

NV32F100x 开发引导及注意事项

Updated 2017. 9. 23

1. # 关于 NV32F100x 的相关资料: 例程包、原理图、及开发相关资料见官网 www.navota.com 及 QQ 群 123763203

建议参阅《NV32F100_参考手册》、《NV32F100_用户手册》以及“库函数中文说明”。

2. # 关于 NMI 引脚 (对应 PB4)

NMI 引脚又称不可屏蔽中断引脚, 低电平有效。开发布板过程中, 需留意此引脚, 建议外接 $4.7\text{K}\Omega$ – $10\text{K}\Omega$ 的上拉电阻。尽管在初始化过程后可以禁用 NMI, 但是在上电复位前, 检测到该脚为低电平, 就会进入 NMI 的中断程序。

3. # 关于倍频系数

当前版本的默认倍频系数为 1280, 不可选其它。下一版本增加 1024 与其它系数, 当前版本的时钟计算请参照库函数中文说明-《系统初始化说明》及《内部时钟管理》。

E. g: FEE 模式, 外部 10M 晶振, 先分频 256, 再倍频 1280, 得到主频= $10000000/256*1280 = 50000000$

4. # 关于用户使用库函数 `ETM_SetETMEnhance()`; 这个函数, 为什么使用 ETM2 可以, 而使用 ETM1 不行?

当前版本 ETM0, 1 是基础型, ETM2 是增强型 (寄存器较 0, 1 多)。访问不存在的寄存器地址会导致硬故障, 该函数为使能 `ETM2_MODE_ETMEN` 位, 而 ETM0, 1 无该寄存器。

ETM0 寄存器有效地址为 $0x40038000$ – $0x40038018$ (2 通道)

ETM1 寄存器有效地址为 $0x40039000$ – $0x40039018$ (2 通道)

ETM2 寄存器有效地址为 $0x4003A000$ – $0x4003A098$ (6 通道)

详细的寄存器说明请参照《NV32F100x-参考手册》-ETM 模块

5. # 关于内部上拉电阻的问题

内部上拉电阻 40–60K, 如果需要强上拉, 可以外接。上拉电阻在端口作为输入时有效, 输出无效, 配置上拉电阻寄存器: `PORT_PUEH/PORT_PUEL`

6. # 关于芯片无法正常下载的现象

首先查看电源和地是否短路, 排除这些因素外, 尝试识别内核并解锁, 详细信息查阅 NV32 解锁教程 (需要使用 JLINK)。MCU 进入低功耗模式后也会导致无法下载。

7. # 时钟校准

芯片出厂时主频校准至 48M, 使用内部 IRC 最高可达 60M, 但是建议使用主频在 55M 范围内 (联系管理员), 详细的时钟管理请见《内部时钟管理》。

8. # 关于 TSSOP20 的两个版本

使用 20 脚时应该注意版本, 一种为有晶振版本, 其 12、13 脚为 PB6、PB7; 一种为无晶振版本, 编号为 NV32F100xT20B, 其 12、13 脚为 PH0、PH1。详细说明请见《NV32F100_用户手册 V1.5》。

9. # 关于开漏引脚 PA2、PA3

该引脚为真开漏引脚 (true open-drain I/O), 该引脚上复用 UART0 需外部接上拉电阻进行数据传输, 注: UART0 也可复用到 PB0、PB1 引脚上 (无须接外部上拉)。

10. # 关于 ETM 模块的时钟问题

ETM 时钟可选外接时钟、总线时钟以及固定频率时钟 (ICSFFCLK)。

11. # 关于 NV32 的 IO 驱动

外接强电流外设, 应当做好隔离保护电路。IO 口的正常驱动能力为 3ma (3.3V), 高驱动口为 6ma (5V) 大电流引脚 8 个 20ma (需要手动开启, 配置寄存器请见 PORT_HDRVE)。

12. # 关于 FLASH 命令加载函数 EFM_LaunchCMD(uint32_t EFM_CMD);

该函数需要放置在 SRAM 中运行, 在 IAR 开发环境中使用 __ramfunc, 在 keil 中选定 flash.c 右击 -Options-Memory Assignment-Code/Const-IRAM1

13. # 关于 DEMO 板上接 PH2 上的 LED 灯一直亮的问题

默认状态下 PH2 上接总线时钟输出, 可在系统选项寄存器 (SIM_SOPT) 中配置。库函数中具体操作请参考库函数中文说明系统初始化部分。

14. # DEMO 板上部分引脚接有电容使用时需注意。

陆续更新-2016.12.29

15. # 对第 7 条的修正

后续芯片出厂时主频校准至 40M, 建议批量跑 40M 以内主频。


16. # 库函数工程以及自建工程修改

- (1) .sysinit.c 中 sysinit () 里修改 EFMCR &=0xFFFF1103;
- (2) .flash.c 中 Flash_Init () 里修改 EFMCR=(clkDIV<<24) + 0x00001103; //divide to 1M hz
- (3) .flash.c 中 EFM_LaunchCMD(uint32_t EFM_CMD) 中添加

```

#ifdef IAR
void __ramfunc EFM_LaunchCMD(uint32_t EFM_CMD)
#else
void EFM_LaunchCMD(uint32_t EFM_CMD)
#endif
{
    DisableInterrupts;
    if((EFMCR&EFM_DONE_MASK)== EFM_STATUS_READY)
    {
        EFMCR = EFM_CMD;
    }
    while(1)
    {
        if((EFMCR&EFM_DONE_MASK) == EFM_STATUS_DONE) break;
    }
    EnableInterrupts;
    __asm("nop");
}

```



17. # 在使用 EnableInterrupts; 指令，需在后加 __asm(“nop”); 指令

陆续更新-2017. 9. 23