

芯片的识别

本勘误表适用于上海先楫半导体科技有限公司的 HPM6700/6400 系列微控制器产品，芯片版本 1.0 和 2.0。

通过芯片封装表面的产品型号信息可以识别本勘误表适用的芯片，请注意芯片型号中版本位为：1 或者 2

例：HPM6750IVM2

本勘误表适用的产品型号有：

HPM6750IVM^{*}、HPM6750IAN^{*}、HPM6730IVM^{*}、HPM6730IAN^{*}、
HPM6450IVM^{*}、HPM6450IAN^{*}、HPM6430IVM^{*}、HPM6430IAN^{*}、
HPM6754IVM^{*}、HPM6754IAN^{*}、HPM6454IVM^{*}、HPM6454IAN^{*}。

HPM64G0CVM2、HPM64G0CAN2

*x=1：版本 1，x=2：版本 2

目录

1	HPM6700/6400 系列微控制器使用限制汇总	1
2	HPM6700/6400 系列微控制器使用限制描述	2
2.1	E00001: RISC-V 处理器指令和数据本地存储器使用限制	2
2.1.1	问题描述	2
2.1.2	规避方法	2
2.1.3	修正情况	2
2.2	E00002: 随机数发生器 RNG 使用限制	3
2.2.1	问题描述	3
2.2.2	规避方法	3
2.2.3	修正情况	3
2.3	E00003: 侵入检测模块 TAMP 使用限制	4
2.3.1	问题描述	4
2.3.2	规避方法	4
2.3.3	修正情况	4
2.4	E00004: 模数转换器 ADC12、ADC16 的 CONT_EN 控制位使用限制	5
2.4.1	问题描述	5
2.4.2	规避方法	5
2.4.3	修正情况	5
2.5	E00005: XDMA 读写 DRAM 的使用限制	6
2.5.1	问题描述	6
2.5.2	规避方法	6
2.5.3	修正情况	6
2.6	E00006: PDMA 和 JPEG 的 RGB2YUV 使用限制	7
2.6.1	问题描述	7
2.6.2	规避方法	7
2.6.3	修正情况	7
2.7	E00007: PDMA 的 PS[0][ORG] 和 PS[1][ORG] 寄存器使用限制	8
2.7.1	问题描述	8
2.7.2	规避方法	8
2.7.3	修正情况	8
2.8	E00008: PDMA 的 YUV2RGB_COEF0 寄存器使用限制	9
2.8.1	问题描述	9
2.8.2	规避方法	9
2.8.3	修正情况	9
2.9	E00009: ADC 的抢占转换模式使用限制	10
2.9.1	问题描述	10
2.9.2	规避方法	10
2.9.3	修正情况	10
2.10	E00010: PDMA 和 LCD 的 Alpha Blending 功能使用限制	11
2.10.1	问题描述	11
2.10.2	规避方法	11

2.10.3 修正情况	11
2.11 E00011: PIOC 寄存器读限制	12
2.11.1 问题描述	12
2.11.2 规避方法	12
2.11.3 修正情况	12
2.12 E00012: RISC-V CPU ILM/DLM 访问限制	13
2.12.1 问题描述	13
2.12.2 规避方法	13
2.12.3 修正情况	13
2.13 E00013: SDP 的 SDPCR 寄存器 INTEN 控制位使用限制	14
2.13.1 问题描述	14
2.13.2 规避方法	14
2.13.3 修正情况	14
2.14 E00016: CAN FD 模式 TDC 使用限制	15
2.14.1 问题描述	15
2.14.2 规避方法	15
2.14.3 修正情况	15
2.14.4 附录信息	15
2.15 E00018: UART DMA 请求使用限制	16
2.15.1 问题描述	16
2.15.2 规避方法	16
2.15.3 修正情况	16
2.16 E00019: PDMA 的图像缩放功能使用限制	17
2.16.1 问题描述	17
2.16.2 规避方法	17
2.16.3 修正情况	17
3 版本信息	18

1 HPM6700/6400 系列微控制器使用限制汇总

本产品的使用限制汇总如表 1:

