f 22 e0，即mov cr4, eax
    11. **0x7f f850**：b8 18 00，即mov ax, SYS\_DATA\_SEL(in Protected16, so Set up the selectors for protected mode entry)
    12. **0x7f f853**：8e d8，即mov ds, ax
    13. **0x7f f855**：8e c0，即mov es, ax
    14. **0x7f f857**：8e e0，即mov fs, ax
    15. **0x7f f859**：8e e8，即mov gs, ax
    16. **0x7f f85b**：8e d0，即mov ss, ax
    17. **0x7f f85d**：66 be 0e fb ff ff，即mov esi, offset NemInitLinearAddress
    18. **0x7f f863**：66 2e ff 2c，即jmp fword ptr **cs:**[si](Protected32)



* + 1. **0x7f f867**：fa，即cli
    2. **0x7f f868**：f4，即hlt
    3. **0x7f f869**：eb fc，即jmp TightLoop

备注：以下不是代码

