

白皮书

FLASH烧写/编程

2

认识Flash

Flash分类

3

- 按照接口
 - 并行
 - 串行
 - SPI
 - I2C
- 按照架构
 - NAND
 - NOR

Flash 共性特点 / 与其他存储器比较

4

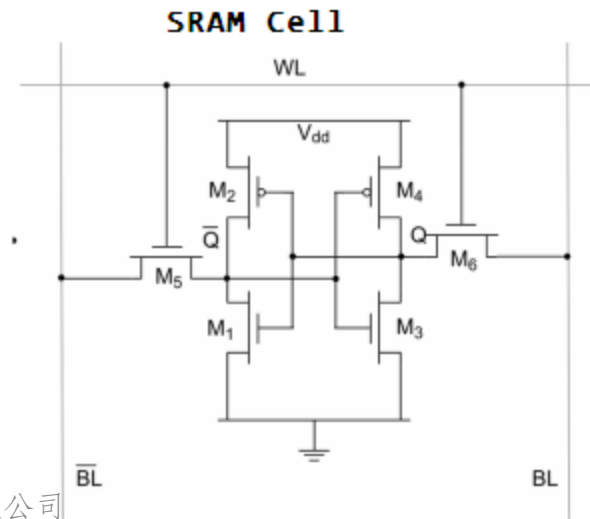
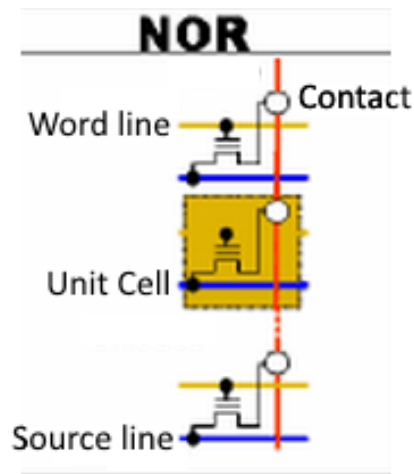
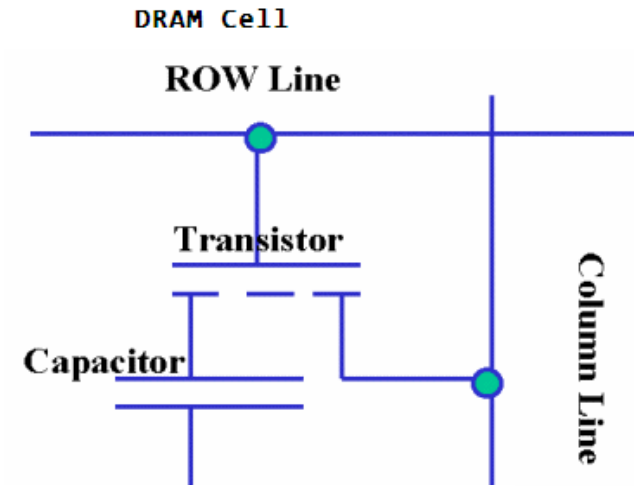
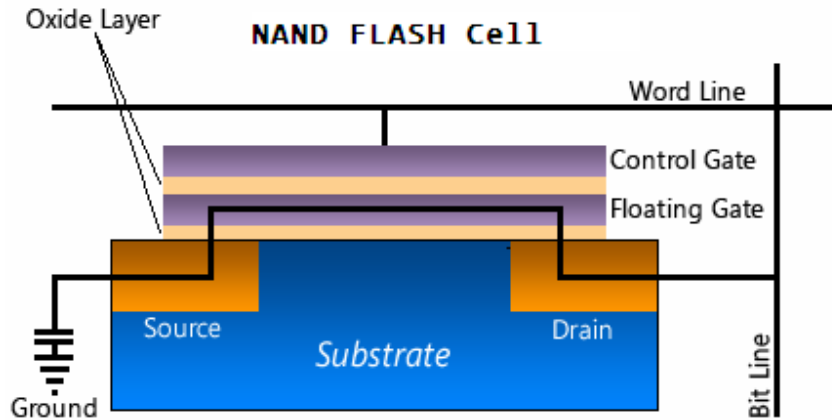
- 非易失
 - ▣ Flash vs. RAM
 - ▣ Flash vs. Battery-powered SRAM
- 电擦除
 - ▣ Flash vs. UV-EPROM
- 按块/扇区擦除
 - ▣ Flash vs. EEPROM
- 属于IC，电子设备
 - ▣ Flash vs. Hard Disk
 - ▣ 固态硬盘 SSD