编号	使用限制描述	影响版本
E00001	RISC-V 处理器指令和数据本地存储器使用限制	1.0
E00002	随机数发生器 RNG 使用限制	1.0
E00003	侵入检测模块 TAMP 使用限制	1.0 和 2.0
E00004	模数转换器 ADC12、ADC16 的 CONT_EN 控制位使用限制	1.0
E00005	XDMA 读写 DRAM 的使用限制	1.0 和 2.0
E00006	PDMA 和 JPEG 的 RGB2YUV 使用限制	1.0 和 2.0
E00007	PDMA 的 PS[0][ORG] 和 PS[1][ORG] 寄存器使用限制	1.0 和 2.0
E00008	PDMA 的 YUV2RGB_COEF0 寄存器使用限制	1.0 和 2.0
E00009	ADC 的抢占转换模式使用限制	1.0 和 2.0
E00010	PDMA 和 LCDC 的 Alpha Blending 功能使用限制	1.0 和 2.0
E00011	PIOC 寄存器读限制	1.0 和 2.0
E00012	RISC-V CPU ILM/DLM 访问限制	1.0
E00013	SDP 的 SDPCR 寄存器 INTEN 控制位使用限制	1.0 和 2.0
E00016	CAN FD 模式 TDC 功能使用限制	1.0 和 2.0
E00018	UART DMA 请求使用限制	1.0 和 2.0
E00019	PDMA 的图像缩放功能使用限制	1.0 和 2.0

表 1: HPM6700/6400 系列微控制器使用限制总结

2 HPM6700/6400 系列微控制器使用限制描述

2.1 E00001: RISC-V 处理器指令和数据本地存储器使用限制

2.1.1 问题描述

RISC-V CPU 读取自身的指令和数据本地存储器 (ILM 和 DLM) 时, 有一定几率, 会读取到全 0 数据。其影响为:

1. RISC-V CPU 从 ILM 执行代码, 有较低概率错误地取指到全 0 指令, 并触发非法指令异常
2. RISC-V CPU 读写 ILM 和 DLM 时, 其他总线主设备, 如 DMA 控制器等, 同时通过 RISC-V 处理器的从接口 (CPUx_xLM_SLV) 访问该存储器, 有较低概率, 该主设备读取到错误的全 0 数据。

2.1.2 规避方法

针对从指令本地存储器执行代码取到全 0 指令, 可以选择以下规避方法之一:

1. RISC-V CPU 可以从指令本地存储器 (ILM) 中执行代码, 当取指错误, 进入非法指令异常处理程序时, 无需特殊处理此异常, 返回后 RISC-V CPU 再次取指结果正确。因此, 从指令本地存储器 (ILM) 中执行代码会有一定的性能损失, 但是不影响程序正确执行。以 Coremark 为例, 性能损失约为 5%。
2. RISC-V CPU 可以从其他存储器中执行代码, 如从片上通用 SRAM (AXI SRAMx), 或者通过 XPI 接口从外部串行 NOR Flash 执行代码。此时, 应打开 RISC-V CPU 的 L1 缓存, 以保证代码执行性能。

针对其他总线主设备通过 RISC-V CPU 的从接口 (CPUx_xLM_SLV) 访问指令和本地存储器 (ILM/DLM) 读取到全 0 数据, 可以选择以下规避方法之一:

1. 避免其他总线主设备访问 RISC-V CPU 的指令和数据本地存储器 (ILM 和 DLM)
2. 当 RISC-V CPU 访问指令和数据本地存储器 (ILM 和 DLM) 其中之一时, 其他总线主设备不要访问该本地存储器。注意。当 CPU 访问其中一个本地存储器, 比如 ILM 时, 允许其他主设备访问另一个本地存储器, 即 DLM。

2.1.3 修正情况

HPM6700/6400 系列微控制器版本 2.0 已修正。

2.2 E00002: 随机数发生器 RNG 使用限制

2.2.1 问题描述

随机数发生器 RNG 无法产生随机数。

2.2.2 规避方法

使用软件的随机数生成算法，生成随机数。

2.2.3 修正情况

HPM6700/6400 系列微控制器版本 2.0 已修正。

2.3 E00003: 侵入检测模块 TAMP 使用限制

2.3.1 问题描述

侵入检测模块 TAMP 的 TAMP4 和 TAMP5 寄存器组，寄存器组中的 CONTROL、POLY，LFSR 寄存器无法读出，读取值总为 0。

TAMP4 和 TAMP5 这 2 组寄存器写入功能正常，用户可以配置这些寄存器，使用相应的侵入检测功能。

2.3.2 规避方法

忽略这 2 组寄存器读取的值。

2.3.3 修正情况

无。

2.4 E00004: 模数转换器 ADC12、ADC16 的 CONT_EN 控制位使用限制

2.4.1 问题描述

ADC12 或者 ADC16 的进行序列转换模式时，如果 CONT_EN 位置 1，ADC 在收到触发信号后，从第二次序列转换起，只会完成序列中最后一个通道转换。

