

## STM32H7 双核 CM4 作为 Master 初始化系统

关键字: STM32H7 Dual Core, CM4, Master

### 1. 引言

STM32H7 双核单片机内部集成了 CM7 和 CM4 这两个内核，在目前官方提供的例程中，大都是使用 CM7 作为 Master 初始化系统时钟，然后通过释放硬件信号量 HSEM 唤醒 CM4，那么是否可以将 CM7 与 CM4 的角色互换，让 CM4 作为 Master 去初始化系统时钟呢？

### 2. 可行的方案分析

答案肯定是可以的，初步先介绍如下两种方案。

#### 2.1 方案一

目前官方提供的例程路径如下，可以使得 D2 域的 CM4 进行系统时钟的初始化：

```
.....\STM32Cube\Repository\STM32Cube_FW_H7_V1.9.0\Projects\STM32H747I-EVAL\Examples\PWR\PWR_D2ON_D1OFFEWARM
```

但是通过查看对应项目工程的 readme.txt 文件，发现：

**IN THIS EXAMPLE CM4 BOOT OPTION SHOULD BE DEACTIVATED BY USING ST LINK UTILITY TO GUARANTEE THAT CM4 WILL NOT BOOT AUTOMATICALLY. TO DEACTIVATE CM4 BOOT GO TO STLINK\_UTILITY->TARGET->OPTION\_BYTES THEN DESELECT BOOT CM4 FROM THE "USER CONFIGURATION OPTION BYTE" MENU.**

也就是需要事先手动的设置修改用户选项字节 Option bytes，然后再进行操作。比较麻烦。

#### 2.1 方案二

那有没有第二种方法，方案二呢？参考我们的 CubeMx 的生成代码的示例，对于 CM4 和 CM7 内核角色的互换，是不是简单的把两者的代码直接互换，并将下图中的 RCC\_FLAG\_D2CKRDY 换成 RCC\_FLAG\_D1CKRDY，将 PWR\_D2\_DOMAIN 换成 PWR\_D1\_DOMAIN 即可？

测试发现这种直接互换的方式是不行的!

```

/* USER CODE BEGIN Boot_Mode_Sequence_1 */
/* Wait until CPU2 boots and enters in stop mode or timeout*/
timeout = 0xFFFF;
while(( _HAL_RCC_GET_FLAG(RCC_FLAG_D2CKRDY) == RESET) && (timeout-- > 0));
if ( timeout < 0 )
{
    Error_Handler();
}
/* USER CODE END Boot_Mode_Sequence_1 */
/* MCU Configuration-----*/
/* Reset of all peripherals, Initializes the Flash interface and the SysTick. */
HAL_Init();
/* Configure the system clock */
SystemClock_Config();
/* USER CODE BEGIN Boot_Mode_Sequence_2 */
/* When system initialization is finished, Cortex-M7 will release Cortex-M4 by means of
HSEM notification */
/*HW semaphore Clock enable*/
__HAL_RCC_HSEM_CLK_ENABLE();
/*Take HSEM */
HAL_HSEM_FastTake(HSEM_ID_0);
/*Release HSEM in order to notify the CPU2(CM4)*/
HAL_HSEM_Release(HSEM_ID_0,0);
/* wait until CPU2 wakes up from stop mode */
timeout = 0xFFFF;
while(( _HAL_RCC_GET_FLAG(RCC_FLAG_D2CKRDY) == RESET) && (timeout-- > 0));
if ( timeout < 0 )
{
    Error_Handler();
}

```

CM7

```

int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
    /* USER CODE BEGIN Boot_Mode_Sequence_1 */
    /*HW semaphore Clock enable*/
    __HAL_RCC_HSEM_CLK_ENABLE();
    /* Activate HSEM notification for Cortex-M4*/
    HAL_HSEM_ActivateNotification(__HAL_HSEM_SEMID_TO_MASK(HSEM_ID_0));
    /*
    Domain D2 goes to STOP mode (Cortex-M4 in deep-sleep) waiting for Cortex-M7 to
    perform system initialization (system clock config, external memory configuration..)
    */
    HAL_PWREx_ClearPendingEvent();
    HAL_PWREx_EnterSTOPMode(PWR_MAINREGULATOR_ON, PWR_STOPENTRY_WFE, PWR_D2_DOMAIN);
    /* Clear HSEM flag */
    HAL_HSEM_CLEAR_FLAG(__HAL_HSEM_SEMID_TO_MASK(HSEM_ID_0));
}

```

CM4

134

对于 CM7 和 CM4 系统初始化部分代码互换的实验，我们会发现，互换后代码会卡在如下的判断处，即使把 timeout 的这个判断去掉，RCC\_FLAG\_D1CKRDY 的时钟标志位也不会被为 RESET 清除的状态：

```

while ((HAL_HSEM_IsSemTaken(HSEM_ID_1) == 0) && (timeout-- > 0));
//while(( _HAL_RCC_GET_FLAG(RCC_FLAG_D1CKRDY) != RESET) && (timeout-- > 0)); //will never become true
if ( timeout < 0 )
{
    Error_Handler();
}

```

也就是 D1 域进入不了 DSTOP 模式：

这是什么原因呢？通过查看 [AN5215 STM32H747/757 advanced power management \(version 1\)](#) 我们知道如下关于域和 CPU 内核之间的关系：

Three power domains can be independently switched ON and OFF, depending on application needs:

- **D1 domain:** this domain contains the Cortex<sup>®</sup>-M7 core, the Flash memory and some peripherals. Thanks to the AXI bus matrix, this domain encompasses high bandwidth features and smart management.
- **D2 domain:** this domain contains the Cortex<sup>®</sup>-M4 core, most of the memories dedicated to I/O processing and most of the peripherals that are less bandwidth demanding.
- **D3 domain:** this domain contains the system control, low-power peripherals and memories designed to manage low-power modes. It is designed to be autonomous, and embeds a 64-Kbyte RAM, a basic DMA controller (BDMA), plus low-power peripherals to run basic functions while D1 and D2 domains can be switched off to save power.

