XAPP1100 (v1.0) November 6, 2008

MultiBoot with Virtex-5 FPGAs and Platform Flash XL

Authors: Jameel Hussein and Rish Patel

Summary

The MultiBoot feature on Virtex®-5 FPGAs and Platform Flash XL provides the user with an efficient and high-performance reconfiguration solution. Platform Flash XL offers high-speed data access with customized signals for flawless configuration for Virtex-5 FPGAs. This application note covers the details (both hardware and software) of setting up successful configuration and reconfiguration of Virtex-5 FPGAs from Platform Flash XL. Also included is a reference design consisting of HDL IP that demonstrates how to invoke the MultiBoot feature for Virtex-5 FPGAs. The software commands for creating the MultiBoot PROM file are also described in detail.

Introduction

Virtex-5 FPGAs are in-system reprogrammable, with some applications reloading the FPGA with one or more bitstream images during normal operation. With this feature, a single, smaller FPGA that is reprogrammed to perform multiple functions can replace a much larger and more expensive ASIC or an FPGA programmed to perform *all* functions.

The Virtex-5 FPGA architecture supports MultiBoot, allowing the FPGA to selectively load its bitstream from an attached PROM containing two or more bitstreams. In this mode, the FPGA application triggers a MultiBoot operation, causing the FPGA to reconfigure from a different bitstream. Once a MultiBoot operation is triggered, the FPGA restarts its configuration process as usual, and the FPGA clears its configuration memory and reconfigures from the PROM with the new bitstream.

For this MultiBoot application, Platform Flash XL is used to configure a Virtex-5 FPGA. Platform Flash XL (XCF128X) is a high-speed configuration and storage device that configures the FPGA in synchronous mode, allowing much faster data access than third-party asynchronous BPI flash PROMs. In addition to high-speed data access, the Platform Flash XL has customized signals that ensure fast power-up sequencing. This handshaking between the FPGA and PROM is an exclusive feature of Xilinx configuration devices and significantly reduces power-up conflicts between devices.

MultiBoot Basics

After the FPGA configures itself from the initial bitstream in the Platform Flash XL, the FPGA can trigger a MultiBoot event and reconfigure itself from a different bitstream within the Platform Flash XL (Figure 1). The number of bitstreams (images) supported by the Virtex-5 FPGA MultiBoot feature depends upon the density of the target FPGA, to a maximum of four images.

The MulitBoot feature can be triggered by different methods. A simple and flexible way to trigger the MultiBoot feature is to create a small state machine (internal to the FPGA) to send the internal PROGRAM_B (IPROG) command sequence to the FPGA configuration control via the ICAP primitive (the IPROG command sequence is a set of commands used to trigger MultiBoot — see "IPROG Command Sequence").

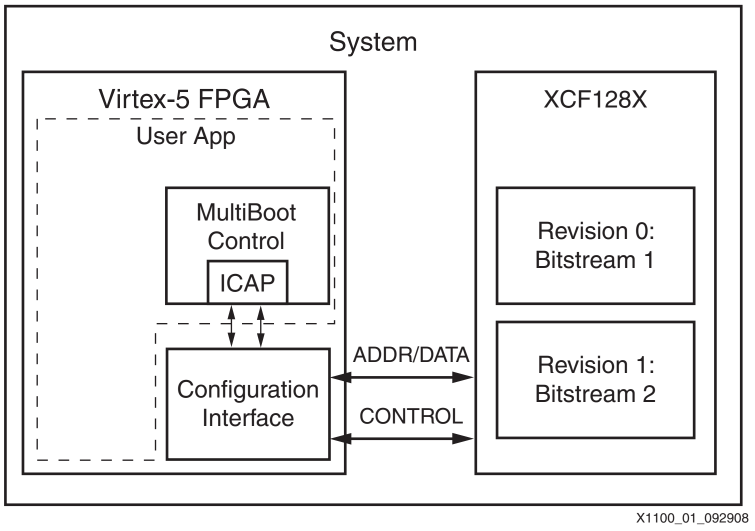


Figure 1: **MultiBoot with XCF128X**

ICAP

The Internal Configuration Access Port (ICAP) primitive provides the user logic access to the Virtex-5 FPGA configuration interface, allowing the user to access configuration registers, readback configuration data, or partially reconfigure the FPGA after configuration is done.