5

NAND vs. NOR

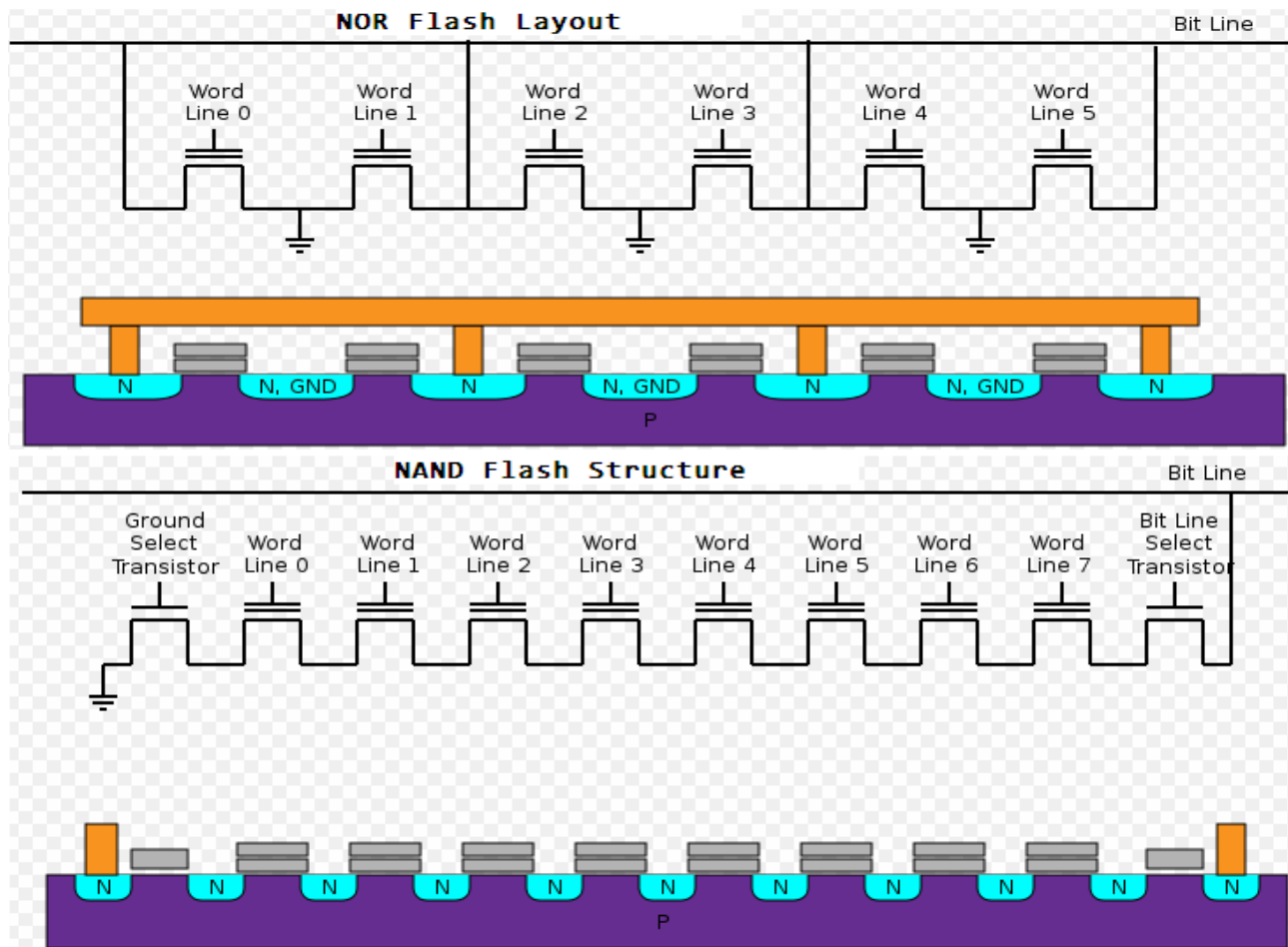
Memory Cells

6



结构

7



块大小

8

- NOR Flash: Typical block sizes are 64, 128, or 256 KB.
- NAND Flash: 每个块(Block)由一定数量的页(Page)组成。每个页的典型大小为512, 2K或4K字节。每个页中还有一些空间用(典型是12到16个字节)于存放纠错信息。Each block consists of a number of pages. The pages are typically 512 or 2,048 or 4,096 bytes in size. Associated with each page are a few bytes (typically 12–16 bytes) that should be used for storage of an error detection and correction checksum. While reading and programming is performed on a page basis, erasure can only be performed on a block basis. Another limitation of NAND flash is data in a block can only be written sequentially.

