使用 ICAP 原语实现 SPI-Multiboot 加载

修订历史记录

Date	Version	Revision
2020/06/11	V1.0	ICAP 实现 SPI-Multiboot：A→B→A

1 重配置Multiboot的应用需求

当多个应用程序同时在一个硬件平台上实现时，各个程序的资源使用和数据通路可能会冲突，这增加了控制电路设计的复杂程度，给开发人员增加了工作量和开发难度。通过多重配置，可以将多个应用程序根据需要分时加载到 FPGA 中，不仅精简了电路设计，而且使系统更加灵活。FPGA 多重配置的特点可以让特定条件下的用户选择片上资源不多的 FPGA 去实现需要很多资源 FPGA 才能实现的功能，这大大降低了开发费用，同时提高了 FPGA 的利用率。

2 ICAP实现Multiboot的原理介绍

FPGA 具有多重配置的特性，允许用户在不掉电重启的情况下，根据不同时刻的需求，可以从 FLASH 中贮存的多个比特文件选择加载其中的一个，实现系统功能的变换。

当 FPGA 完成上电自动加载初始化的比特流后，可以通过触发 FPGA 内部的多重启动事件使得 FPGA 从外部配置存储器(SPI FLASH)指定的地址自动下载一个新的比特流来重新配置。FPGA 的多重配置可以通过多种方式来实现。本文采用的是基于 ICAP 原语的状态机编码方式。通过调用 Xilinx 自带的 ICAP 原语，编写状态机按照一定的指令流程对 ICAP 原语进行不断的配置，可以控制 FPGA 重新配置。

2.1 IPROG指令序列

在调用了 ICAP 原语接口之后，通过 Verilog/VHDL 编码的方式实现状态机。通过状态机发送 IPROG 指令给 ICAP 原语，ICAP 原语在接收到这些指令后会根据指定的地址自动加载配置文件。

IPROG 指令的作用跟外部 Program_B 管脚的作用类似，都是对 FPGA 芯片进行复位操作，该复位操作对 FPGA 内部的应用程序进行复位，复位过程中除专用配置管脚和 JTAG 管脚，其他输入/输出管脚均为高阻态，同时 IPROG 指令不能复位专用重配置逻辑，如 WBSTAR 寄存器、TIMER 寄存器、BSPI 寄存器和 BOOTSTS 寄存器。IPROG 指令能够触发 FPGA 开启初始化流程，同时拉低 INIT 和 Done 信号。完成复位操作后，将默认的加载地址用热启动地址寄存器(Warm Boot Start Address，WB-STAR)中的新地址替换。

2.1.1 IPROG指令包

Table 7-1: **Example Bitstream for IPROG through ICAPE2**

Configuration Data (hex)(1)	Explanation
FFFFFFFF	Dummy Word
AA995566	Sync Word
20000000	Type 1 NO OP
30020001	Type 1 Write 1 Words to WBSTAR
00000000	Warm Boot Start Address (Load the Desired Address)
30008001	Type 1 Write 1 Words to CMD
0000000F	IPROG Command
20000000	Type 1 NO OP

Notes:

1. See Parallel Bus Bit Order, page 83.

After the configuration logic receives the IPROG command, the FPGA resets everything except the dedicated reconfiguration logic, and the INIT_B and DONE pins go Low. After the FPGA clears all configuration memory, INIT_B goes High again. Then the value in WBSTAR is used for the bitstream starting address. The configuration mode determines which pins are controlled by WBSTAR.

2.1.2 WBSTAR寄存器

WBSTAR 寄存器的组成如下图所示。

Table 5-34: **WBSTAR Register**

Description	RS[1:0]		RS_TS_B	START_ADDR																									
Bit Index	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-35: **WBSTAR Register Description**

Name	Bit Index	Description
RS[1:0]	[31:30]	RS[1:0] pin value on next warm boot. The default is 00.
RS_TS_B	29	RS[1:0] pins 3-state enable. 0: 3-state enabled (RS[1:0] disabled) (default) 1: 3-state disabled (RS[1:0] enabled)
START_ADDR	[28:0]	Next bitstream start address. The default start address is address zero.