Definition and Pins

The ICAP_VIRTEX5 primitive operates similarly to the SelectMAP configuration interface but differs in three aspects:

- The primitive is located internally to the FPGA.
- The interface has separate read and write buses, as opposed to the bidirectional bus found in SelectMAP.
- Restrictions on readback for the SelectMAP interface do not apply to the ICAP interface after configuration. For example, users can perform readback through the ICAP interface even if bitstream encryption is used.

The general SelectMAP timing diagrams and the SelectMAP bitstream ordering information, described in UG191, *Virtex-5 FPGA Configuration User Guide,* also apply to the ICAP interface.

The ICAP_VIRTEX5 pin names are listed in Table 1.

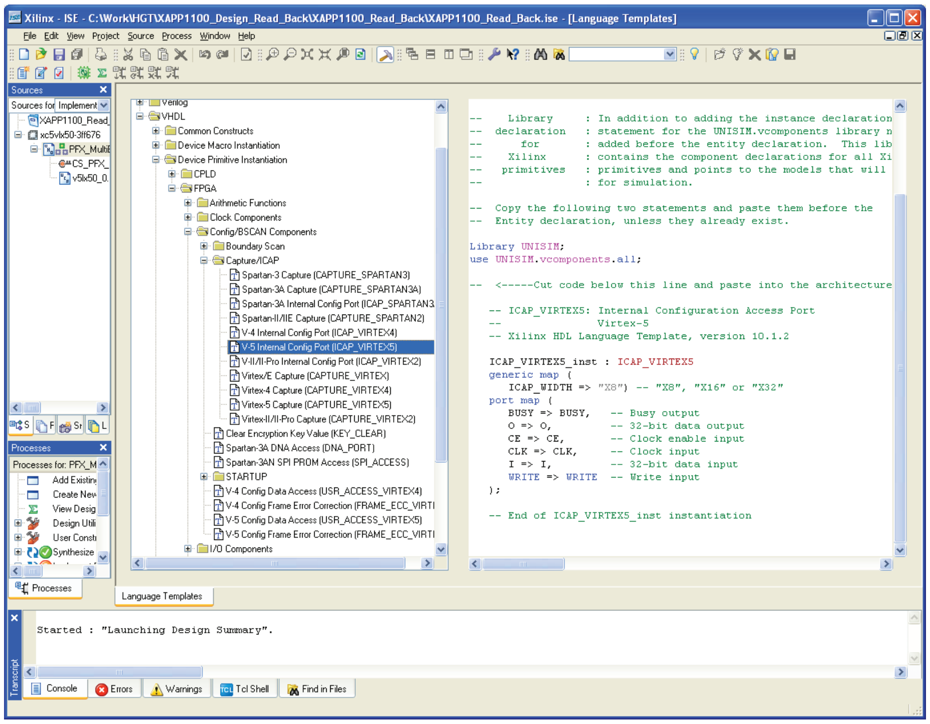
Table 1: **ICAP_VIRTEX5 Pin Table**

Pin Name	Type	Description
CLK	Input	ICAP interface clock
CE	Input	Active-Low ICAP interface select. Equivalent to CS_B in the SelectMAP interface.
WRITE	Input	Equivalent to the RDWR_B signal in the SelectMAP interface: 0=WRITE 1=READ.
I[31:0]	Input	ICAP write data bus. The bus width depends on ICAP_WIDTH attribute. The bit ordering is identical to the SelectMAP interface.
O[31:0]	Output	ICAP read data bus. The bus width depends on the ICAP_WIDTH attribute. The bit ordering is identical to the SelectMAP interface.
BUSY	Output	Active-High busy status. Only used in read operations. BUSY remains Low during writes.