容量

9

- NAND Flash的单元尺寸几乎是NOR Flash器件的一半，NAND结构可以在给定的模具尺寸内提供更高的容量。
- NOR Flash常见的单Die容量到512MByte。
- NAND现在一般以Gbyte为单位。

性能

10

- NOR的读速度比NAND快。
- NAND的写入速度比NOR快。
- NAND的擦除速度比NOR快。
- NAND的擦除单元更小，相应的擦除电路更少。

TABLE 1 PERFORMANCE COMPARISON BY SPECIFICATION

	NOR multilevel cell (Mbytes/sec)	NAND 90-nm single-level cell (×8, large block) (Mbytes/sec)	Samsung OneNAND 90 nm (Mbytes/sec)	M-Systems mDOC 90-nm H1 (Mbytes/sec)
Read	108	16.2	108	9.5 (sustained)
Write	0.14	6.8	8.2	3.9 (sustained)
Erase (single)	0.11	64	64	NA
Erase (multiple)	0.11	NA	2	NA

Notes:

Samsung OneNAND and M-Systems' mDOC hybrid devices offer an alternative to NOR for booting systems. OneNAND is a system on chip, and mDOC is a multichip module.

NOR data courtesy of Samsung.

性能 (续)

11

Characteristic	NAND Flash MT29F2G08A	NOR (Q-Flash) MT28F128J3
Random access read	25 μ s (first byte) .03 μ s each for remaining 2111 bytes	.12 μ s
Sustained read speed (sector basis)	23 MB/s (x8) or 37 MB/s (x16)	20.5 MB/s (x8) or 41 MB/s (x16)
Random write speed	~300 μ s/2112 bytes	180 μ s/32 bytes
Sustained write speed (sector basis)	5 MB/s	.178 MB/s
Erase block size	128KB	128KB
Erase time per block (typ)	2ms	750ms

接口

12

- **NOR Flash**带有**SRAM**接口，有足够的地址引脚来寻址，可以很容易地存取其内部的每一个单元。
- **NAND**器件使用复杂的I/O口来串行地存取数据，各个产品或厂商的方法可能各不相同。通常用**8**个引脚用来传送控制/命令、地址和数据信息。

成本

13

- NAND Flash的单元尺寸几乎是NOR Flash器件的一半，由于生产过程更为简单，NAND结构可以在给定的模具尺寸内提供更高的容量，也就相应地降低了价格。
- 想想一个U盘多少钱？

可靠性和耐用性——寿命(耐用性)

14

- 在NAND闪存中每个块的最大擦写次数是一百万次，而NOR的擦写次数是十万次。NAND存储器除了具有10比1的块擦除周期优势，典型的NAND块尺寸要比NOR器件小8倍，每个NAND存储器块在给定的时间内的删除次数要少一些。

可靠性和耐用性——位交换/软错误

15

- 背景：
 - 所有存储器件都受软错误/位交换现象的困扰。在某些情况下，一个比特位会发生反转或被报告反转了。
 - 一位的变化可能不很明显（比如用来存储多媒体信息时不是致命的），但是如果发生在一个关键文件（操作系统、配置文件或其他敏感信息）上，这个小小的故障可能导致系统停机或者其他异常。为解决此问题，必须采用错误探测/错误更正(EDC/ECC)算法。
- 位反转的问题更多见于NAND闪存，NAND的供应商建议使用NAND闪存的时候，须同时使用EDC/ECC算法。This ECC may correct as little as one bit error in each 2048 bits, or up to 22 bits in each 2048 bits. If ECC cannot correct the error during read, it may still detect the error.
- NOR通常不需要。
- 题外话：你认为DRAM会有这个问题吗？