以及关于域进入相应低功耗模式的条件：

<p>D1 and D2 subsystems</p>	<p>The domain bus matrix clock is stalled:</p> <ul style="list-style-type: none"> <li>• The domain CPU subsystem is in CStop mode,</li> </ul> <p>and</p> <ul style="list-style-type: none"> <li>• the other domain CPU subsystem has no peripheral allocated in the domain or the other domain CPU subsystem having an allocated peripheral in the domain is also in CStop mode,</li> </ul> <p>and</p> <ul style="list-style-type: none"> <li>• at least one PDDS_Dn bit for the domain selects DStop mode.</li> </ul>
-----------------------------	--

也就是说，如果要想 D1 域进入 DSTOP 模式，那么在 D1 域内的除了内核 CM7 之外，其 subsystem 分配的外设资源等也需要进入低功耗模式，而实例中，CM4 占用到的 Flash 在 D1 域，仍然处于活跃的状态，这也将会阻止 D1 域进入 DSTOP 模式，即 D1CKRDY 标注为会一直为 SET 的状态，也就是 while 循环的条件一直为真，与 timeout 设置的时长无关；

那如何改进呢？

### 2.3 方案二的完善

知道了 RCC\_FLAG\_D1CKRDY 不能被 RESET 的原因，我们可以用其他的判断条件替换：

具体代码实现如下，CM4 的代码：

```

69  /*
70  int main(void)
71  {
72  /* USER CODE BEGIN Boot_Mode_Sequence_0 */
73  int32_t timeout;
74  /* Wait until CPU2 boots and enters in stop mode or timeout*/
75  timeout = 0xFFFF;
76
77  while ((HAL_HSEM_IsSemTaken(HSEM_ID_1) == 0) && (timeout-- > 0));
78  if ( timeout < 0 )
79  {
80  Error_Handler();
81  }
82  /* MCU Configuration-----*/
83  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
84  HAL_Init();
85  SystemClock_Config(); /* Configure the system clock */
86  __HAL_RCC_HSEM_CLK_ENABLE(); /*HW semaphore Clock enable*/
87  /*Take HSEM */
88  HAL_HSEM_FastTake(HSEM_ID_0);
89  /*Release HSEM in order to notify the CPU2(CM4)*/
90  HAL_HSEM_Release(HSEM_ID_0,0);
91  /* wait until CPU2 wakes up from stop mode */
92  timeout = 0xFFFF;
93  while((__HAL_RCC_GET_FLAG(RCC_FLAG_D1CKRDY) == RESET) && (timeout-- > 0));
94  if ( timeout < 0 )
95  {
96  Error_Handler();
97  }
98
99  while (1)
100  {
101  }
102  /* USER CODE END 3 */
103  }
    
```

CM7 的代码：

CM4CM7\_Boot\_CM7

Files

- CM4CM7\_Boot - CM4CM7\_Bo...
- Application
  - EWARM
  - User
    - CM4
    - CM7
      - Core
        - main.c
        - stm32h7xx\_hal\_ms...
        - stm32h7xx\_it.c
- Drivers
  - CMSIS
    - system\_stm32h7xx\_dual...
    - STM32H7xx\_HAL\_Driver
  - Output
    - CM4CM7\_Boot\_CM7.map
    - CM4CM7\_Boot\_CM7.out

```

72 int main(void)
73 {
74     __HAL_RCC_HSEM_CLK_ENABLE();
75     /* Activate HSEM notification for Cortex-M7*/
76     HAL_HSEM_ActivateNotification(__HAL_HSEM_SEMID_TO_MASK(HSEM_ID_0));
77     /*
78     Domain D1 goes to STOP mode (Cortex-M7 in deep-sleep) waiting for Cortex-M4 to
79     perform system initialization (system clock config, external memory configuration.. )
80     */
81     __SEV();
82     HAL_PWREx_ClearPendingEvent();
83     HAL_HSEM_FastTake(HSEM_ID_1);
84     HAL_PWREx_EnterSTOPMode(PWR_MAINREGULATOR_ON, PWR_STOPENTRY_WFE, PWR_D1_DOMAIN);
85     /* Clear HSEM flag */
86     __HAL_HSEM_CLEAR_FLAG(__HAL_HSEM_SEMID_TO_MASK(HSEM_ID_0));
87     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
88     HAL_Init();
89     /* Initialize all configured peripherals */
90     MX_GPIO_Init();
91
92     while (1)
93     {
94         HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_1); // toggle yellow user LED
95         HAL_Delay(100);
96     }
97     /* USER CODE END 3 */
98 }

```

---

## 参考文献

文件编号	文件标题	版本号	发布日期
AN5215	STM32H747/757 advanced power management	1	9-May-2019

## 版本历史

日期	版本	变更
2021 年 10 月 29 日	1.0	首版发布

### 重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 [www.st.com/trademarks](http://www.st.com/trademarks)。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图 etc）信息，我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2020 STMicroelectronics - 保留所有权利