Usage

The user must instantiate the ICAP primitive in the MultiBoot controller before it can be used. The ICAP primitive can be found in ISE® software, under **Edit** → **Language Templates**, **VHDL/Verilog** → **Device Primitive Instantiation** → **FPGA** → **Config/BSCAN -Components** → **Capture/ICAP** → **V-5 Internal Config Port** (Figure 2). The user must specify the operating data width (x8, x16, and x32) of the ICAP via the ICAP_WIDTH attribute (the reference design in this application note is formatted for the x32 mode for 32 bit data loading).

Note: The instructions for instantiating the ICAP primitive in the user design are included in the ICAP template file generated by ISE software.


Xilinx - ISE - C:\Work\HGT\XAPP1100_Design_Read_Back\XAPP1100_Read_Back\XAPP1100_Read_Back.ise - [Language Templates]
File Edit View Project Source Process Window Help
Sources
Sources for: Implement
XAPP1100_Read
xc5vlx50-3ff676
PFX_MultiB
CS_PFX_
v5lx50_0.
Verilog
VHDL
Common Constructs
Device Macro Instantiation
Device Primitive Instantiation
CPLD
FPGA
Arithmetic Functions
Clock Components
Config/BSCAN Components
Boundary Scan
Capture/ICAP
Spartan-3 Capture (CAPTURE_SPARTAN3)
Spartan-3A Capture (CAPTURE_SPARTAN3A)
Spartan-3A Internal Config Port (ICAP_SPARTAN3
Spartan-II/IIE Capture (CAPTURE_SPARTAN2)
V-4 Internal Config Port (ICAP_VIRTEX4)
V-5 Internal Config Port (ICAP_VIRTEX5)
V-II/II-Pro Internal Config Port (ICAP_VIRTEX2)
Virtex/E Capture (CAPTURE_VIRTEX)
Virtex-4 Capture (CAPTURE_VIRTEX4)
Virtex-5 Capture (CAPTURE_VIRTEX5)
Virtex-II/II-Pro Capture (CAPTURE_VIRTEX2)
Clear Encryption Key Value (KEY_CLEAR)
Spartan-3A DNA Access (DNA_PORT)
Spartan-3AN SPI PROM Access (SPI_ACCESS)
STARTUP
V-4 Config Data Access (USR_ACCESS_VIRTEX4)
V-4 Config Frame Error Correction (FRAME_ECC_VIRT
V-5 Config Data Access (USR_ACCESS_VIRTEX5)
V-5 Config Frame Error Correction (FRAME_ECC_VIRT
I/O Components
Processes
Processes for: PFX_M
Add Existing
Create New
View Desig
Design Utili
User Constr
Synthesize
Language Templates
Console
Errors
Warnings
Tcl Shell
Find in Files
Transcript

```
--    Library     : In addition to adding the instance declaration
--  declaration   : statement for the UNISIM.vcomponents library n
--      for       : added before the entity declaration.  This lib
--    Xilinx      : contains the component declarations for all Xi
--   primitives   : primitives and points to the models that will
--                : for simulation.

--  Copy the following two statements and paste them before the
--  Entity declaration, unless they already exist.

Library UNISIM;
use UNISIM.vcomponents.all;

--  <-----Cut code below this line and paste into the architecture

   -- ICAP_VIRTEX5: Internal Configuration Access Port
   --                Virtex-5
   -- Xilinx HDL Language Template, version 10.1.2

   ICAP_VIRTEX5_inst : ICAP_VIRTEX5
   generic map (
      ICAP_WIDTH => "X8") -- "X8", "X16" or "X32"
   port map (
      BUSY => BUSY,   -- Busy output
      O => O,         -- 32-bit data output
      CE => CE,       -- Clock enable input
      CLK => CLK,     -- Clock input
      I => I,         -- 32-bit data input
      WRITE => WRITE  -- Write input
   );

   -- End of ICAP_VIRTEX5_inst instantiation
```

```
Started : "Launching Design Summary".
```

x1100_02_100708

Figure 2: **ICAP Primitive in Language Templates in ISE Software**

IPROG Command Sequence

The effect of the IPROG command is similar to a pulsing PROGRAM_B pin, except IPROG does not reset the dedicated reconfiguration logic. The start address set in the Warm Boot Start Address Register (WBSTAR) is used during reconfiguration instead of the default address. The IPROG command is sent through ICAP_VIRTEX5 to trigger a MultiBoot configuration.

