



# HME-M7

## 常见问题解答汇总

2022 年 04 月

京微齐力（北京）科技有限公司

## 注意

© 2018-2022 京微齐力（北京）科技有限公司版权所有

未经京微齐力（北京）科技有限公司书面许可，不得以任何形式或方式，如电子，机械，形式，包括影印、录音或其他数据储存和检索系统形式复制或转移此文档的任何部分，或将其翻译为其它任何语言或计算机语言。

所有商标均为京微齐力（北京）科技有限公司所有。

## 手册版本号

HME-M7ANC02-QA

## 联系我们

如果您在使用我们的产品过程中有任何疑问或问题，请与京微齐力（北京）科技有限公司或者您当地的代理商联系，或发送邮件至：

[sales@hercules-micro.com](mailto:sales@hercules-micro.com)

## 环境保护

本产品中包含的某些物质可能会对环境或人体健康有害，为避免将有害物质释放到环境中或危害人体健康，建议采用适当的方法回收本产品，以确保大部分材料可正确地重复使用或回收。有关处理或回收的信息，请与当地权威机构联系。

## 声明

本手册中包含的信息已经仔细检查并认为是完全可靠的。但是，不对手册中可能或潜在的错误负责。京微齐力（北京）科技有限公司保留停止发布或修改手册而不事先通知的权利。为确保获得最新的产品信息，建议用户及时更新手册版本。

本手册介绍的产品并没有被授权用作为生命保障设备或系统中的关键部件。在此使用到的术语有如下定义：1.生命保障设备或系统是满足以下条件的设备或系统，(a)被通过手术植入人体内或 (b)用来保障或维持生命，当按照标签上的使用说明正确使用时，有理由认为其工作的中断将会给使用者带来巨大的伤害。2.所谓关键部件是指生命保障设备或系统中满足以下条件的部件，即有理由认为该部件中断工作将会导致整个生命保障设备或系统中断工作，或者是影响到后者的安全性和有效性。

## 版本信息

下表列出了本产品手册的历史版本信息。

发布时间	文档版本	修订信息
2018年10月	1.0	初始版本
2022年04月18日	HME-M7ANC02-QA	更新版本

## 目录

注意 .....	2
版本信息 .....	3
目录 .....	4
关于本手册.....	5
1. M7 常见问题解答汇总 .....	6

## 关于本手册

本文档为 HME-M7 系列芯片在使用中的常见问题的解答汇总。

HME-M7 的相关信息与文档请访问：

[http://www.hercules-micro.com/content/details73\\_346.html](http://www.hercules-micro.com/content/details73_346.html)。

# M7 常见问题解答汇总

本章为 M7 系列芯片常见问题解答汇总。

## 1. M7 常见问题解答汇总

**Q1: M7 在 AS 模式下，下载线第一次可以下载配置文件到 M7 且配置成功，但后面 JTAG 检测不到 Device ID，也不能下载配置文件，这种情况怎么解决？**

### 原因

M7 例化 Arm IP 时，fp2soc\_rst\_n 信号是用作 FP 逻辑复位 ARM 核。硬件在内部把这个信号连到了 ARM 系统复位和 JTAG 模块的复位端口上。当 FP 通过 fp2soc\_rst\_n 复位 ARM 的时候，同时也就复位了 JTAG 模块。若 M7 导致 fp2soc\_rst\_n 一直是低电平状态，就会出现 JTAG 检测不到 Device ID，也不能下载配置文件的情况。

### 可能导致这种情况的场景

先检查设计上是否有导致 fp2soc\_rst\_n 始终处于复位状态的情况并改正。

1. 用 PLL 的 lock 输出作 fp2soc\_rst\_n 的复位源。若 PLL 的时钟输入没有或其他原因导致 PLL 不能锁定，就会导致 fp2soc\_rst\_n 是复位状态，从而导致出现 JTAG 检测不到 Device ID，也不能下载配置文件的情况。

2. 外部 IO 信号作为 fp2soc\_rst\_n 的复位源。这种情况下，若是用户忘记分配这个 IO 复位信号，工具会优化掉相关逻辑，导致配置成功后，fp2soc\_rst\_n 信号是复位状态；或者用户分配了 IO，但连接这个 IO 的外部驱动是 0，也会使 fp2soc\_rst\_n 是复位状态。

### 解决办法

1. 断电，M7 的 CFG\_MODE 切换到高变成 PS 模式。再上电，用 JTAG download 正确的 bitstream 数据到 SPI Flash 中。断电后，切换 M7 的 CFG\_MODE 为低变成 AS 模式。

