FPGA IAP程序设计说明

参考文档：

ug191:Virtex-5 FPGA Configuration User Guide

xapp1020:Post-Configuration Access to SPI Flash Memory with Virtex-5 FPGAs

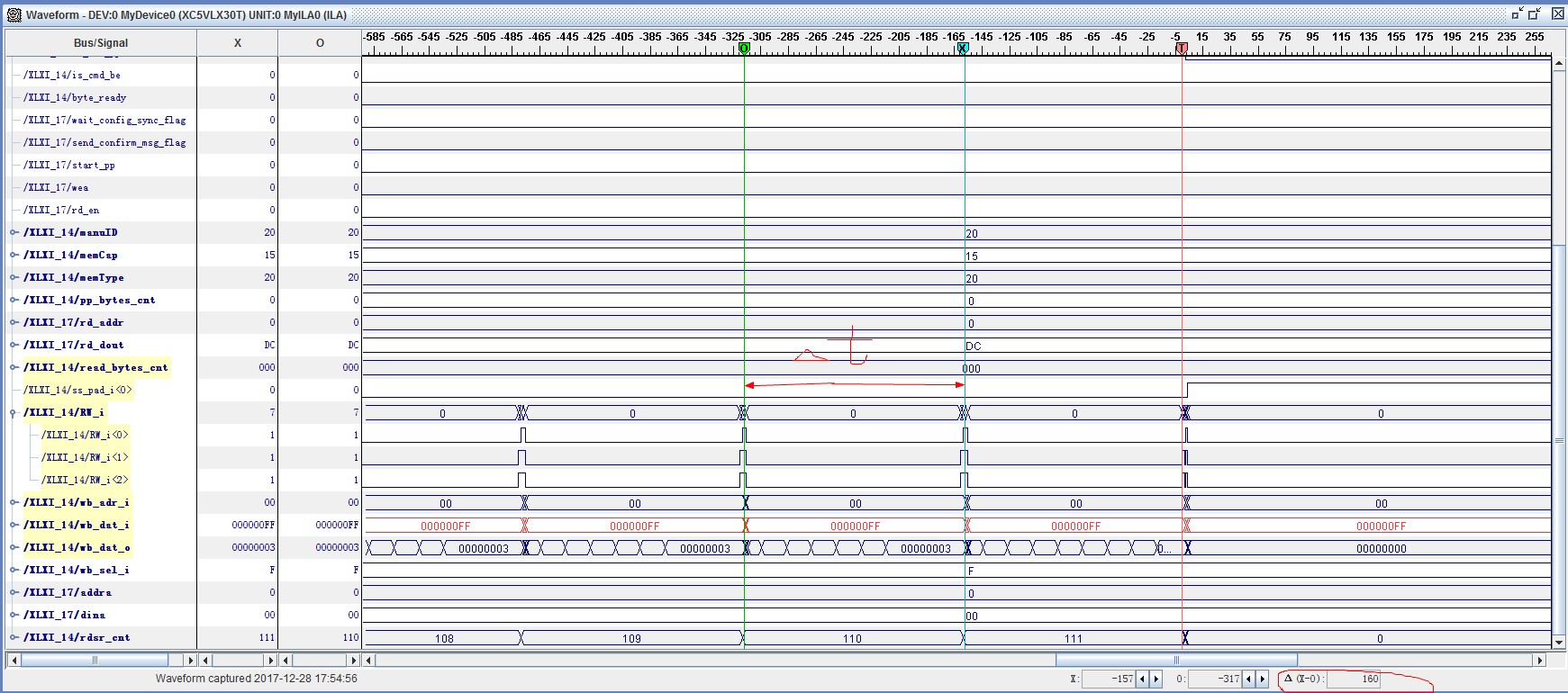
xapp1100:MultiBoot with Virtex-5 FPGAs and Platform Flash XL