After a successful configuration, the MultiBoot controller determines the start address of the next bitstream, sets the WBSTAR register, and then issues an IPROG command via the ICAP interface.

The sequence of commands are:

1. Send the Sync word.
2. Program the WBSTAR register for the next bitstream start address.
3. Send the IPROG command.

Table 2 shows an example bitstream for sending the IPROG command via ICAP. The MultiBoot controller uses a state machine to send these commands to the ICAP primitive. Typically, the control waits for an external trigger before starting the sequence. After the trigger is received, the controller sets WRITE Low, then sets CE Low on the next rising clock edge, followed by the data sequence listed in Table 2.

Table 2: **Command Sequence for Sending IPROG via ICAP**

Configuration Data (Hex)	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

After the configuration logic receives the IPROG command, the FPGA performs a complete reset (except the dedicated reconfiguration logic) and takes the INIT_B and DONE pins Low. After the FPGA clears all configuration memory, INIT_B is taken High again. Then the value in WBSTAR is used for the bitstream starting address, and reconfiguration begins.

MultiBoot Reference Design

The reference design (MultiBoot_Module) included with this application note is based on the standard hardware setup for Slave SelectMAP configuration for Virtex-5 FPGAs using Platform Flash XL (see "Hardware Layout"). The user application in the FPGA must incorporate a MultiBoot control module to send the necessary commands to trigger a MultiBoot reconfiguration. Finally, a PROM file containing multiple bitstreams in locations addressed by the MultiBoot module is needed.

Module Design

The MultiBoot module includes both a state machine and the ICAP primitive. The user application signals a MultiBoot event to the module via a trigger (Figure 3). The state machine handles the required SelectMAP data ordering, passing the configuration data to the FPGA configuration engine via the ICAP primitive. The configuration data sent follows the required SelectMAP data loading order. The flow for the state machine is shown in Figure 4.

In addition to a trigger, the module also requires a clock signal (CLK). This clock input should follow the SelectMAP clock speed limitations (1–100MHz).