可靠性和耐用性——坏块处理

16

- **NAND**器件中的坏块是随机分布的。以前也曾有过消除坏块的努力，但发现成品率太低，代价太高，根本不划算。**NAND**器件需要对介质进行初始化扫描以发现坏块，并将坏块标记为不可用。
- **NOR**出现概率较小，一般不需要坏块管理。

易于使用

17

- **NOR Flash**可以非常直接地使用，可以像其他存储器那样连接，并可以在上面直接运行代码。
- 由于需要I/O接口，**NAND**要复杂得多。各种**NAND**器件的存取方法因厂家而异。在使用**NAND**器件时，必须先写入驱动程序，才能继续执行其他操作。向**NAND**器件写入信息需要相当的技巧，因为设计师绝不能向坏块写入，这就意味着在**NAND**器件上自始至终都必须进行虚拟映射。

软件支持

18

- 当讨论软件支持的时候，应该区别基本的读/写/擦操作和高一级的用于磁盘仿真和闪存管理算法的软件，包括性能优化。
- 在NOR器件上运行代码不需要任何的软件支持，在NAND器件上进行同样操作时，通常需要驱动程序，也就是内存技术(Memory Technology Device, MTD)驱动程序(MTD driver)。NAND和NOR器件在进行写入和擦除操作时都需要MTD driver。
- 使用NOR器件时所需要的MTD driver要相对少一些，许多厂商都提供用于NOR器件的更高级软件，这其中包括M-System的TrueFFS驱动，该驱动被Wind River System、Microsoft、QNX Software System、Symbian和Intel等厂商所采用。
- 驱动还用于对DiskOnChip产品进行仿真和NAND闪存的管理，包括纠错、坏块处理和损耗平衡。

硬件支持

19

- **NOR:** 支持SRAM接口的处理器都可以支持。
- **NAND:** 看CPU。
 - 举例
 - MPC850/MPC860/MPC8260/MPC8548: No
 - MPC8313: Yes

小结

20

- **NOR**的特点是芯片内执行(XIP, eXecute In Place), 这样应用程序可以直接在Flash闪存内运行, 不必再把代码读到系统RAM中。**NOR**的传输效率很高, 在1~4MB的小容量时具有很高的成本效益, 但是很低的写入和擦除速度大大影响了它的性能。**NOR Flash**在应用上则往往属于嵌入式的设计, 更重视其效能与可靠性, 元器件的市场需求与获利都相对较为稳定。
- **NAND**结构能提供极高的单元密度, 可以达到高存储密度, 并且写入和擦除的速度也很快。应用**NAND**的困难在于Flash的管理和需要特殊的系统接口。**NAND Flash**逐渐成为便携式产品的主要存储装置, 是高数据存储密度的理想解决方案。

21

如何烧写/编程

看不同阶段

22

- 研发调试
- 样机
- 大批量生产

编程方案——按供应商

23

- 专用编程器
 - 脱机
 - PC端软件+编程器
- 厂家提供的开发软件+仿真器/调试器
- 第三方软件+适配器//**We are here.**

24

不同方案比较

板卡上线前先烧写好Flash

25

- 好处：总时间最短，大批量发货可以选择
- 坏处：需要额外支付烧写费用，且因为芯片被拆封、烧片、重新装回，可能在加工焊接引入新问题，例如：
 - 静电损坏
 - 放置方向错误
 - 管脚变形
 - 漏烧/烧写不成功
 - 烧错了——这是真实案例，且不至一次，因为文件烧错了，Flash焊到板上以后CPU无法启动，且配置字也是错误的，导致仿真器无法连上。怎么办：把Flash全部焊下来换掉。
 - 上帝在关闭一扇门的同时，也打开了另一扇窗：最终客户找到我们，使用了哲发的yaJFPb方案，通过JTAG直接擦除重烧Flash。最大程度的挽回了时间，也降低了损失。