2.4.2 规避方法

可以选择以下规避方法之一：

1. 使用抢占转换模式，抢占转换模式每个序列支持最多 4 个连续转换。
2. 使用序列转换模式时，将 CONT_EN 位和 RESTART_EN 位都置 1，这样，ADC12/ADC16 会不停地连续循环转换整个序列。

2.4.3 修正情况

HPM6700/6400 系列微控制器版本 2.0 已修正。

2.5 E00005: XDMA 读写 DRAM 的使用限制

2.5.1 问题描述

当使用 XDMA 读或写 DRAM 时，如果设置的数据传输位宽小于 64 位，会发生数据丢失。

2.5.2 规避方法

如果 XDMA 的源数据来自 DRAM，则须将 SRCWIDTH 寄存器位设置为 64 位传输。如果 XDMA 的目标数据存放在 DRAM，则须将 DSTWIDTH 寄存器位设置为 64 位传输。

2.5.3 修正情况

无。

2.6 E00006: PDMA 和 JPEG 的 RGB2YUV 使用限制

2.6.1 问题描述

PDMA 和 JPEG 的图像格式转换功能在将 RGB 格式图像转换成 YUV 格式图像时,即 PDMA 的 RGB2YUV 寄存器 YCBCR_MODE 位置 0,或者 JPEG 的 RGB2YUV 寄存器 YCBCR_MODE 位置 0 时,转换结果不正确。

2.6.2 规避方法

建议用户使用 PDMA 和 JPEG 图像格式转换功能将 RGB 格式图像转换为 YCbCr 格式图像,而不是 YUV 格式图像。建议把 PDMA 的 RGB2YUV 寄存器 YCBCR_MODE 位置 1,或者 JPEG 的 RGB2YUV 寄存器 YCBCR_MODE 位置 1。

2.6.3 修正情况

无。

2.7 E00007: PDMA 的 PS[0][ORG] 和 PS[1][ORG] 寄存器使用限制

2.7.1 问题描述

PDMA 的 PS[0][ORG] 和 PS[1][ORG] 寄存器里高十六位和低十六位写时顺序正确如本产品用户手册所示, 读时高十六位和低十六位次序交换了。

2.7.2 规避方法

用户如果通过软件读出 PS[0][ORG] 和 PS[1][ORG] 寄存器, 并将读取值与 golden 数据比对, 需将高十六位和低十六位次序交换后再比对。

2.7.3 修正情况

HPM6700/6400 系列微控制器版本 2.0 已修正。

2.8 E00008: PDMA 的 YUV2RGB_COEF0 寄存器使用限制

2.8.1 问题描述

RGB2YUV_COEF0 里第 30 位是 ENABLE 位，但该 bit 读出值和写入值正好相反。

2.8.2 规避方法

用户如果通过软件读出该位和 golden 数据比对，需将该位取反后再比对。

2.8.3 修正情况

无。

2.9 E00009: ADC 的抢占转换模式使用限制

2.9.1 问题描述

本产品支持 4 个电机控制子系统，每个子系统内部通过互联管理器 TRGM 输出 3 个抢占转换触发信号，共 12 个信号同时连接到每个 ADC 上，也就是每个 ADC 有 12 个抢占转换触发输入，对应每个 ADC 也有 12 个抢占转换队列。对于单个 ADC，每个信号会触发一次抢占队列转换（1 到 4 次 ADC 转换）。

由于没有抢占触发信号使能位，当某个抢占转换信号触发后，每个打开的 ADC 都会进行至少一次 ADC 转换（如果没有配置相应队列，会对 CHAN0 进行一次转换）。

多个电机协同工作时，如果各电机生产的 ADC 抢占转换触发信号周期不同，可能会导致两个以上的电机，在同一时间段产生触发信号，前一个触发信号已经让 ADC 在工作（4 个 ADC 都会工作），后一个触发信号就会被 ADC 丢失，同时产生一个错误中断。

2.9.2 规避方法

建议用户采用以下方法之一：

1. 多个电机协同工作时，所有电机触发 ADC 抢占转换的周期相同，并且错开各自生成 ADC 抢占转换的时机
2. 多个电机协同工作时，每个电机系统只触发 1 个 ADC，并使用序列转换模式

2.9.3 修正情况

无。

2.10 E00010: PDMA 和 LCDC 的 Alpha Blending 功能使用限制

2.10.1 问题描述

1. PDMA 和 LCDC 的 alpha blend 不支持 alpha，仅支持 blend(即 alpha=0x0 或者 0xFF 的情况)。
2. PDMA 和 LCDC 的 alpha blend 的 SKBlendMode_Modulate 模式不能用。