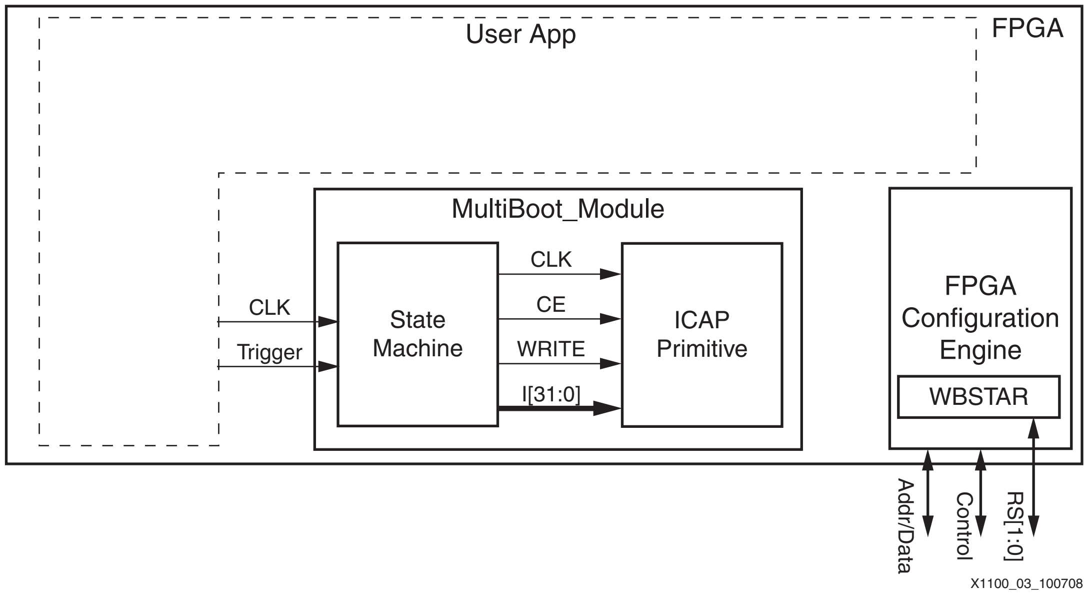


Figure 3: **Reference Design Block Diagram**

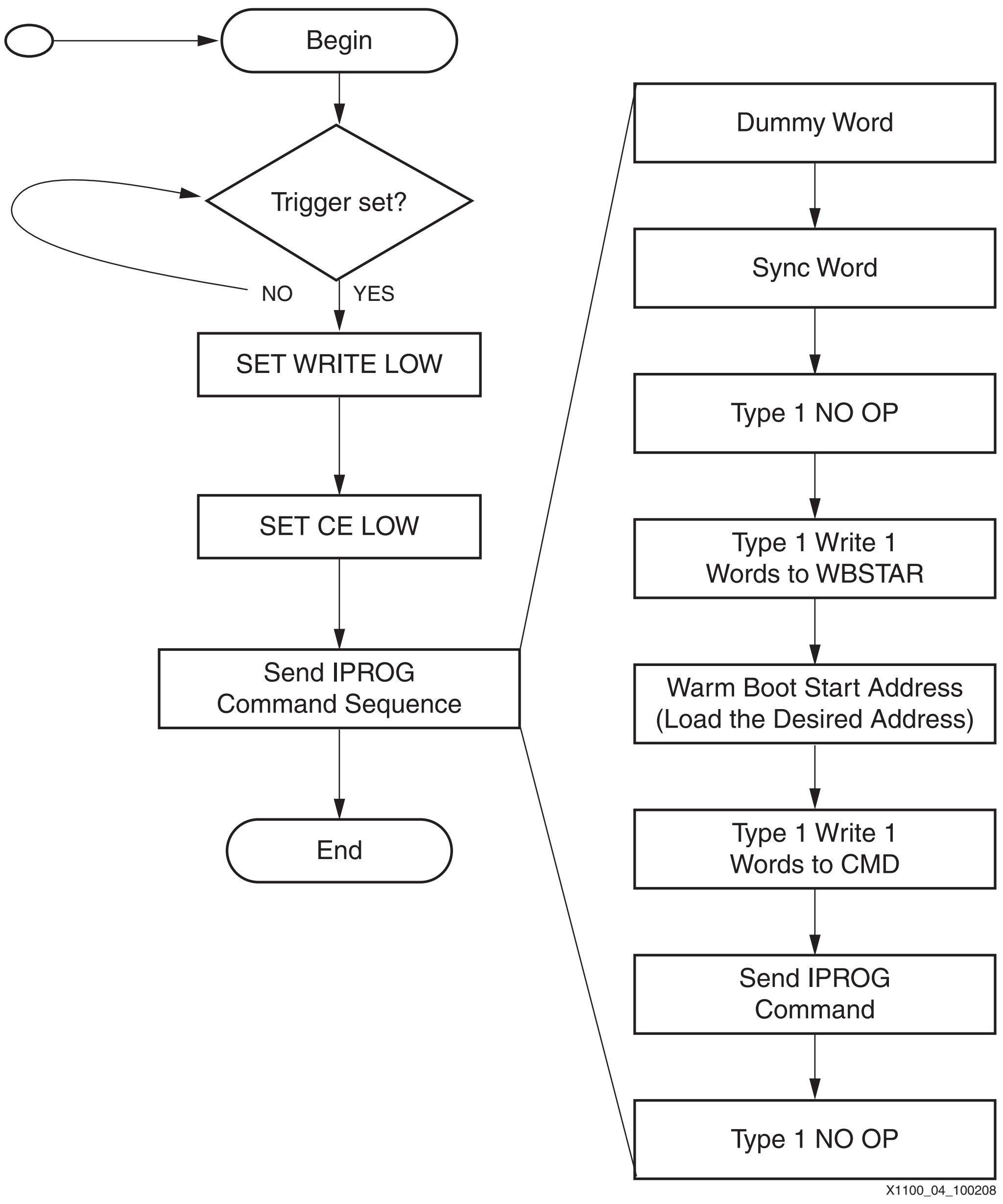

Begin
Trigger set?
NO
YES
SET WRITE LOW
SET CE LOW
Send IPROG Command Sequence
End
Dummy Word
Sync Word
Type 1 NO OP
Type 1 Write 1 Words to WBSTAR
Warm Boot Start Address (Load the Desired Address)
Type 1 Write 1 Words to CMD
Send IPROG Command
Type 1 NO OP
X1100_04_100208

Figure 4: **State Machine Flow Diagram**

ICAP Instantiation

The ICAP primitive used in the reference design is slightly modified from the sample in the Language Templates section in the ISE software. The reference design only uses the I, CE, WRITE, and CLK signals of the ICAP primitive. The remaining signals, BUSY and O, are omitted from the ICAP instantiation because the module does not need to read any of data passing through the ICAP interface. These signals are used when reading data from the ICAP primitive.

In addition, the reference design requires that the ICAP_WIDTH attribute for the primitive be set to X32, enabling the design to send 32-bit data to the primitive (both I and O are set to 32 bits).