带座的BootROM+应用程序Flash

26

- 此方案某些高端CPU设计上可见
- 好处：研发人员认为调试方便，BIOS死了重烧下就行
- 坏处：
 - ▣ 嗯，这是10多年的设计方案。
 - ▣ 需要Flash座子（通常PLCC的），增加成本，降低可靠性。
 - ▣ 占用PCB面积。
 - ▣ 烧写时需要把BootROM抠下来，再到编程器上去烧写。
 - ▣ 如果最后只保留1片表贴的Flash，则需要设计bootrom和Flash间切换，需要设计0欧电阻跳线，调试需要更换电阻。
 - ▣ 量产发货板卡没有PLCC座子（否则可能接触不好，且成本也高），若现场返回的故障板，需要处理后才能分析。

通过ISP/IAP编程

27

- CPU带简易Bootloader
- 好处：不需要额外设备
- 坏处：需要CPU支持
- 渠道众多
 - ▣ 原厂软件(如NXP Flash Magic, Atmel SAM-BA)
 - ▣ 第三方软件
 - ▣ 自己应用程序

通过仿真器/调试器

28

- 好处：原厂方案。
- 坏处：
 - ▣ 仿真器太贵，尤其是高端CPU
 - 其实，哪怕是J-Link，正版的也不便宜
 - ▣ 软件，也是要付费的
 - ▣ 不接受定制开发
 - ▣ 如果时钟或RAM等问题甚至Flash数据出错，都可能导致仿真器无法连接，从而无法使用