M25P16\_datasheet

1. FLASH编程需要时间计算：

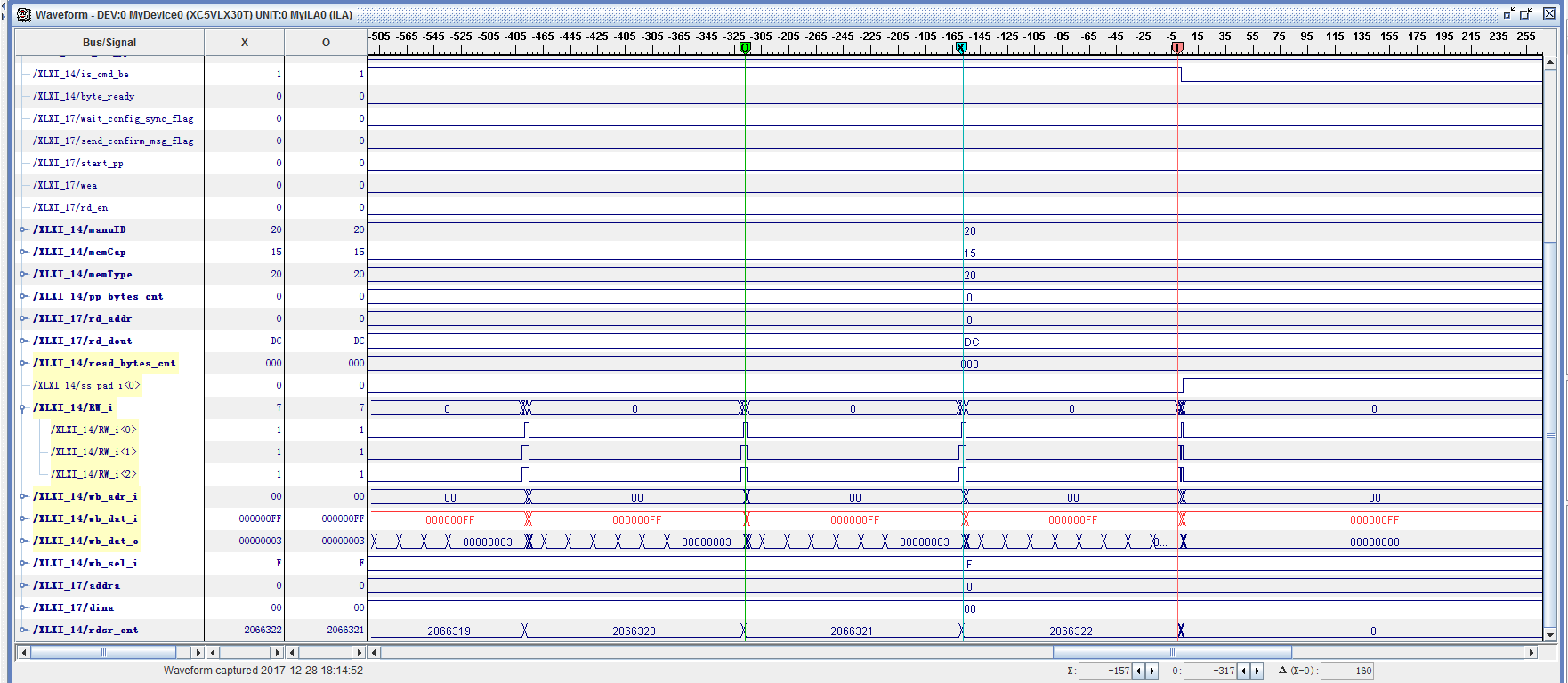
Chipscope抓取时钟为：25M(0.04us)。

页写指令(PP指令)，每一页256B数据写编程完成后，读取状态寄存器，等待变成完成，把256B数据从FLASH缓冲区写入到FLASH存储单元。从Chipscope抓取数据来看，rdsr\_cnt变量，记录读取状态寄存器次数，Δ(X-O)之间是160个采样时钟，即160个25M时钟，则一次读取占用时间160x0.04=6.4us，PP指令数据写入FLASH缓冲区后，开始读取状态寄存器，读取次数为112，则耗时112\*6.4=716.8us，数据手册提供典型值1.4ms，最大值5ms，写数据满足要求。