Table 7-2: **WBSTAR Controlled Pins According to Configuration Mode**

Configuration Mode	Pins Controlled by WBSTAR
Master Serial	RS[1:0]
Master SPI	START_ADDR[23:0] is sent to the SPI device serially. (1)
Master BPI	RS[1:0], A[28:00]
Master SelectMAP	RS[1:0]
JTAG	RS[1:0]
Slave SelectMAP	RS[1:0]
Slave Serial	RS[1:0]

Notes:

1. When using ICAPE2 to set the WBSTAR address, the 24 most significant address bits should be written to WBSTAR[23:0]. For SPI 32-bit addressing mode, WBSTAR[23:0] are sent as address bits [31:8]. The lower 8 bits of the address are undefined and the value could be as high as 0xFF. Any bitstream at the WBSTAR address should contain 256 dummy bytes before the start of the bitstream..

In all configuration modes except SPI mode, RS[1:0] is controllable by WBSTAR. The START_ADDR field is only meaningful for the BPI and SPI modes.

（1）对于 BPI 模式来说，可以通过 RS[1:0]来控制具体位流的读取，也可以通过 STAT_ADDR[28:0]地址来控制具体位流的读取。

（2）对于 SPI 模式来说，只有 STAT_ADDR[23:0]地址来表征 FLASH 器件的地址，当使用 32 位地址的 SPI（容量大于等于 256Mb）时，需要将时间存储的高 24 地址赋值给 STAT_ADDR[23:0]。因此在位流存储的起始地址早于 255 时，这就要求位流中的 dummy 数目要大于 256 个，否则就会出现易失部分有效位流读取，导致加载失败。为了安全起见，在使用大于等于 256Mb 的 FLASH 时，可以适当在位流头前加入 Dummy。

2.2 ICAP原语接口时序

ICAP 原语接口时序跟 Select Map 接口时序非常相似。SelectMap 模式下，FPGA 的配置和回读是通过 CSI_B,RDWR_B 和 CCLK 来控制的。当 CSI_B=0 时，Select Map 接口使能；当 CSI_B=1 时，Select Map 接口不使能。当 RDWR_B=0 时，数据端口为输入，功能为配置 FPGA；当 RDWR_B=1 时，数据端口为输出，功能为回读 FPGA。SelectMap 模式下，当 CSI_B=0 时，RDWR_B 由 0 变为 1 或者由 1 变为 0 都会触发 ABORT。当 CSI_B=0 时，RDWR_B 由 0 变为 1 的 ABORT 现象是配置 IO 由输入变为输出，同时 ABORT 的状态信息体现在数据管脚上，此时数据管脚为输出。当 CSI_B=0 时，RDWR_B 由 1 变为 0 的 ABORT 现象是配置 IO 由输出变为输入，这时 ABORT 的状态信息无法体现在数据管脚上，因为此时数据管脚为输入。

为了避免 ABORT 发生，建议在 CSI_B=0 前，设置好 RDWR_B=0 或 RDWR_B=1，即通过控制 CSI_B 信号来控制配置或回读，或者通过控制 CCLK 供给来控制配置或回读。比如配置 FPGA 的三种接口时序。