第三方

29

□ 详见下文

30

哲发方案

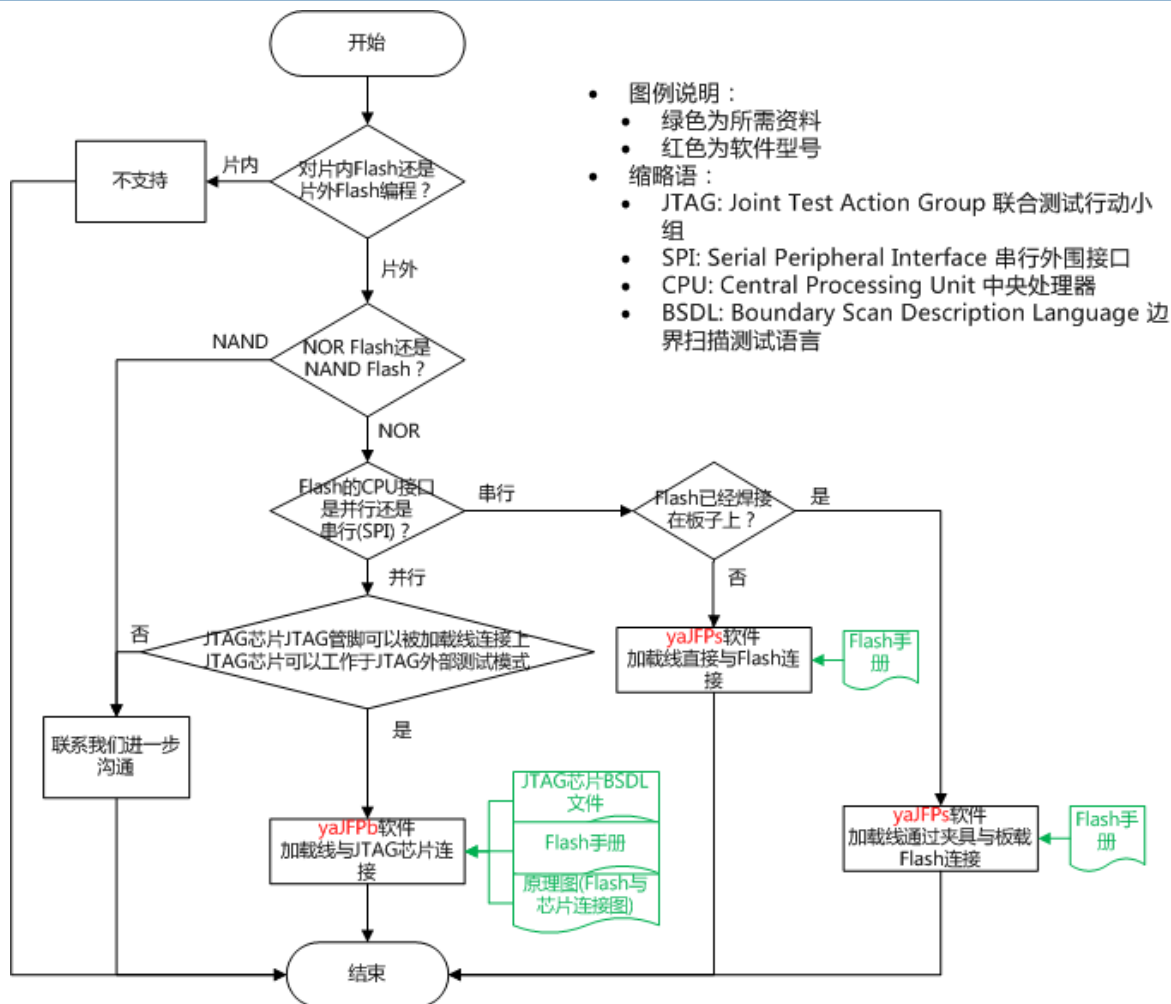
Flash编程相关产品

31

- **yaJFP: Yet Another JTAG Flash Programmer**
 - ▣ **2个商用版本**
 - **yaJFPb: Boundary Scan Test Sub-edition**
 - **yaJFPs: Serial Interface Sub-edition**
 - ▣ **2个内部版本**
 - **yaJFPe: EJTAG Sub-edition**
 - **yaJFPa: ARM Sub-edition**

选型指南

32



33

yaJFPb

产品特点

34

- 不需要使用带插座的Flash，提高可靠性；
- 不需要借助仿真器、调试器，不需要原厂软件，大大降低成本、提高生产灵活性；
- 轻量级绿色软件；
- 与CPU架构无关，支持PowerPC/ARM/MIPS等架构的CPU及DSP、NP、CPLD/FPGA或者其他SoC；
- 允许JTAG链上有多个器件；
- 使用CPU的JTAG接口，无需增加任何额外电路；
- 不需要CPU任何启动代码即可完成烧写；不需要配置CPU内部寄存器、初始化时钟内存等步骤；
- 即使CPU死机或者因Flash数据损坏导致板卡不能启动时，同样可以回读Flash内容，为维修调试提供依据；
- 使用WH-USB-JTAG电缆，加载Flash速度是并口电缆10倍左右。使用WH-USB-HiJTAG电缆，加载速度最快是WH-USB-JTAG电缆的5倍左右；