配置如图 1 所示，设计 CFG\_MODE，AS 模式和 PS 模式跳线切换。

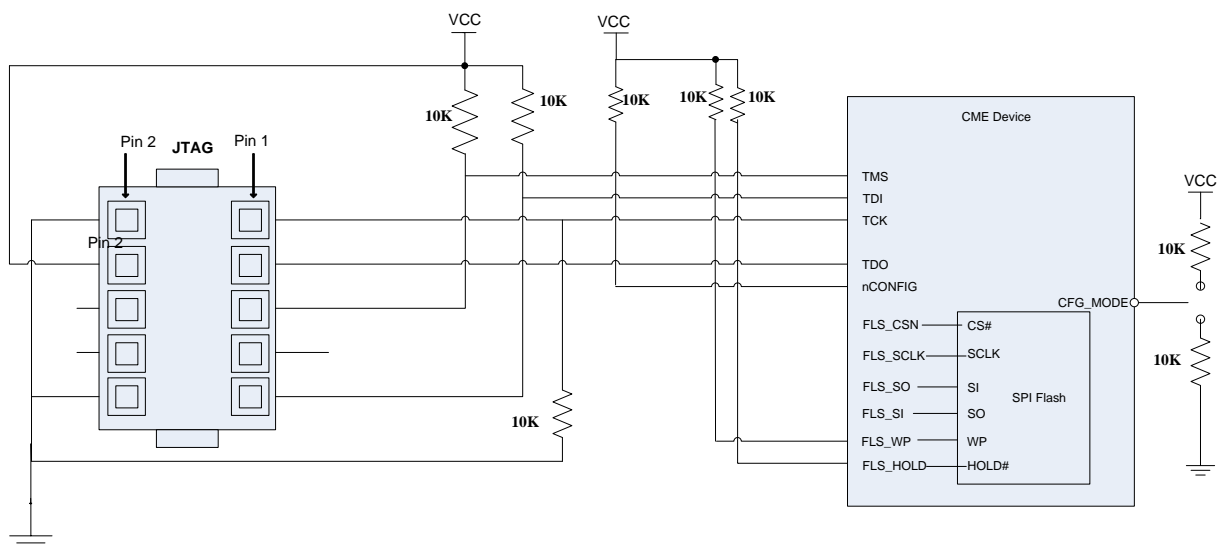


图 1 模式切换配置示意图

**特别注意：M7 144 封装的内嵌了 Flash，固定为 AS 模式，外部无法改成 PS 模式。**

**Q2： ARM 在线调试的时候，程序执行到 SystemInit 函数中的 GLB\_SelectSysClkSource (SYS\_CLK\_SEL\_DLL)处停住了。**

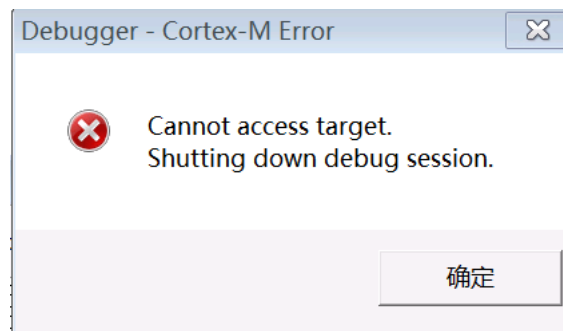
### 原因

ARM 程序选择的系统时钟源与 Fuxi 设计中的系统时钟源不一致。

### 解决办法

系统时钟可以来自指定的 DLL2、外部 Xin，内部 OSC 和其他时钟源，要确保 SystemInit 函数中的 GLB\_SelectSysClkSource 选择与 Fuxi 中一致，当例化 DDR 时选择 DLL；当用 PLL 输出或其他全局时钟时选择 external。

**Q3： ARM 在线调试的时候，下载过程中弹出如图 2 所示窗口，怎么解决？**



**图 2 提示窗口**

这很可能是 Keil 工程中的 Memory 窗口打开了，且打开的区域不能访问，比如访问的 0XA0000000 地址，但是 FP 里没有扩展这个地址，导致 Keil 读这个地址出现异常，锁住了系统总线。

在 Keil 的\*.uvopt 文件中有如下语句：

```
<MemoryWindow1>
  <Mm>
  <WinNumber>1</WinNumber>
  <SubType>0</SubType>
  <ItemText>0xa0000000</ItemText>
</Mm>
</MemoryWindow1>
```