2.10.2 规避方法

建议用户采用以下方法之一：

1. 仅使用 alpha=0x0 或者 0xFF。
2. 如果 alpha 不等于 0x0，也不等于 0xFF，建议使用软件计算。
3. 不使用 PDMA 和 LCDC 的 alpha blend 的 SKBlendMode_Modulate 模式。

2.10.3 修正情况

无。

2.11 E00011: PIOC 寄存器读限制

2.11.1 问题描述

1. 写入 PIOC 寄存器时，结果正常。
2. 读取 PIOC 寄存器时，返回的寄存器值不正确。

2.11.2 规避方法

建议用户采用以下方法：

1. 写入 PIOC 寄存器时，应使用内存映射表列出的寄存器地址。
2. 读取 PIOC 寄存器时，应在原地址基础上减去 0x700 的偏移。

2.11.3 修正情况

无。

2.12 E00012: RISC-V CPU ILM/DLM 访问限制

2.12.1 问题描述

1. 总线上其他主设备访问过 CPU0 或者 CPU1 的本地存储器从接口 (CPU0_ILM_SLV, CPU0_DLM_SLV, CPU1_ILM_SLV, CPU1_DLM_SLV)。
2. CPU0 或者 CPU1 子系统经历过复位, 或者掉电之后重新上电。
3. 此时总线上其他主设备再访问改本地存储器从接口会导致异常。

2.12.2 规避方法

建议用户采用以下方法之一:

1. 不对 CPU0 或者 CPU1 子系统单独复位, 或者单独掉电。建议用户对 CPU0 或者 CPU1 子系统进行时钟门控来降低功耗。
2. 如果限制其他总线主设备, 使它们不访问 CPU0 或者 CPU1 的本地存储器从接口, 那么 CPU0 或者 CPU1 子系统仍然允许单独复位和单独掉电。

2.12.3 修正情况

HPM6700/6400 系列微控制器版本 2.0 已修正。

2.13 E00013: SDP 的 SDPCR 寄存器 INTEN 控制位使用限制

2.13.1 问题描述

1. SDP 的 SDPCR 寄存器 INTEN 控制位置 0 时，SDP 仍能生成中断。

2.13.2 规避方法

建议用户采用以下方法：

1. 用户可以使用 SDP 描述符中 PKTCTL 控制字的位 1（即 PKTCTL[1]: PKTINT）来控制中断的使能或者关闭。

2.13.3 修正情况

无。

2.14 E00016: CAN FD 模式 TDC 使用限制

2.14.1 问题描述

发送 CAN FD 帧，并使能了 TDC(发送延时补偿) 功能时，一些特定帧的 ID 使得在 FDF 位（也称 EDL 位）之前插入充填位（stuff bit）的情况下，发送延时补偿功能异常，CAN FD 帧发送异常。

2.14.2 规避方法

建议用户采用以下方法之一：

- 发送 CAN FD 帧时，不使用 TDC(发送延时补偿) 功能，此种情况下发送比特率不超过 2.5Mbps。
- 发送超过 2.5Mbps 的 CAN FD 帧时，避免使用会引发在 FDF 位（也称 EDL 位）之前插入充填位（stuff bit）的 CAN ID

2.14.3 修正情况

无。

2.14.4 附录信息

受影响的 CAN ID 如下: 0x8, 0x18, 0x28, 0x38, 0x48, 0x58, 0x68, 0x88, 0x98, 0xa8, 0xb8, 0xc8, 0xd8, 0xe8, 0x100, 0x108, 0x118, 0x128, 0x138, 0x148, 0x158, 0x168, 0x178, 0x17c, 0x188, 0x198, 0x1a8, 0x1b8, 0x1c8, 0x1d8, 0x1e8, 0x1f8, 0x208, 0x218, 0x228, 0x238, 0x248, 0x258, 0x268, 0x278, 0x27c, 0x288, 0x298, 0x2a8, 0x2b8, 0x2c8, 0x2d8, 0x2e8, 0x300, 0x308, 0x318, 0x328, 0x338, 0x348, 0x358, 0x368, 0x378, 0x37c, 0x388, 0x398, 0x3a8, 0x3b8, 0x3c8, 0x3d8, 0x3e8, 0x3f8, 0x408, 0x418, 0x428, 0x438, 0x448, 0x458, 0x468, 0x478, 0x47c, 0x488, 0x498, 0x4a8, 0x4b8, 0x4c8, 0x4d8, 0x4e8, 0x500, 0x508, 0x518, 0x528, 0x538, 0x548, 0x558, 0x568, 0x578, 0x57c, 0x588, 0x598, 0x5a8, 0x5b8, 0x5c8, 0x5d8, 0x5e8, 0x5f8, 0x608, 0x618, 0x628, 0x638, 0x648, 0x658, 0x668, 0x678, 0x67c, 0x688, 0x698, 0x6a8, 0x6b8, 0x6c8, 0x6d8, 0x6e8, 0x700, 0x708, 0x718, 0x728, 0x738, 0x748, 0x758, 0x768, 0x778, 0x77c, 0x788, 0x798, 0x7a8, 0x7b8, 0x7c8, 0x7d8, 0x7e8, 0x7f8