2.2.1 连续的配置接口时序

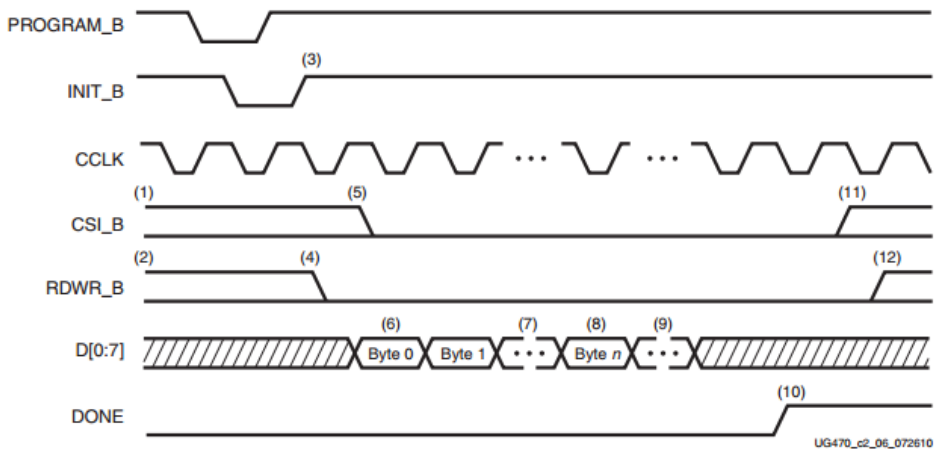


Figure 2-6: **Continuous x8 SelectMAP Data Loading**

2.2.2 CSI_B控制的断续配置接口时序

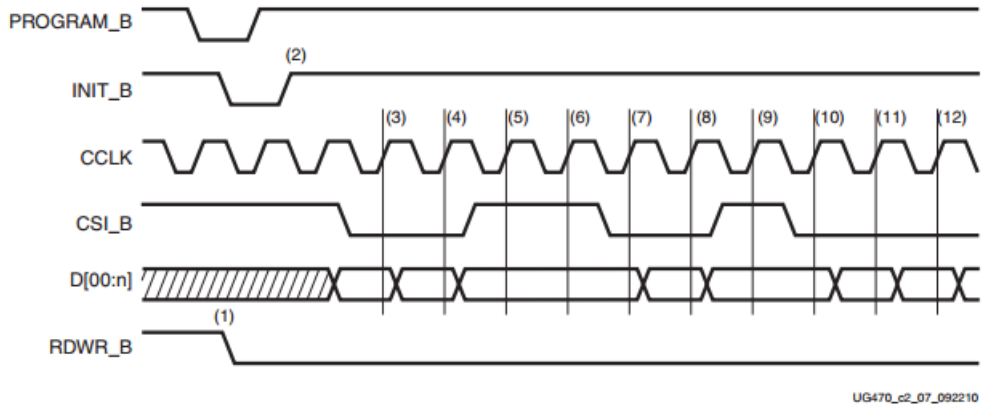


Figure 2-7: **Non-Continuous SelectMAP Data Loading with Free-Running CCLK**

2.2.3 CCLK控制的断续配置接口时序

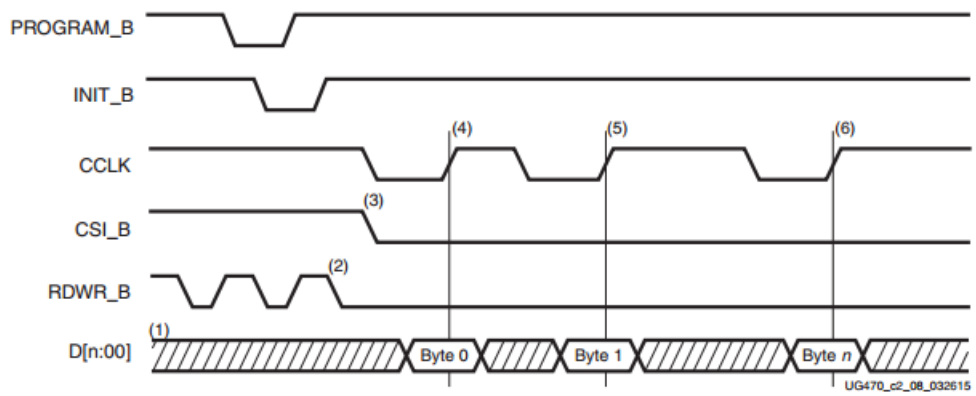


Figure 2-8: **Non-Continuous SelectMAP Data Loading with Controlled CCLK**

2.3 SelectMap数据顺序

Select Map 有 X8、X16 和 X32 模式。Select Map 的位顺序的关系如下表所示。

Table 2-11: **Bit Ordering**

SelectMAP Data Bus Width	Data Pins																											
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
x32	24	25	26	27	28	29	30	31	16	17	18	19	20	21	22	23	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
x16																	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
x8																									0	1	2	3	4	5	6	7