关闭 Keil 工程，删掉这些语句并保存。再打开工程，就能正常在线调试。

**Q4： 没用到的 bank，bank 电源管脚是否能接地？**

Float 或接正常 IO 电压，不能接地。

**Q5： LQFP144、LQFP216 和 LQFP256 封装的 M7 芯片连接 DDR2/3 颗粒，两层板是否能 Layout？**

能。

**Q6: IO42\_NMI\_2 这个脚悬空或接高会有什么结果?**

会导致 ARM 程序一直进入 NMI 中断复位程序。不用作 ARM NMI 中断源，需把该 Pin 下拉。

在 Fuxi 软件的 IP wizard 中选用 ARM Cortex-M3 时，在选择 ARM Cortex-M3 的 features 时，不要选择 ARM NMI pin 功能。这样该 NMI 对应的 M7 的 pin 就不会作为 ARM Cortex-M3 的 NMI 中断源。

**Q7: I/O DDR 模式怎么用?**

LVDS 可以用作 input DDR 模式；DDR 可以用作 output DDR 模式。Inout 需要分别连到 LVDS 和 DDR I/O pin 上实现 DDR 模式双向 I/O。

**Q8: M7 外接 DDR 应当注意的问题有哪些?**

## 1. 电路设计

a) DQ、DQS、DM 信号线上必须串电阻  $22\Omega$ ；

b) DDR3/2 差分时钟线上靠近 M7 芯片端串  $22\Omega$  (M7 的 DDR3/2 时钟 PAD 到  $22\Omega$  电阻的走线距离尽可能短)，差分时钟端端接  $150\Omega$  (电阻位置要尽可能靠近 DDR3/2 颗粒的时钟 PAD)；

c) M7 DDR calibration 的电阻：

➤ DDR2 颗粒接  $300\Omega$ ；

➤ DDR3 颗粒接  $240\Omega$ ；

d) 地址线等控制线不串电阻简化 PCB 布线；

## 2. FPGA 设计

DQ、DQS、DM

**Q9: ADC 的 VREF 可以悬空吗?**

ADC 的 VREF 悬空表示 ADC 用内部的 reference，但需接电容滤波。

**Q10: M7 上的接口如 RTC\_XIN, XSCI, XSCO 等没有用到的输入信号悬空对设计有没有影响?**

无影响。

**Q11: XIN,XOUT,RTC\_XIN,RTC\_XOUT 是否都处于 BANK11? BANK11 作为 LVDS 配置的时候供电 2.5V, 会对这些信号有什么影响?**

对这些信号没有影响。

**Q12: 关于 USB: 必须使用 XSCI, XSCO 为 USB 提供时钟吗? 用 M7 的全局时钟网络是否可行?**

USB 的时钟既可以使用外部的晶体通过 XSCI, XSCO 提供的时钟，也可以使用 M7 的全局时钟，需要在 Fuxi 工程中给 USB 提供时钟，同时对应 USB 的时钟选择寄存器要配置相应的值。用 M7 的 PLL 产生的 12M 时钟作为 USB 控制器的时钟需要注意 PLL 的输出时钟的精确度。但 PLL 产生多个时钟时，有可能 12M 的时钟不准，这会导致 USB 传输错误。

**Q13: 关于以太网部: 参考设计中为 M88E1111 提供 GTX\_CLK 时钟的有多个, 是否可以由 IO\_06P\_ETH\_PHY\_GTX\_CLK\_O\_4 提供?**



可以用 IO\_06P\_ETH\_PHY\_GTX\_CLK\_O\_4。

**Q14: DDR 的 PLL 是否需要指定使用某个特定 CMB?**

ARM wizard 在指定 DDR 的时候, 已经把 pll 和 DLL 固定到特定的位置上, 不能手动修改位置。

**Q15: 手册上说 M7 内嵌 FLASH 的时候只能用使用 AS 和 JTAG 配置模式, 没有 PS 模式的情况下怎么用外部控制器配置 M7? 如果内部的 FLASH 用于 AS 配置, 是否意味着 FLASH 不能同时用于 ARM? AS 配置 M7 需要占用内部 FLASH 多少容量?**

没有 PS 模式的情况下, JTAG 可以操作 SPI Flash, ARM 也可以操作 SPI Flash。AS 的 M7 只用 128K 的 SRAM 做 code 的话, 大约需要占用 4Mb 的 Flash 空间, M7 部分封装上面带的 Flash 是 16Mb 的, 除去芯片配置占用的空间, 其它空间用户均可以使用。但 ddr 中跑 code 的话, 那就不够。若是不够, 在设计的时候用外部的 EVM 存 ARM 的 code。

**Q16: M7 的 SPI 接口是可以配置成 Master or Slave mode?**

M7 的 SPI 只支持 Master 模式, 不支持 slave 模式。

**Q17: 有没有 M7 上 SPI Flash 读写控制和 SPI 接口控制的相关代码?**

在 M7 的 FLASH demo 程序中有 SPI Flash 读写控制程序, SPI 中有 SPI 接口的相关代码。这些代码会放到 Fuxi example 目录下。