SelectMAP Data Ordering

To ensure that the ICAP primitive receives the command sequence properly, commands and data must follow the correct bit ordering. The bit or data ordering scheme for the Virtex-5 family differs from that of the Virtex-4 family — data must be bit swapped on a byte basis. For each group of 8 data pins, the bits must be ordered so that the MSB is at D0 and the LSB is at D7. This same scheme needs to be followed for each byte of x16 and x32 data (Figure 5).

The state machine in the reference design is constructed to reformat the data as required.

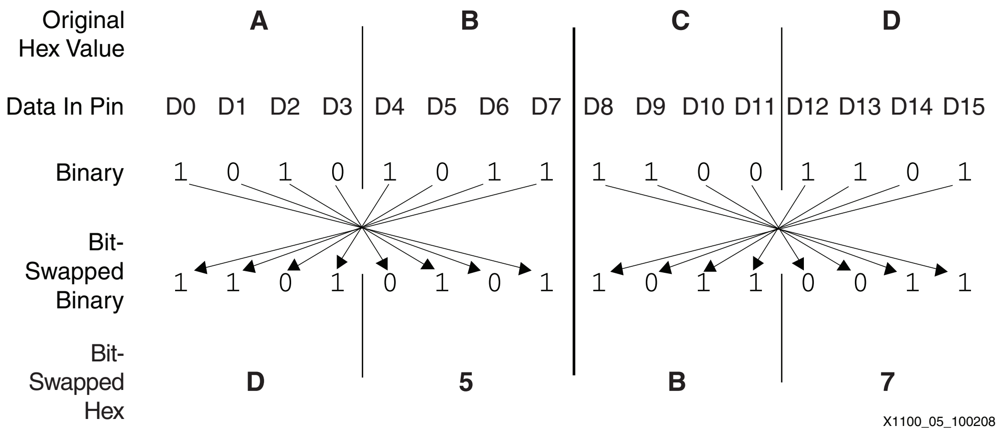


Figure 5: **x16 Data Ordering Example**

SelectMAP Data Loading

Because the ICAP primitive behaves like an external SelectMAP interface, the state machine in the MultiBoot module must follow the same data loading protocol (Figure 6). Any alterations to the provided code must follow the same protocol for sending data.