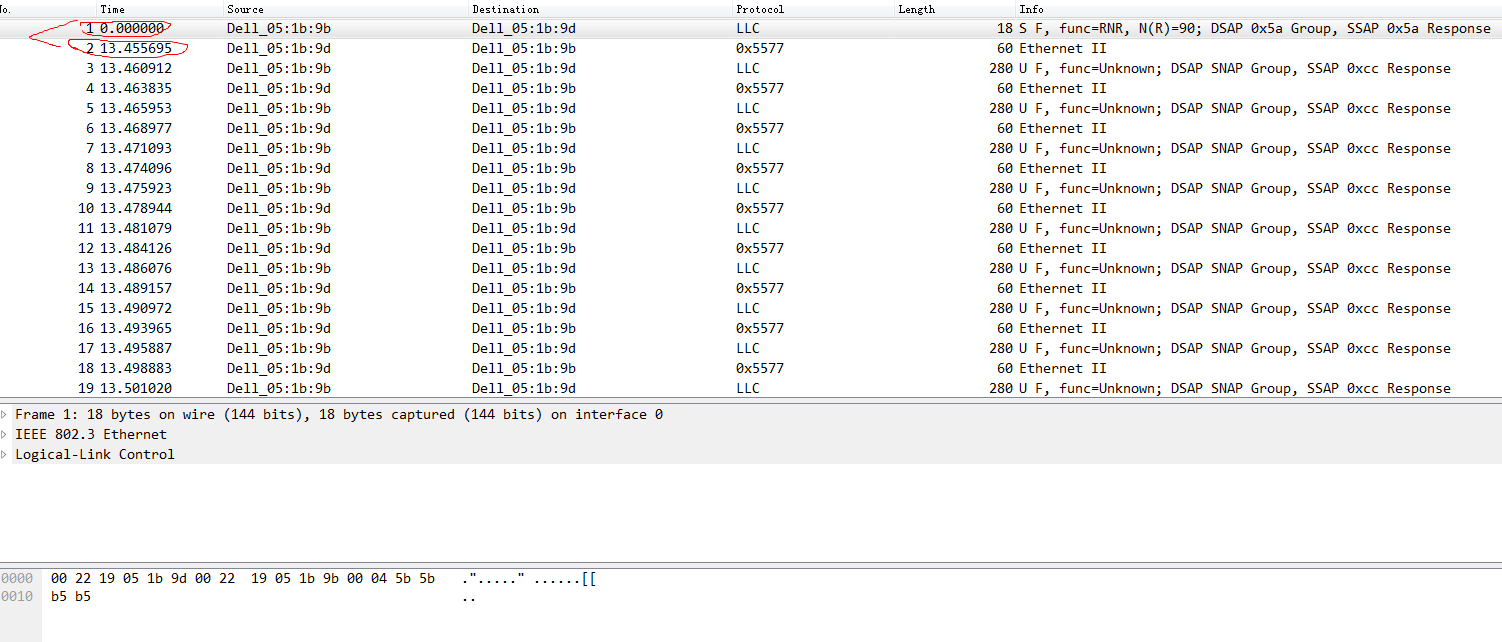


芯片擦除指令(Bulk Erase指令)，在开始编程时，需要把芯片内原有的配置程序擦除干净。FLASH编程只能把原有内容由1编程为0，而不能把0编程为1。要使FLASH单元内容从0编程为1，只有使用擦除指令。M25Pxx系列FLASH芯片，只有两种擦除指令，芯片整片擦除或者块擦除。这里使用整片擦除指令，把FLASH所有单元复位为1。整片擦除时间为：