X16 数据变换关系如下图所示，X8 和 X32 类推。

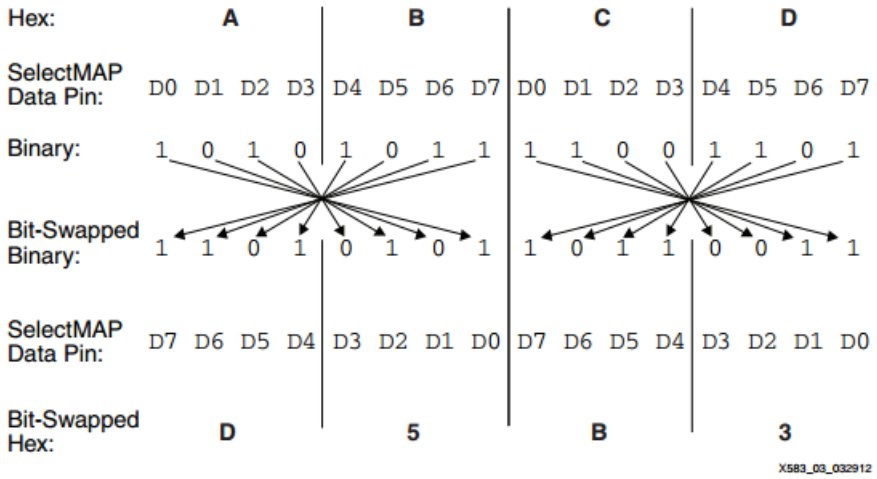


Figure 3: **Bit-Swapping Example for x8**

因此 IPROG 的指令序列对应的实际 Select Map 数据如下表所示。

序号	IPROG 指令序列	Select Map	指令含义
1.	FFFFFFFF	FFFFFFFF	Dummy Word
2.	AA995566	5599AA66	Sync Word
3.	20000000	04000000	Type1 Noop
4.	30020001	0C400080	Write 1 Word to WBSTAR
5.	XXXXXXXX	XXXXXXXX	WBSTAR（期望地址）
6.	30008001	0C000180	Write 1 Word to CMD
7.	0000000F	000000F0	IPROG 指令
8.	20000000	04000000	Type1 Noop
9.	20000000	04000000	Type1 Noop

3 ICAP实现Multiboot的工程开发

3.1 功能需求

在 SPI 的 flash 里烧写有 A 和 B 两个程序，FPGA 上电后，自动加载 A 程序，根据外部给 FPGA 指示信号，FPGA 自动切换加载 B 的程序，同时在 B 程序运行期间，根据外部给 FPGA 指示信号，FPGA 自动切换加载 A 的程序。

3.2 验证平台

KC705 的评估板，使用的 FLASH 为 N25Q128。

3.3 接口协议

3.3.1 数据包

ICAP 传输的数据包即满足 SelectMap 数据顺序要求的 IPROG 指令包。

序号	IPROG 指令序列	Select Map	指令含义
程序 A 中的 IPROG 指令序列包			
1.	FFFFFFFF	FFFFFFFF	Dummy Word
2.	AA995566	5599AA66	Sync Word
3.	20000000	04000000	Type1 Noop
4.	30020001	0C400080	Write 1 Word to WBSTAR
5.	00800000	00100000	WBSTAR（期望地址）
6.	30008001	0C000180	Write 1 Word to CMD
7.	0000000F	000000F0	IPROG 指令
8.	20000000	04000000	Type1 Noop
9.	20000000	04000000	Type1 Noop
程序 B 中的 IPROG 指令序列包			
1.	FFFFFFFF	FFFFFFFF	Dummy Word
2.	AA995566	5599AA66	Sync Word
3.	20000000	04000000	Type1 Noop
4.	30020001	0C400080	Write 1 Word to WBSTAR

5.	00000000	00000000	WBSTAR（期望地址）
6.	30008001	0C000180	Write 1 Word to CMD
7.	0000000F	000000F0	IPROG 指令
8.	20000000	04000000	Type1 Noop
9.	20000000	04000000	Type1 Noop

指令序列包中的 WBSTAR 期望地址的计算方法：由于需要在 128Mb 的 SPI-FLASH 中存储 2 份位流，因为 SPI-FLASH 是按照字节存储的，因此 128Mb 的 SPI-FLASH 大小也为 16MB，现计划每份位流存储大小平均分配，均为 8MB,远小于实际 7K325T 的位流大小，因此需要对位流进行压缩。

第 1 份位流存储的地址为 0X000000~0X7FFFFF，对应程序 A 的地址；

第 2 份位流存储的地址为 0X800000~0XFFFFFF，对应程序 B 的地址；

位流号	WBSTAR		WBSTAR	Select Map 的 WBSTAR
	SPI 不可用地址[31:24]（16 进制）	SPI 可用地址[23:0]（16 进制）		
1	0X00	0X000000	0X00000000	0X20000000
2	0X00	0X800000	0X00800000	0X30020000