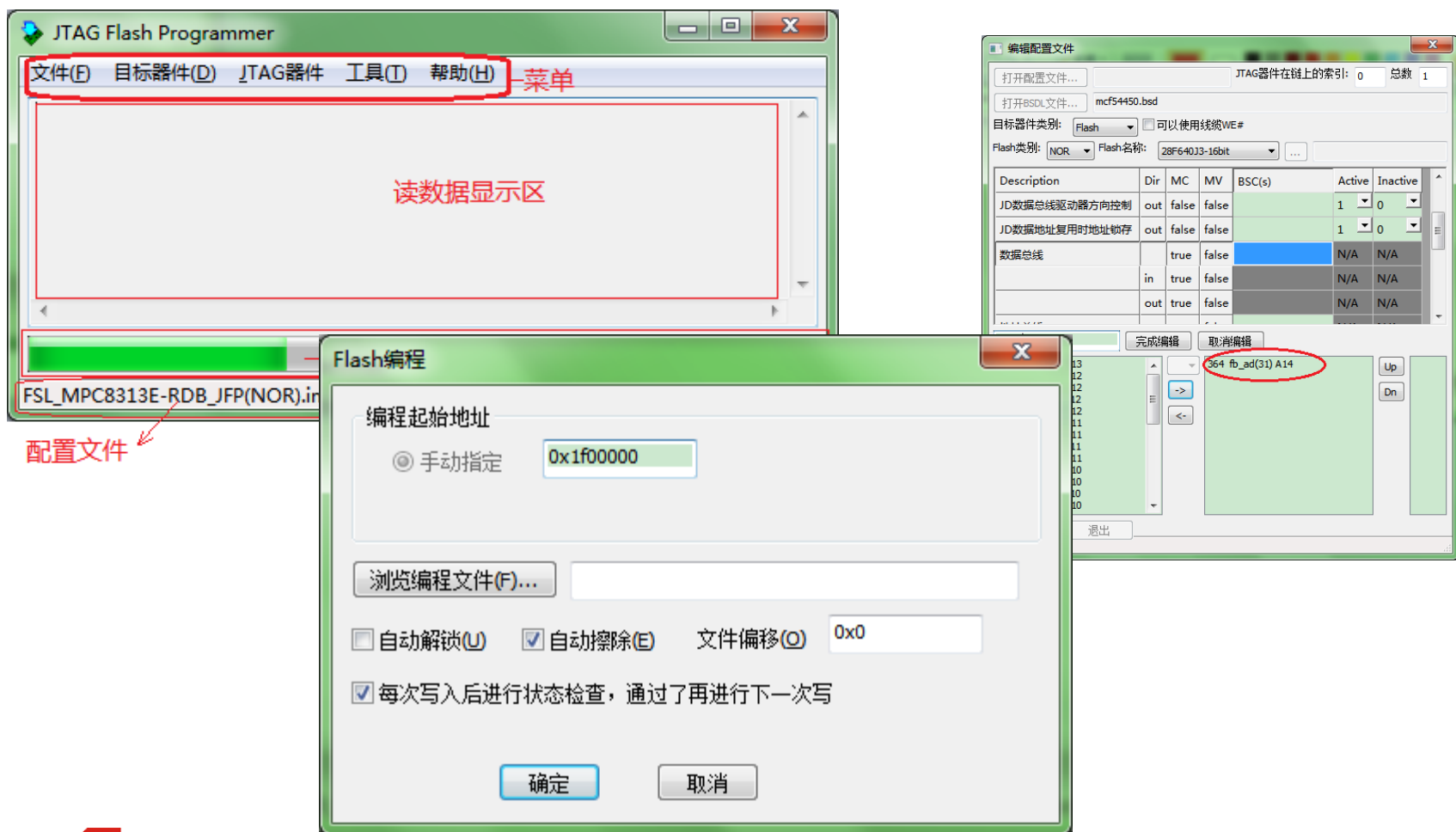
产品功能

35

- 支持 numonyx, Spansion, Intel, AMD, ST, SST, EON, 等厂家的并行NOR Flash, 且提供后续新Flash型号的全程支持;
- 对Flash读、编程操作;
- 支持块操作 (擦除、加锁、解锁) ;
- 除了Flash相关功能以外, 还可以通过点灯等操作验证JTAG器件的基本功能;
- 可以读写其他芯片;
- 通过程序可以读写与JTAG器件(通常是CPU)相连的其他芯片(如CPLD, ASIC等), 这样在CPU没有跑起来或没有软件的情况下甚至都可以去对单板做一些调试, 从而提高效率, 缩短开发周期;
- 软件也可以设置自动运行模式, 依次执行所有操作后退出程序, 无需人工干预, 方便与现有生产自动化系统集成;
- 支持对编程文件进行多种哈希计算, 并且可以与预设值进行匹配检查防止写入错误文件;
- 支持批量编程: 自动对不同地址烧写不同文件, 不需要人工干预;
- 针对整片 Flash Flash Flash烧写进行速度优化;

软件截图

36



37

FAQ

PP_Q1: 请问 JFP 烧写速度多快?

38

- 首先，一次操作所需时间与器件的 **BOUNDARY_LENGTH** 有关，**BOUNDARY_LENGTH** 越大则耗时越长。
- 其次，与加载线有关，如果使用 **WH-USB-JTAG** 电缆，则比并口线快十倍左右。如果使用 **WH-USB-HiJTAG**，烧写速度还可以更快，最高可以是 **WH-USB-JTAG** 的 5 倍。
- 再次，还跟 **CPU** 数据地址是否复用有关，如果复用的话则速度比其他条件相同时慢一倍（需要先输出地址锁存后再输出数据）。
- 第四，与 **Flash** 编程时内部写缓冲区大小有关。**Flash** 编程时内部缓冲区越大，速度越快。
- 最后，跟 **Flash** 与 **CPU** 相连的数据总线宽度有关，数据宽度越宽，效率越高。
- 一个参考：**BOUNDARY_LENGTH=476**，数据地址不复用，16位，32字节缓冲区，**WH-USB-JTAG**，烧写速度在 **6KiB/s** 左右。
- 又一个参考：**BOUNDARY_LENGTH=2281**，数据地址不复用，16位，32字节缓冲区，**WH-USB-HiJTAG**，烧写速度在 **10KiB/s** 左右；相应的，如果 **BOUNDARY_LENGTH=2281**，数据地址不复用，16位，32字节缓冲区，**WH-USB-JTAG**，烧写速度在 **2KiB/s** 左右。

PP_Q5: 有了JFP软件，我还需要具备哪些条件才能读写Flash?