2066323\*6.4=13.22s，而数据手册典型擦除时间为17s，最大40s。



根据wireshark抓包时间戳，下发擦除指令和完成擦除确认数据包间隔13.455s。数据擦除时间正常。

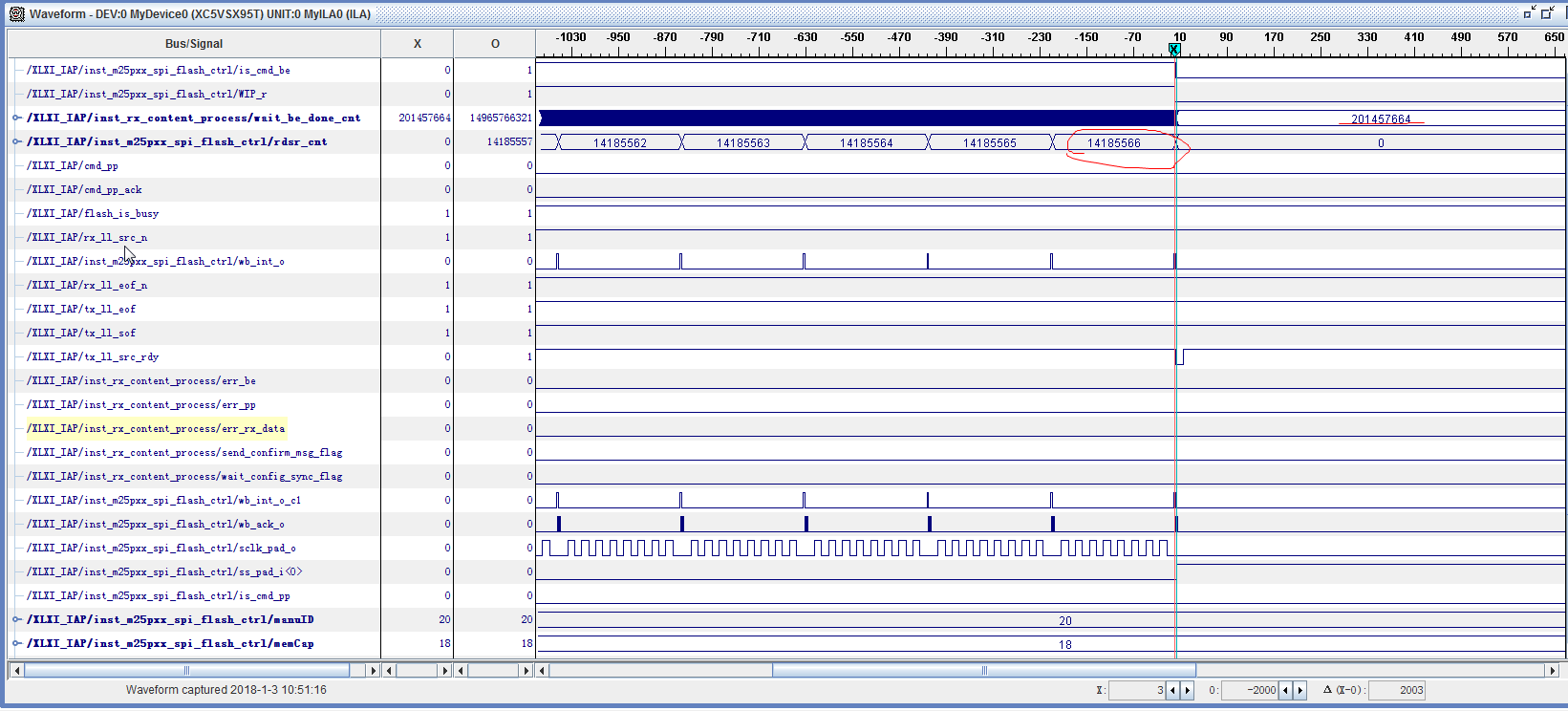


图？ Wireshark抓包时间

本程序模块主要使用了擦除指令和页写指令两个，这两个指令也是最耗时的指令，根据使用Chipscope检测得到结果，编程时间满足指标。