3.3.2 接口时序

采用连续的配置接口时序。

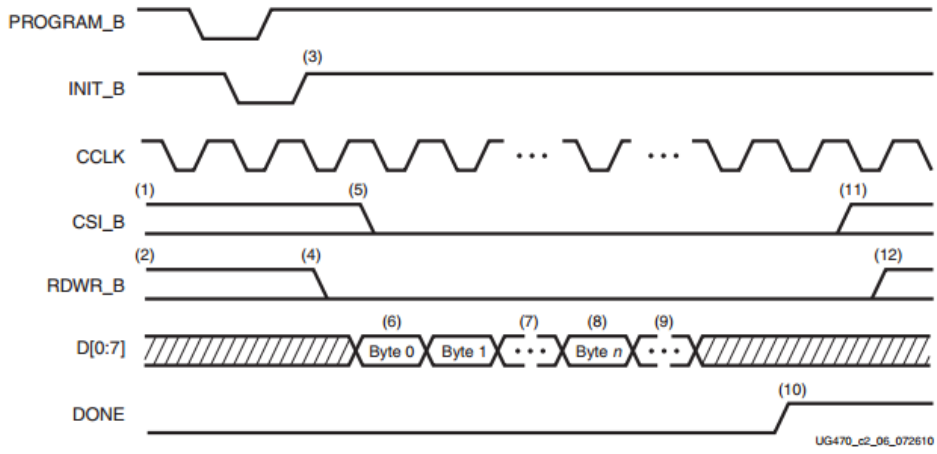


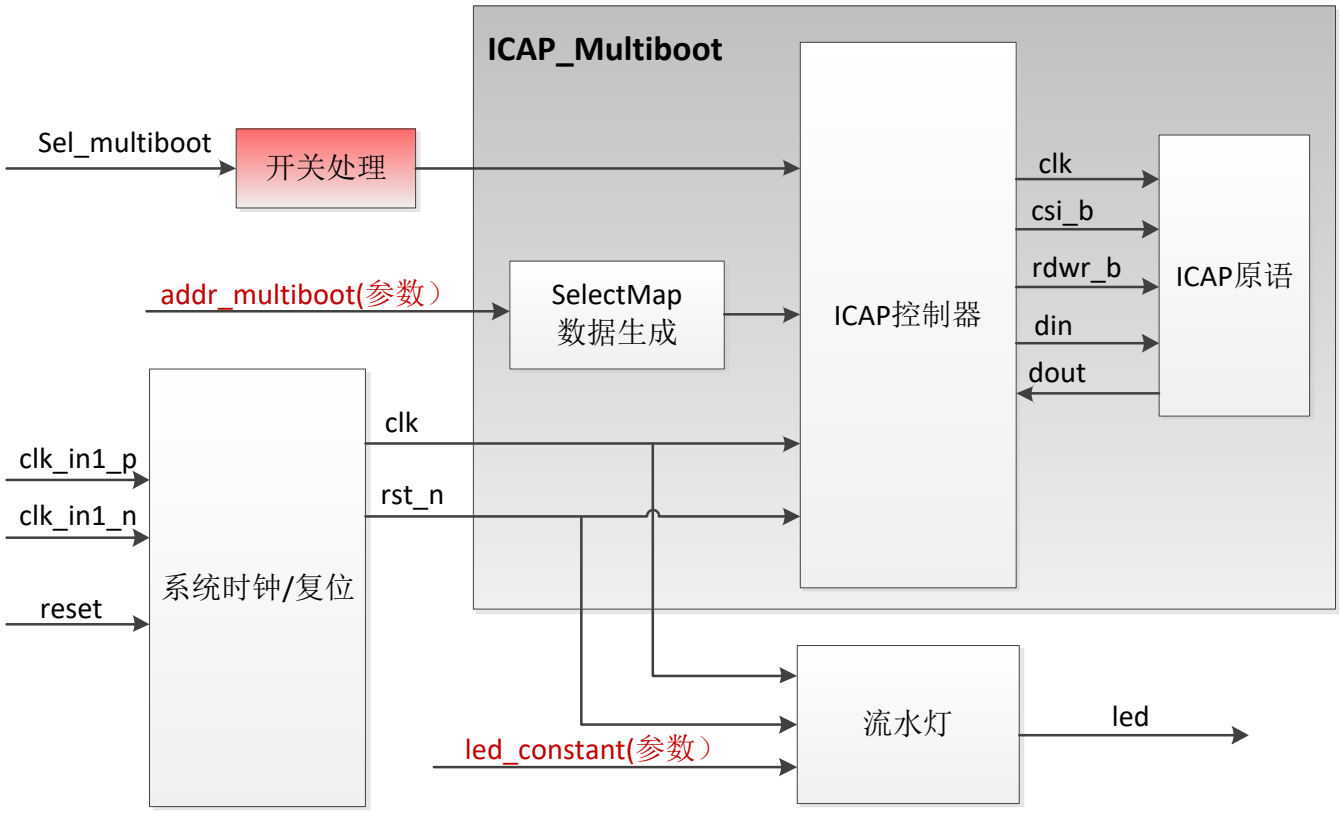
Figure 2-6: **Continuous x8 SelectMAP Data Loading**