**Q18: 配置输出文件 bin, PS, JTAG, SPI, Hex 各种格式有什么区别?**

Bin 文件是二进制格式的 CME 配置文件, 包含了注释和纯配置数据; PS 是 PS 模式下的 16 进制的以 Byte 为单位的 CME 配置文件, 包含了注释和纯配置数据; SPI 是 AS 模式下的 16 进制的以 Byte 为单位的 CME 配置文件, 包含了注释和纯配置数据; 文件是 Hex 格式的 CME 配置文件, 为纯配置数据。

**Q19: 在 SPI 配置 FLASH 中 FPGA 的配置数据和 ARM 的程序段和数据段是如何存放的? 如果想用配置 FPGA 的 SPI FLASH 的话应该如何用? 怎么保证在后续操作 SPI Flash 过程中不会对 FPGA 的配置数据造成影响? 配置文件最大有多大? 哪个地址开始可以作为存储空间?**

用 Cable 下载的时候, GUI 上的 Console 输出会显示 M7 的 Image 占用的大小。用户可以根据这些信息来规划自己的存储空间。

**Q20: SPI controller 和 SPI interface 等 IP 核能在芯片发布的时候一块发布吗? Fuxi 软件里是灰色的, 无法使用。**

M7 中的 SPI Flash 用法与 M5 不同, 无 SPI interface 的接口。用户只需通过 ARM 的程序控制就行。实例代码见 Fuxi 的 example 中的 FLASH demo。

**Q21: ADC 的外部基准是 1.0V, 从客户端反馈来看, 1.0V 的基准不好买, 能买到的也是 1.024V 的电压基准芯片 (售价在 2-3\$左右), 价格较贵, 是否我们的 ADC 可以支持 1.25V 的基准, 或者是否有低成本的 1.0V 基准芯片推荐?**

可以用 1.024V 的电压基准芯片, 不支持 1.25V 的电压基准。

**Q22: 从目前的测试来看 (请参考 CME\_M7\_DDR\_Training 培训文档), FPGA 在 160MHz 下访问 DDR3 的实际性能是 280MB/s (写), 180MB/s (读)。M7 Datasheet 中 DDR 的性能标称 666Mbps\*16bit (即: 1,332MB/s), 就算是除去命令的开销, 性能也应该要到达 1GB/s 以上才正常。如何解决这个问题?**

FPGA AHB Master 的时序最大约为 160MHz，因此最大带宽为 480MB/s。DDR 控制器最大 burst 是 8，每次 burst 完成后，DDR controller 要重发相关 DDR 操作时序；FPGA AHB Master 与系统的 AHB 总线是通过异步桥通讯，每次重新交互需要消耗不少时钟周期。上述原因导致 FPGA 在 160MHz 下访问 DDR3 的实际性能是 280MB/s（写），180MB/s（读）。

### Q23: ARM Cortex-M3 是否支持 SDRAM 作为 Code Memory 以及性能如何？

支持。

### Q24: M7 以太网是否支持 RMII 和 RGMII 接口？如果支持，如何支持？

不支持。

### Q25: M7 是否支持 4 线 SPI Flash 作为配置芯片？如果支持，如何支持？

不支持。

### Q26: M7 中断嵌套怎么处理？

Cortex-M3 支持中断多级嵌套。M7 的中断嵌套需要如下处理：中断服务程序中在操作外设寄存器时需要先关全局中断，操作完后再开中断。

### Q27: M7 串口有时候收不到数，串口程序无反应？

当串口打开了接收中断并收到数后，内部会自动启动一个定时器（最长 255 个 clock，而且这个定时器无法关闭）计时，如果当这个定时器计数到后，仍然没有将串口的数据读走，整个串口就进入了 timeout 状态。在这种状态下，即使再收到数据，也不再会触发接收中断。这就是用户反馈的串口接收经常死掉。