对于大容量FLASH，芯片擦除时间更长，此系列型号中，M25P128是容量最大的FLASH。

Chipscope抓取截图如下。完成擦除后，等待计数为：14185566，则消耗时间为14185566\*6.4=90.8s，



图？ M25P128 FLASH擦除时间

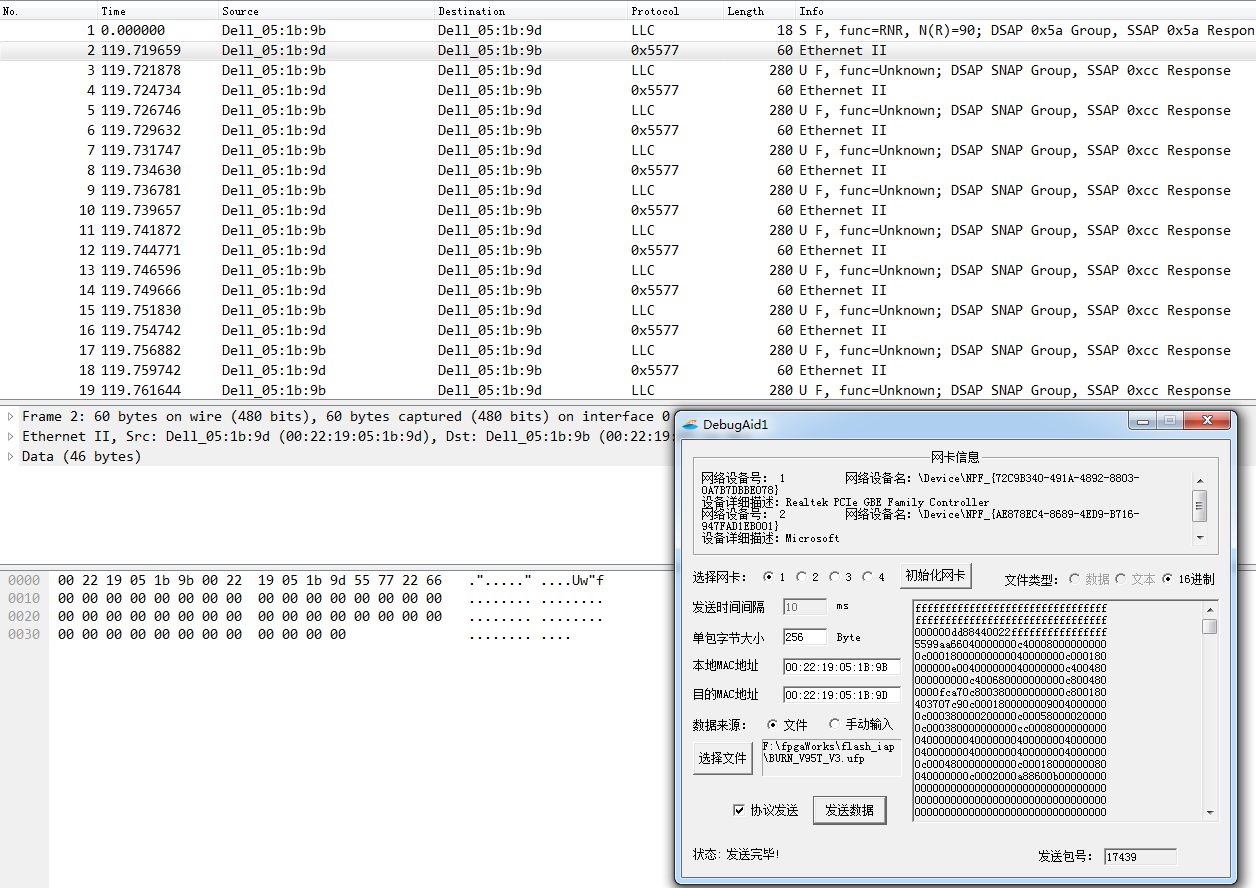
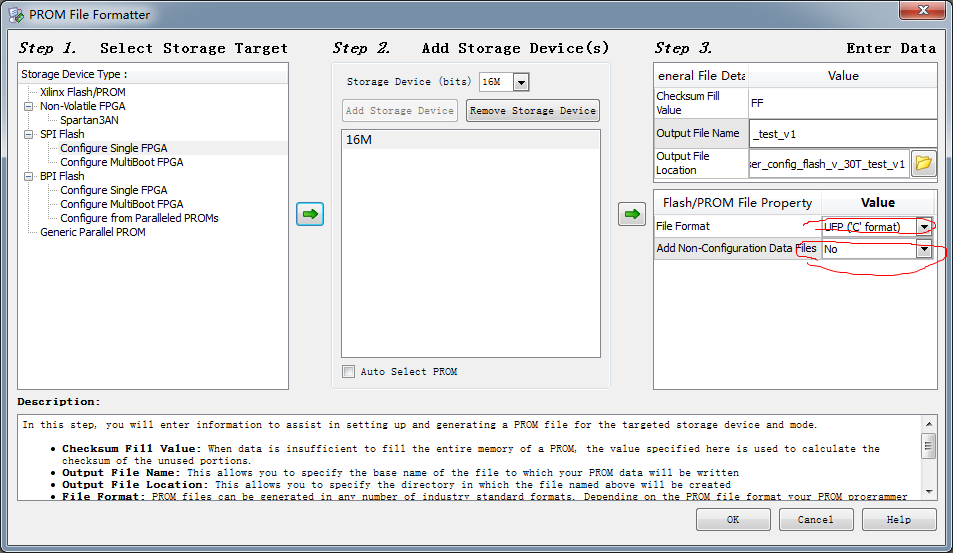