3.4 工程实现

3.4.1 系统组成

整个系统工程包括两个工程，分别为 Golden 工程和 Update 工程，为了减少开发时间，此 Demo

程序中的 Golden 工程和 Update 工程很相似，Golden 工程和 Update 工程在此次案例中差异见下图中的红色部分。



注：红色部分表示Golden工程和Update工程存在差异

各功能块的功能描述如下。

（1） 系统时钟/复位：对输入的时钟和复位信号进行处理，得到满足要求的系统时钟和系统复位信号。

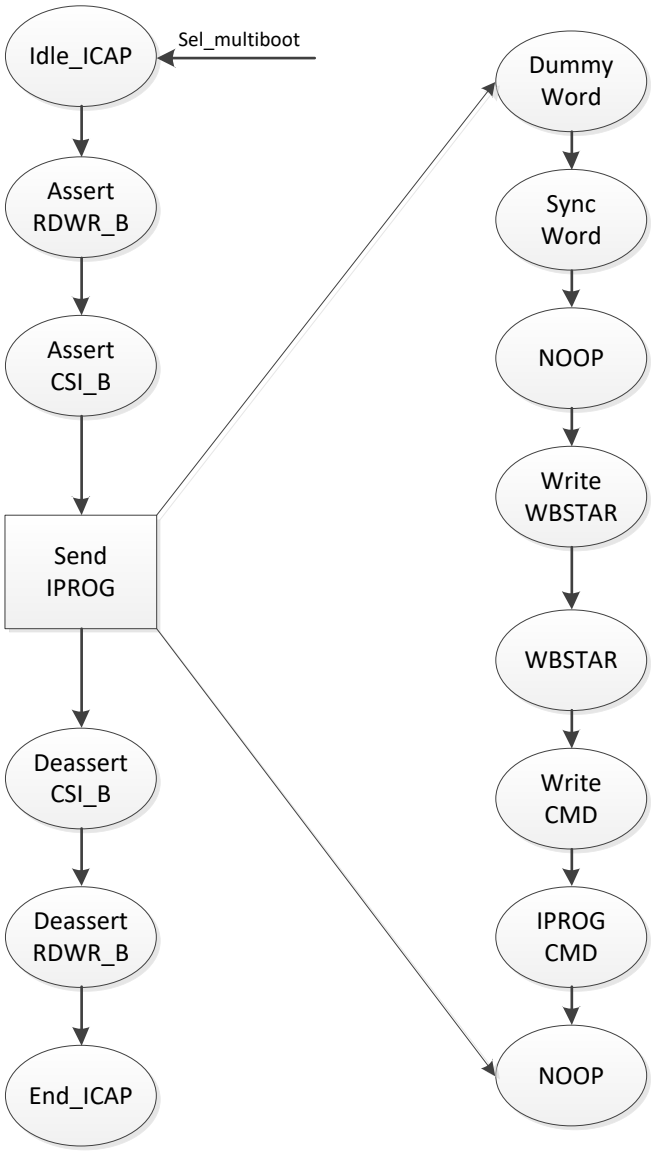
（2） 开关处理：对输入的 Sel_multiboot 进行简单处理，Golden 工程是直接输出给 ICAP 控制器，Update 工程是取反后输出给 ICAP 控制器。

（3） 流水灯：此功能是为了验证 Golden 工程和 Update 工程切换是否成功，两个工程的流水灯的功能差异通过输入的参数 led_constant 来体现。

（4） ICAP_multiboot：此功能模块通过 ICAP 原语给 FPGA 的配置电路发送 IPROG 指令序列，触发 FPGA 执行多重加载功能，其中 SelectMap 数据生成是用来将输入的指令进行 bit_SWAP 转换，从而满足 SelectMap 传输要求，ICAP 控制器是将输入的 SelectMap 数据以 SelectMap 连续配置接口时序发送给 ICAP 原语，从而达到触发 Multiboot 的功能，其中输入的参数 addr_multiboot 表示即将切换程序的存储首地址。

