

# ASM9260T 开发笔记

## ASM9260T 的启动流程：

芯片上电复位后，最先运行的是位于 0xFFFF0000 地址开始的 BootCode，它使用 [0x40001000,0x40001FFF]范围的片内 SRAM 作为 Data 段和 Stack。BootCode 检测 GPIO2\_[5,6,7]三个管脚，决定从哪里（例如：SPI-Flash、IIC-EEPROM、Nand-Flash、Nor-Flash）加载 sysloader 到 SRAM 的[0x40000000,0x40000FFF]内存，然后跳转到 sysloader 的入口向量。所以设计的 sysloader 代码必须小于 4KB，一旦 sysloader 运行起来，就可以使用[0x40001000,0x40001FFF]范围的内存作为 sysloader 的 Data 段和 Stack，因为 BootCode 使命已经完成。sysloader 负责初始化系统 PLL，进而设定 CPU 时钟、HCLK 时钟；然后初始化片内 32MB 的 SDRAM，并从设定的永久存储器读取程序镜像到 SDRAM，最后跳转到位于 SDRAM 中的程序入口向量地址执行。

通常，我们将 sysloader 存放在 SPI-Flash 的 0x0000 首部位置，将应用程序镜像放在 SPI-Flash 的 0x10000 偏移处；而应用程序的执行地址安排在 SDRAM 的 0x20008000 开始处。应用程序运行后，将 64 字节异常向量表拷贝到 0x20000000 处，并利用 MMU 将 0x20000000 开始的 1MB 空间映射到虚拟地址 0x00000000。[0x20004000,0x20007FFF]的 16KB 内存，用于存放 MMU 地址转换表，16KB 可存放整个一级页表。

由于 sysloader 仅有 4KB 的代码空间，很难实现复杂的功能；我们需要在 sysloader 和应用程序之间，再设计一个 bootloader，用于实现复杂的以太网 ISP，升级，加密，调试等功能。执行的顺序是：BootCode→sysloader→bootloader→application。地址安排如下表：

程序名称	SPI-Flash 加载地址	执行地址	备注
BootCode	-----	0xFFFF0000	Code 小于 16KB
sysloader	0x00000000	0x40000000	Code 小于 4KB
bootloader	0x00010000	0x20008000	Code 小于 64KB
application	0x00020000	0x20020000	与 SDRAM 容量有关

Sysloader 运行过程中，始终在 SVC 模式执行；不开任何中断，禁用 MMU，禁用 Icache 和 Dcache。

Bootloader 运行环境与 application 很接近。但必须在跳转到 application 之前，回到 SVC 模式，关闭所使用过的外围设备时钟源(除了 DMA 和 SDRAM)及其中断；并冲刷 Dcache，清除 Icache；等等。总而言之，就是要让 application 似乎是被 sysloader 所加载的一样，就好像没有 bootloader 执行过一样。