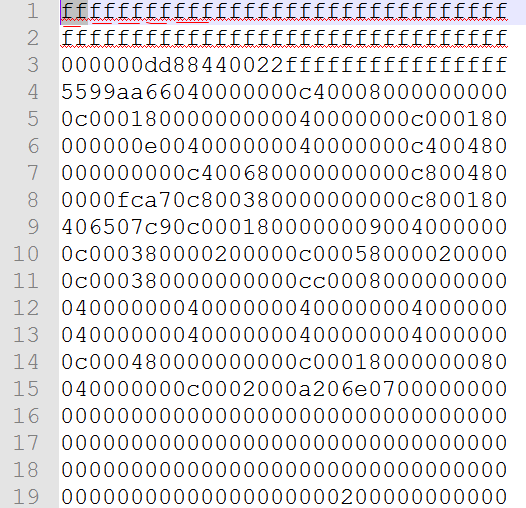
图 M25P128编程显控和wireshark抓包数据

1. 配置文件格式

本次编程配置文件为ASCII格式文本文件，改格式文本文件不包含任何配置数据，是纯配置文件的ASCII格式，产生的文件后缀名为：.ufp。产生该文件的方式为使用iMPACT，步骤与产生MCS格式文件相同，只是最后输出格式选择UFP(“C” Format)，同时不添加非配置数据。



则产生的.ufp则为纯ASCII格式的，可以用notepad++打开.ufp文件查看，文件内容中，两个字符组成1个16进制字节，是为一个配置字节。PC端控制程序则把ASCII数据读取，转换为16进制字节，下发到FPGA中完成配置。



1. 在线更新配置后热启动

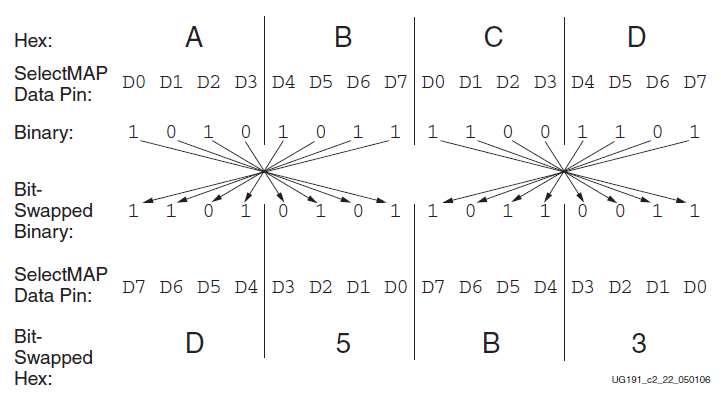
在线完成FLASH配置数据更新后，可以断电重启，完成FPGA配置；或者热配置，即在FLASH完成更新后，给出指令，调用ICAP\_VIRTEX5原语，内部控制FPGA配置专用引脚PROGRAM\_B和INIT\_B引脚等专用配置引脚产生脉冲信号，启动FPGA重配置流程。