39

- 硬件上，需要JTAG管脚可以被连接上、JTAG功能正常；文件上，需要Flash的芯片资料(Datasheet)和CPU(可以是其他SoC)的资料。
- 所需CPU的资料因为JTAG编程方法不同而不同，详细说明如下：
 - ▣ 方法1，通过标准的外部测试模式。此时需要CPU的BSDL文件和管脚连接关系(原理图)。yaJFPb使用这种方法。此时不使用CPU内部逻辑，可以把CPU当作一个仅有JTAG功能的空壳子。
 - ▣ 方法2，通过JTAG调试模式。如果没有原理图或者器件的BSDL文件，那么正常情况下是无法通过标准的外部测试模式来进行烧写的。此时，如果CPU支持EJTAG的调试(yaJFPe使用这种方法，厂家的开发环境使用仿真器也使用这种方法)，那么可以不要BSDL文件和原理图，相应的需要知道CPU的JTAG指令长度。

PP_Q6: 本方案与CodeWarrior+USB TAP烧写Flash相比, 优势是什么?

40

□ 就烧写Flash而言:

- 比CodeWarrior更好用! 无需初始化时钟/内存/内部寄存器。焊接没有问题, 基本上就可以对Flash编程烧写。即使配置字有错, 只要JTAG可用, 就可以操作Flash。
- 自动计算编程区域相应的扇区/块并可以自动在编程前擦除。
- 性价比更高: USB TAP多少钱? CWH-UTP-PPCC-HE: USD495. CodeWarrior最便宜的版本多少钱? CWP-BASIC-NL: USD995. ——但是, 如果只是烧写Flash, 那么您可以用不到上述两者总价几乎十分之一的价格就可以获得Flash编程功能的正版软件!
- 多数情况下烧写速度更高。
- 另外: 除了PowerPC, 还支持ARM/MIPS等所有架构的CPU以及CPLD/FPGA或者其他SoC。
- 轻量级绿色软件。
- 允许JTAG链上有多个器件。
- 支持定制开发。

□ 就生产/量产而言:

- CodeWarrior+USB TAP成本很高, 无法量产。而yaJFPb方案硬件成本很低, 软件批量购买价格更低。
- 软件启动可以自动加载相关板卡设置文件。
- 批量编程: 支持对不同地址烧写不同文件, 不需要人工干预。
- 软件一经启动, 就不需要关闭再运行。在烧写完成后板卡断电, 拔下CPU侧JTAG线缆, 换另一块板卡, 插上线缆, 就可以直接编程不需要重启软件或重新连接。
- 软件也可以设置自动运行模式, 依次执行所有操作后退出程序, 无需人工干预, 方便与现有生产自动化系统集成。

PP_Q8: 典型客户问题，都有哪些、能分享下吗？

41

- 这些来自客户真实案例的问题，最终通过哲发yaJFP系列产品得到了完美解决。
 - 先烧好Flash再贴片，可是程序烧错了，板子起不来，需要一片片吹下来更换；但更糟糕的是——复位配置字(RCW, Reset Configuration Word)不对，仿真器连不上，无法在线烧录；
 - 仿真器太贵，不巧的是还比较容易坏；
 - 每种架构的CPU/DSP都需要一种仿真器来烧写Flash；
 - 用PLCC封装的Flash做BootRom，每次都要拿出来放到编程器上烧写，且PLCC座子不可靠；
 - 开发软件庞大，安装和操作太麻烦，生产线工人用来烧Flash效率太低；
 - SDRAM焊接有问题（当然，后来才知道），仿真器无法烧写Flash；
- 哲发yaJFP(Yet Another JTAG Flash Programmer)Flash烧写软件，凝聚业界十多年经验精华，支持PowerPC(PPC)/MIPS/ARM等所有架构CPU或DSP以及任何厂家的CPLD/FPGA，轻量级绿色软件，便捷易用，支持全自动化操作，研发生产皆相宜。

还有问题.....

42

- 请联系我们。

43

谢谢!

<http://www.zhefar.com/>

杭州哲发科技有限公司