2.15 E00018: UART DMA 请求使用限制

2.15.1 问题描述

UART 的 DMA 功能有以下使用限制:

1. UART 的 FCR 寄存器的 DMAE 控制位失效。当 DMAE 位清 0 时, UART Rx FIFO 非空时, 仍会生成 UART Rx DMA 请求; UART Tx FIFO 为空时, 仍会生成 UART Tx DMA 请求。
2. UART 的 FCR 寄存器 RFIFOT 用于配置 RX FIFO 的触发阈值, 当 RFIFOT 为非零时 (即配置触发阈值为 RX FIFO 充电深度达到 1/4, 1/2, 3/4 RX FIFO 容量时), 只要 RX FIFO 非空, 仍可能生成 UART RX DMA 请求。
3. UART 的 FCR 寄存器 TFIFORST 和 RFIFORST 可用于复位 UART Tx FIFO 和 Rx FIFO, 清空 Tx FIFO 和 Rx FIFO 内已有的数据, 但不能清除已产生的 DMA 请求。DMA 如果尚未响应这个 DMA 请求 (通常此情况发生在 DMA 完成了完整的 UART 数据收发任务后, 尚未重新配置时), 重新配置或者打开后, 有可能响应这个已生成 UART DMA 传输请求, 接收到或者发送错误数据。

2.15.2 规避方法

对于使用限制 1 和 2, 可采用以下方法规避:

- 不使用 DMA 收发 UART 数据时, 避免通过配置 DMAMUX, 将 UART Tx 和 Rx 的 DMA 请求连接到任一 DMA 通道。
- 使用 DMA 接收数据时, 应将 UART 的 FCR 寄存器 RFIFOT 配置为 0, 即 Rx FIFO 非空时, 就生成 UART Rx DMA 请求。并将 DMA 通道的 CHCTRL 寄存器的 SRCBURSTSIZE 位配置为 0, 即单次只进行 1 次传输。

对于使用限制 3, 可采用以下方法规避:

- 除非发生 FIFO 溢出, 不清空 UART Tx FIFO 和 Rx FIFO, 配置 DMA 响应已生成的 DMA 请求。
- 如果必须清空 UART Tx FIFO 或 Rx FIFO, 建议在清空 FIFO 后, 配置 DMA 进行一次 Dummy 数据传输操作, 响应可能生成的 DMA 请求。

2.15.3 修正情况

无。

2.16 E00019: PDMA 的图像缩放功能使用限制

2.16.1 问题描述

PDMA 的缩放功能在某些情况下会出现竖线:

- 当输入 PS[CTRL[FORMAT]] 为 ARGB8888 或者 RGB565 时, 如果 PS[CTRL[DECX]] \geq 0x1, 则缩放结果可能出现竖线。
- 当输入 PS[CTRL[FORMAT]] 为 YUYV1P422 时, 不论 PS[CTRL[DECX]] 为哪个值, 则缩放结果都有可能出现竖线。

2.16.2 规避方法

建议用户采用以下方法:

- 仅对 RGB565 或者 ARGB8888 格式的图像使用 X 轴缩放功能, 并将 PS[CTRL[DECX]] 设置为 0x0。

2.16.3 修正情况

无。

3 版本信息

日期	版本	描述
Rev1.0	2022/01/27	Rev1.0 发布。
Rev1.1	2022/04/15	Rev1.1 发布。 更新产品系列名称和型号列表。 更新文档封面和页眉，添加版本信息。 添加 E00010、E00011、E00012 和 E00013。
Rev1.2	2022/05/05	Rev1.2 发布。 添加 E00016。
Rev2.0	2023/03/03	Rev2.0 发布。 更新文档封面信息。 更新文档封面产品型号信息。 更新芯片版本 1.0 和 2.0 的修正情况。 添加 E00018、E00019。

表 2: 版本信息

免责声明

上海先楫半导体科技有限公司（以下简称：“先楫”）保留随时更改、更正、增强、修改先楫半导体产品和/或本文档的权利，恕不另行通知。用户可在先楫官方网站 <https://www.hpmicro.com> 获取最新相关信息。

本声明中的信息取代并替换先前版本中声明的信息。