Before the IPROG command sequence is sent to the device, WRITE (RDWR_B for SelectMAP) must be at a valid Low before CE (CS_B for SelectMAP) goes Low. In the reference design, these transitions occur on separate clock edges to eliminate contention.

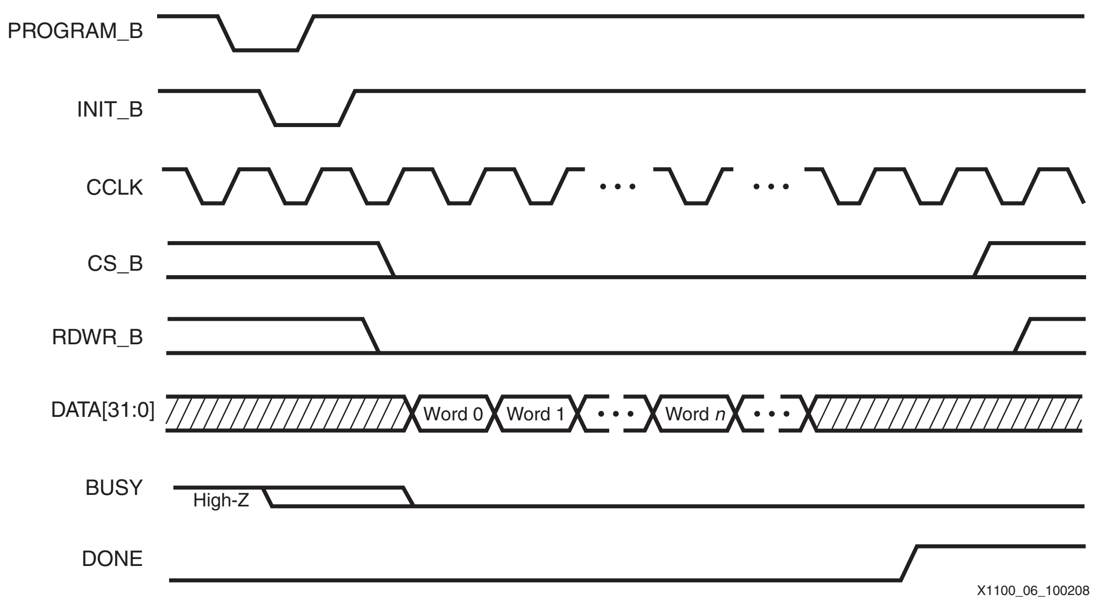


Figure 6: **Continuous x32 SelectMAP Data Loading**

IPROG State Machine

After a trigger is received from the user application, the state machine sends the IPROG command sequence to the ICAP primitive, following SelectMAP data ordering and loading protocols.

Table 3 shows the command sequence *after* the bits have been swapped for 32-bit data loading. This sequence is sent to the ICAP in the S_ICAP state of the reference design.

Table 3: **Bit-Swapped Command Sequence for Sending IPROG via ICAP**

Configuration Data (Hex)	I[31:0] (Hex)	Explanation
`FFFFFFFF`	`FFFFFFFF`	Dummy Word
`AA995566`	`5599AA66`	Sync Word
`20000000`	`04000000`	Type 1 NO OP
`30020001`	`0C400080`	Type 1 Write 1 Words to WBST AR
`00000000`	`00000000`	Warm Boot Start Address (Load the Desired Address)
`30008001`	`0C000180`	Type 1 Write 1 Words to CMD
`0000000F`	`000000F0`	IPROG Command
`20000000`	`04000000`	Type 1 NO OP

Warm Boot Start Address Register (WBSTAR)

One of the configuration registers, the Warm Boot Start Address Register, shown in Table 4 and described in Table 5, is used to store the starting address of the bitstream in the external flash configuration device. One of the commands in the IPROG command sequence is to set the WBSTAR register with the address desired for reconfiguration. The address chosen for the WBSTAR must align with the starting address of the bitstream programmed in the external configuration device. Setting this register correctly is important for successful reconfiguration. For more details about the WBSTAR register refer to the *Virtex-5 FPGA Configuration User Guide.*