3.4.2 ICAP控制器设计

整个工程中的关键在于 ICAP 控制器的设计，本案例中 ICAP 控制器使用状态机来实现。

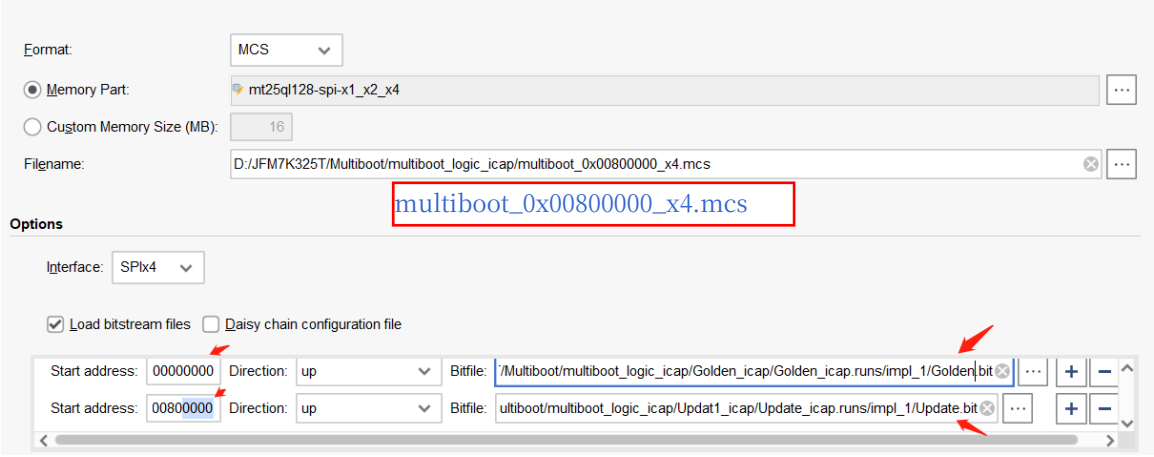


3.4.3 位流生成

位流 Bit 生成：由于 7K523T 器件本身的特性，生产的 bit 流文件大小为 91548896，约为 87Mb，约为 11MB。而 SPIFLASH 的容量为 128 Mb 即 16 MB，这个容量无法满足在一个 SPI FLASH 上贮存两个 bit 流文件，需要对生成的 bit 流进行压缩。在约束 XDC 文件中添加压缩命令即可。

```
43
44 set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
45 set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]
46 set_property BITSTREAM.CONFIG.CONFIGRATE 12 [current_design]
47 set_property CONFIG_MODE SPIx4 [current_design]
48
49 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
50 set_property BITSTREAM.CONFIG.CONFIGFALLBACK ENABLE [current_design]
```

固化 Mcs 生成：Xilinx 系列的 FPGA 需要将后缀名为 mcs 的内存镜像文件固化到外部配置存储器中，FPGA 上电后才能自动加载配置文件。一般的 mcs 文件只包含一个 bit 流文件，多重启动的 mcs 固化文件包含多个 bit 流文件。在将多个 bit 流整合到 mcs 文件的过程中，需要指定每个 bit 流的起始地址，这样 FPGA 专用配置逻辑才能根据地址找到对应的 bit 流。在程序设计 WBSTAR 地址时，确定了 Golden 位流存储的起始地址为 0X00000000，Update 位流存储的起始地址为 0X00800000，因此在将 Bit 整合到 Mcs 过程中需要指定对应的存储起始地址，否则就无法加载成功了。



3.5 板级验证

整个 Multiboot 的功能是 FPGA 上电自动从外部 SPI FLASH 加载一个 Golden 的 bit 流，当需要执行 Update 的程序时，需要外部给一个触发条件，Golden 程序会根据触发条件以及启动地址发起重新配置指令，从而 FPGA 开始重新配置。

在板级验证中，通过拨动 SW11 的最右边的开关，可以满足触发条件实现多重配置。在 FPGA 上电后，自动加载 Golden 程序，LED 灯呈现一盏流水灯，当拨动 SW11 的最右边的开关后，FPGA 加载 Update 程序，LED 灯呈现两盏流水灯，这时再动 SW11 的最右边的开关，FPGA 又加载 Golden 程序，LED 灯呈现一盏流水灯。板级验证的结果符合预期结果，功能正确，Multiboot 跳转正确。