热配置详细说明见Xilinx文档。

1. 软件协议

配置文件为ASCII格式的二进制配置文件，只包含了配置数据，去除了地址和校验信息。地址和校验信息是用于配置软件使用的，不用下载到FLASH文件内。

二进制配置文件是位倒置的(ug191)，如下图：



把配置数据从文件读出后，要反转后，发送到SPI FLASH。如从文件独到0xAB，则发送数据0xD5。位反转也是是否成功的关键。

千兆网下发数据，根据数据协议，进行数据擦除或者Page编程。

FPGA端处理流程如下：

PC端下发0x5B5BB5B5，开始芯片编程，完成芯片擦除操作。整片擦除不同容量器件擦除时间不同。M25P16擦除时间在15s左右。

完成擦除后，开始Page写入编程。

配置程序分包处理，每包256B数据。二进制配置数据去除了地址信息和校验信息，直接纯配置数据下发。FPGA把256B数据缓存，然后慢速时钟写入到FLASH。每包数据开始字节为0xABCD5B00。后续数据为数据包地址索引(FLASH编程页地址)、数据包长度。

每个数据包是256B分包。FLASH页是256B存储量。配置数据最后一包数据可能小于256B，由于FLASH擦出后内容为全1，则把最后一包数据在显控端补1，是配置尾数据为0xFF占位符，凑够256B，简化FPGA程序设计。

原有的PC端控制程序有部分BUG，对最后一包数据处理不当。FLASH是256B为1Page分页编程，则把配置数据按256B分包处理。以XC5VLX30T为例说明。XC5VLX30T芯片配置数据长度为：0x0011dfbf(1171391)个配置数据，则为0x0011df(4575)个数据包。可见，最后一包数据不是256的整数倍，则第0x0011df(4575)个数据包，里面有0xDF+1=0xC0个数据包即(192B数据)。

为便于统一处理数据包，可以对最后一包数据剩余部分使用0xFF填充，凑够256B数据。 因为FLASH内容擦除后默认就是0xFF。这样，所有的数据包都是256B，则FPGA端程序不用对最后一包数据做特殊处理。

每次FPGA端完成擦除、页写入指令，向PC端发送完成确认指令，发送确认数据为0x55772266。PC端在10ms时间内收到确认信息，则发送下一包数据，否则，重新下发指令或者下发数据。



图？ FPGA端数据处理流程

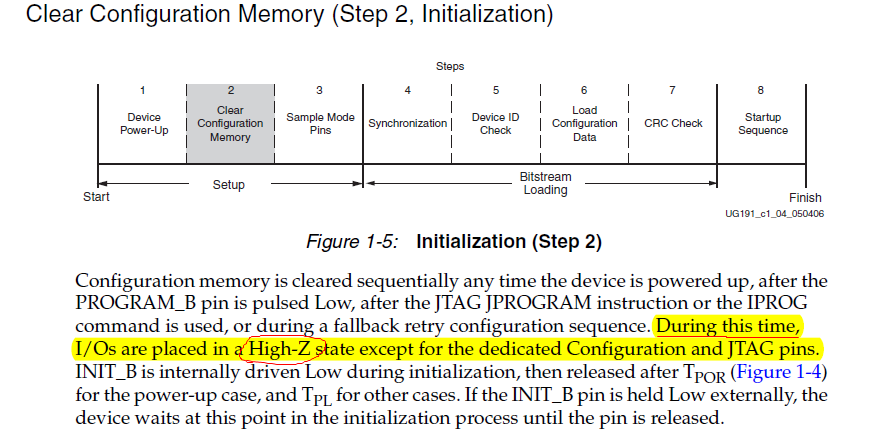


图？ PC端界面

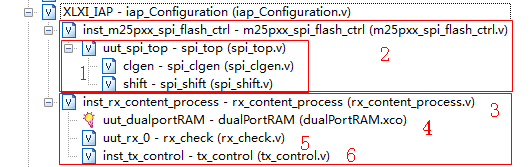
改进：本程序接收以太网下发的数据包完成FLASH页编程，但是是否写入正确没有确认。可以在写入完成后，把FLASH内数据读出，与接收的数据包对比，完成校验。

