# InitBoot

#define USB\_SPACE\_HYGX(PhysicalDiedId,UsbIndex,Address) (UINT32)(0x10000000+( PhysicalDiedId<<28)+(0x600000\*UsbIndex)+Address)  
  
Typedef FCH\_RESET\_DATA\_BLOCK{

FCH\_XHCI

}  
  
FchTSXhciInitBootProgram  
{

UINT32 CurrentPowerState

1.1Write PMCSR.PowerState in USB PCie head to D0

PMCSR 是 ​PCI/PCIe 设备电源管理的核心寄存器

1.2等待CurrentPowerState为D0

2.USB\_S5\_RESET=1

3.USB\_SOFT\_CONTROLLER\_RESET=1

USB 控制器中的一个 ​软件复位控制位，用于通过软件触发 USB 控制器的完全复位

4.wait at least 100us

5.Platform/Customer based host configuration :if there is needed to program USB container register to change USB features or mode,do in this step.Followubg are some examples  
a. to change USB port numbers,program Poart Control(0x0A58\_0130),and Port Disable Write Once(0x0A58\_0128) and/or Port Disable Read Write (0x0A58\_012C)

6. Program internal registers

7 Program USB\_S5\_RESET=0

8.Wait at least 200us

9. Update USB3.1 PHY FW

10.Program USB\_SOFT\_CONTROLLER\_RESET=0

11.Program internal register

12.reset usb PHY

}

Enable Socket 0 XHCI-0 controller

FchTSXhciInitBootProgram ...Start.

当前操作的是第 ​0 号 USB 控制器​（可能为主控制器或首个实例）。

​CURRENT\_PWR\_STS=0x0​：

控制器当前电源状态为 ​D0 状态（完全供电）​​ 或 ​未进入低功耗模式。

0x0 在不同平台可能表示不同的电源管理状态（需结合芯片手册），但通常表示正常工作模式。

UsbIndex=0, CURRENT\_PWR\_STS=0x0.

​**触发函数​**：FchTSXhciPlatformPortDisable

​**功能​**：检查或配置 USB 端口的禁用状态。

​**总线地址​**：Bus 0x0

当前操作的 USB 总线为 ​0 号总线​（通常是主板上的主总线）。

​**端口控制号​**：PortCtrlNum 0x46

**拆分字段：**

​Usb2PortCtrlNum=0x6​：控制 USB 2.0 端口的参数（定义编号为 0x6）。

​Usb3PortCtrlNum=0x4​：控制 USB 3.0 端口的参数（定义编号为 0x4）。

0x46 是原始十六进制值（转换二进制为 01000110），可能通过位掩码组合 USB2/USB3 的控制字段。

​端口禁用状态​：UsbPortDisable 0x0

​所有端口均未禁用​（禁用标志位全为 0），此时端口处于 ​启用状态。

FchTSXhciPlatformPortDisable on Bus 0x0 - PortCtrlNum 0x46(Usb2PortCtrlNum=0x6, Usb3PortCtrlNum=0x4) - UsbPortDisable 0x0

FchTSXhciPlatformPortDisable on Bus 0x0 - PortCtrlNum 0x46(Usb2PortCtrlNum=0x6, Usb3PortCtrlNum=0x4) - UsbPortDisable 0x0

**​操作流程​：**

​**轮询（Polling）​​：**持续检查 SRAM 初始化完成标志（sram\_init\_done）。

​**断言（Assert）​​：**确认外部加载完成（sram\_ext\_ld\_done），即配置数据成功写入 SRAM。

**流程详解​**

**​​(1) SRAM 初始化（sram\_init\_done）​​**

​目的​：

初始化 PHY 内部的 SRAM，确保其可读写并准备好接收配置数据。

​硬件行为​：

复位 SRAM 控制器。

校准时序和电压（如 DDR 接口训练）。

设置 SRAM 的初始状态（清除错误标志）。

​轮询机制​：

固件循环读取 PHY 寄存器中的 sram\_init\_done 标志位，直到其置 1。

**​​(2) 外部配置加载（sram\_ext\_ld\_done）​​**

​目的​：

将外部存储的配置数据（如固件、通道训练参数）加载到 SRAM 中。

**​数据来源​：**

主板 SPI 闪存中的固件镜像。

CPU 微码更新（Microcode Patch）。

**​断言触发​：**

写入配置数据后，固件向 PHY 发送信号，硬件设置 sram\_ext\_ld\_done 标志。

Polling NBIO 0 Bus (0x0) PHY 0 sram\_init\_done

Assert bus (0x0) PHY 0 sram\_ext\_ld\_done

Polling NBIO 0 Bus (0x0) PHY 1 sram\_init\_done

Assert bus (0x0) PHY 1 sram\_ext\_ld\_done

Polling NBIO 0 Bus (0x0) PHY 2 sram\_init\_done

Assert bus (0x0) PHY 2 sram\_ext\_ld\_done

Polling NBIO 0 Bus (0x0) PHY 3 sram\_init\_done

Assert bus (0x0) PHY 3 sram\_ext\_ld\_done

**​usbIndex=0​：**

当前操作的 USB 控制器索引号为 0（通常为第一个 XHCI 控制器实例）。

**​smnAddr​：**

通过 ​**系统管理网络（SMN）**​​ 访问的硬件寄存器地址，用于**配置 XHCI 控制器的功能**。

​0x16C0D060​：可能为端口配置寄存器（如设置端口数量、传输模式）。

​0x16C0D0E0​：可能为中断控制寄存器（如中断向量分配）。

​0x16C0D160​：可能为 DMA 引擎配置寄存器。

​0x16C0D1E0​：可能为电源管理寄存器。

usbIndex=0 smnAddr=(0x16C0D060), Cnt=(18)

usbIndex=0 smnAddr=(0x16C0D0E0), Cnt=(0)

usbIndex=0 smnAddr=(0x16C0D160), Cnt=(0)

usbIndex=0 smnAddr=(0x16C0D1E0), Cnt=(0)

Enable Socket 0 XHCI-1 controller

# 当进入S3时