## 解决办法

解决串口死掉问题，可以在打开接收中断的时候，同时打开 UART\_Int\_TxTimeoutNotEmpty 中断位，当没有及时的读到接收的数据而导致进入 timeout 状态后，UART\_Int\_TxTimeoutNotEmpty 中断位也会被置位，在 TxTimeoutNotEmpty 中断处理分支中，将 fifo 中的数据读空，再清除 TxTimeoutNotEmpty 中断位。但这种解决方法只能解决串口无法进入接收中断的表现，依然解决不了客户的整体设计不合理的问题，也就是说客户的串口未来可能还是会因为 overflow 而丢数据。示例代码如下：

init:

```
UART_StructInit(&init);  
UART_Init(UART2, &init);  
UART_Cmd(UART2, TRUE);  
UART_ITConfig(UART2, UART_Int_RxNotEmpty, TRUE);  
  
UART_ITConfig(UART2, UART_Int_TxTimeoutNotEmpty, TRUE);
```

interrupt:

```
void UART2_IRQHandler(void)  
{
```

```
unsigned char data;
```

```
uint8_t databuf[16];
```

```
uint8_t count, size;
```

```
if( UART_GetITStatus(UART2, UART_Int_RxNotEmpty))
```

```
{
```

```
    UART_ReceiveData(UART2, 1, &data);
```

```
    while( 1 != UART_SendData(UART2, 1, &data));
```

```
    //communication( data );
```

```
    UART_ClearITPendingBit(UART2, UART_Int_RxNotEmpty);
```

```
}
```

```
if( UART_GetITStatus(UART2, UART_Int_TxTimeoutNotEmpty))
```

```
{
```

```
    count = UART_ReceiveData(UART2, 16, databuf);
```

```
    UART_ClearITPendingBit(UART2, UART_Int_TxTimeoutNotEmpty);
```

```
}
```

```
}
```

### **Q28: M7 USB 用户使用有什么注意的地方?**

用 M7 的 PLL 产生的 12M 时钟作为 USB 控制器的时钟需要注意 PLL 的输出时钟的精确度。但 PLL 产生多个时钟时，有可能 12M 的时钟不准，这会导致 USB 传输错误。

### **Q29: M7 各项性能怎样?**

#### 1. Cortex-M3

各 CPU 数据性能如表 1 所示。

测试采用数据搬移的方式，即一读一写。如果是连写，会比这个结果高。如果是连读，会比这个结果低。这里的 performance 是指 bus 实际传输的数据。例如读 1 个 byte，写一个 byte，这个算成 2byte。测试时 PBUS 跑 200Mhz, ARM200Mhz, FP 跑 100Mhz。总线理论最高带宽是 800MByte/s。如果将时钟频率提高，

效率会等比例提高。可以看到：

- a) 效率最高的是 SOC DMA 从 SRAM2SRAM 的搬移，超过 600Mbyte；
- b) 操作 DDR，由于 DDR 本身的开销，效率会降低，不到 300Mbyte；
- c) 涉及到 FP 的操作，由于 FP 本身的频率较低，加上异步桥的转换，效率会进一步降低，分别是 235MByte（SRAM）和 159MByte（DDR）；
- d) CPU 的操作都是 single 操作，效率很低；如果 CPU 采用 async 方式，就更低了；最高 115Mbyte（sync、SRAM），其余在 50MByte 左右。

**表 1 各 CPU 数据性能**

Bus Master	条件	Master	搬移路径	搬移数据量 (读写各半)	时间 (纯数据时间)	数据速率 MB/S
<b>SOC DMAC (burst16)</b>	PBUS clock	200M	SRAM2SRAM	8192*2 B	0.0263MS	622
	PBUS clock	200M	DDR2DDR	8192*2 B	0.0580MS	282
	FP clock	100M	FP2FP	8192*2 B	0.0662MS	247
<b>FP master (burst16)</b>	PBUS clock	200M	SRAM2SRAM	8192*2 B	0.0698MS	235
	FP clock	100M	DDR2DDR	8192*2 B	0.0103MS	159
<b>ARM SYNC (single)</b>	PBUS clock	200M	SRAM2SRAM	256*4*2	0.01778MS	115
	FP clock	200M	DDR2DDR	256*4*2	0.0465MS	44
<b>ARM ASYNC (single)</b>	PBUS clock	200M	SRAM2SRAM	256*4*2	0.04208MS	48
	FP clock	200M	DDR2DDR	256*4*2	0.0641MS	32
<b>USB</b>	PBUS clock	200M	SRAM2SRAM	256*4*2	0.04208MS	48
	FP clock	100M	DDR2DDR	256*4*2	0.0641MS	32
<b>1000M Ethernet</b>	PBUS clock	200M	SRAM2SRAM	256*4*2	0.04208MS	48
	FP clock	100M	DDR2DDR	256*4*2	0.0641MS	32

## 2. LVDS

LVDS 的有关性能如表 2 所示。

**表 2 LVDS 有关性能**

项目	描述	频率
7: 1 Transmitter	IO hardware serdes	800Mbps
7: 1 Receiver	IO hardware serdes	800Mbps
SDR TX	FP logic	800Mbps
SDR RX	FP logic	400Mbps