模块分为125M时钟和SPI模块时钟。SPI模块时钟速度低于25M，内部进行分频操作，SPI SCLK时钟为1M。SPI时钟不能太高。

上电配置过程中，一般IO引脚状态为 High-Z状态(ug191)：



IAP模块包含的文件，调用结构如下：



1) spi\_top.v表示SPI模块，从opencores网站下载，免费IP，wishbone接口；

2) m25pxx\_spi\_flash\_ctrl.v表示FLASH接口操作模块，根据指令和数据，完成芯片相关指令操作；

3) rx\_content\_process.v与PC接口，接收下发指令和配置数据，产生指令控制信号到m25pxx\_spi\_flash\_ctrl.v模块

4) 512B双口RAM，Xilinx IP核，使用到的接口如下：8bit数据宽度，512深度数据。在目标板上使用时，选择这些接口。

dualPortRAM your\_instance\_name (

.clka(clka), // input clka

.ena(ena), // input ena

.wea(wea), // input [0 : 0] wea

.addra(addra), // input [9 : 0] addra

.dina(dina), // input [7 : 0] dina

.clkb(clkb), // input clkb

.enb(enb), // input enb

.addrb(addrb), // input [9 : 0] addrb

.doutb(doutb) // output [7 : 0] doutb

);

5) rx\_check.v 网口接收数据过滤选项

显控目的地址和PC源地址设定，要与FPGA端一致。

测试中，设定FPGA地址：00-22-19-05-1b-9d

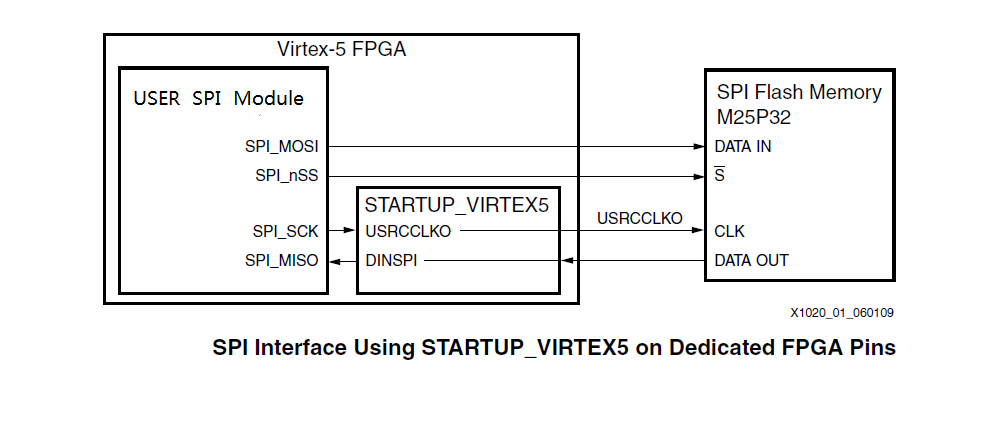
PC 地址：00-22-19-05-1b-9b

6) tx\_control.v，确认报文模块，FLASH模块操作成功后，上传确认报文，操作失败，则不发确认报文，PC端计时会超时溢出；

7) iap\_Configuration.v模块是顶层模块，负责通信接口与SPI FLASH器件接口

8) mux\_tx.v模块，在有多个通信接口时，选择输出优先级。在IAP启动后，IAP程序具有最高等级。

功能框图：



正常情况下，配置接口是专用引脚，配置SPI接口对用户是不可访问的。要使用SPI接口，则需要使用Xilinx 原语STARTUP\_VIRTEX5作为接口，实现对SPI CLK和DIN数据接口操作。SPI\_MOSI和SPI\_nSS是对用户可访问的。

完成USER SPI Module，在上电正常配置完成后，即可完成对外部SPI的重配置。