Table 4: **WBSTAR Register**

Description	Reserved			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: **WBSTAR Register Description**

Name	Bit Index	Description
RS[1:0]	[28:27]	RS[1:0] pin value on next warm boot
RS_TS_B	26	RS[1:0] pins 3-state enable. `0`: Disabled `1`: Enabled
START_ADDR	[25:0]	Fallback bitstream start address

The reference design takes advantage of the fact that during SelectMap mode, WBSTAR drives the values set in bits 27 and 28 to the external FPGA pins RS[1:0]. These pins are connected to high-order address pins (A[22:21]) of the external Platform Flash XL configuration device. These bits are used to divide the 128 Mb flash device into four equal revisions.

Note: The RS_TS_B bit must also be set to `1` to enable the RS pins.

Setting MultiBoot Address

The reference design is targeted to a Virtex-5 FXT FPGA, which has a bitstream size of roughly 14 Mbs, requiring the start address of the next bitstream be at least 14 Mbs into the memory space of the configuration device. In addition, there must be sufficient buffer space between bitstreams in the configuration device because the Virtex-5 FPGA keeps loading the bitstream until the DONE pin goes High. When DCI match and DCM lock wait is enabled before the DONE cycle, padding (all `0`s or all `1`s) is required between bitstreams to compensate for the total match or lock time. Otherwise, the following bitstream can potentially overwrite the previous one. The DCI match time is typically less than 1 ms (for the DCM lock time, refer to the DCM section in DS202, *Virtex-5 Data Sheet*.

The formula for the padding size is:

$$\frac{\text{cfg\_bus\_width} \times \text{total\_lock\_time}}{\text{CCLK\_period}} \qquad \textit{Equation 1}$$

In the reference design, the start address for the second bitstream used for MultiBoot reconfiguration is set to revision 1, which translates to `0x0C000000` in the WBSTAR register. RS[1:0] is set to `01`, allocating 32 Mb of space for the bitstream.

Note: The actual value in the code is different. The value that appears is `0x03000000` — the result of the required SelectMAP bit swapping (see "SelectMAP Data Ordering").

The reference design is set up to reconfigure from revision 1. If the user wants to use more than two revisions, the RS pins must be set accordingly, requiring adjustment to the WBSTAR register (see Table 6 for the appropriate addresses). To reconfigure from a different revision, the state machine in the reference design must be modified.

Table 6 lists the correct setting for WBSTAR for a four-revision application.

Table 6: **Address Description**

Multiboot to Revision	WBSTAR Register Description			WBSTAR Address	PROMGen Address	WBSTAR Address (after data ordering)
	RS[1:0]	RS_TS_B	START_ADDR			
Rev 0	0x00	1	0x0	0x04000000	0x000000	0x20000000
Rev 1	0x01	1	0x0	0x0C000000	0x200000	0x30000000
Rev 2	0x10	1	0x0	0x14000000	0x400000	0x28000000
Rev 3	0x11	1	0x0	0x1C000000	0x600000	0x38000000

Hardware Layout

The hardware layout for this reference design is based on the hardware setup for Slave SelectMAP configuration for Virtex-5 FPGAs using Platform Flash XL. The only modification is to connect the RS pins on the FPGA to the high-order address pins A[22:21] between the Platform Flash XL and the FPGA. This connection allows for iMPACT software indirect programming via the FPGA's high-order address pins and for revision selection via the RS pins during a MultiBoot operation.

Refer to DS617, *Platform Flash XL High-Density Storage and Configuration Device*, for the full specification of the hardware setup for configuration in Slave SelectMAP mode.

A schematic for the Virtex-5 FPGA slave SelectMap configuration mode with Platform Flash XL is shown in Figure 7. Refer to Table 7 for more detail on the behavior of the RS pin and the address pins during configuration and iMPACT indirect programming.

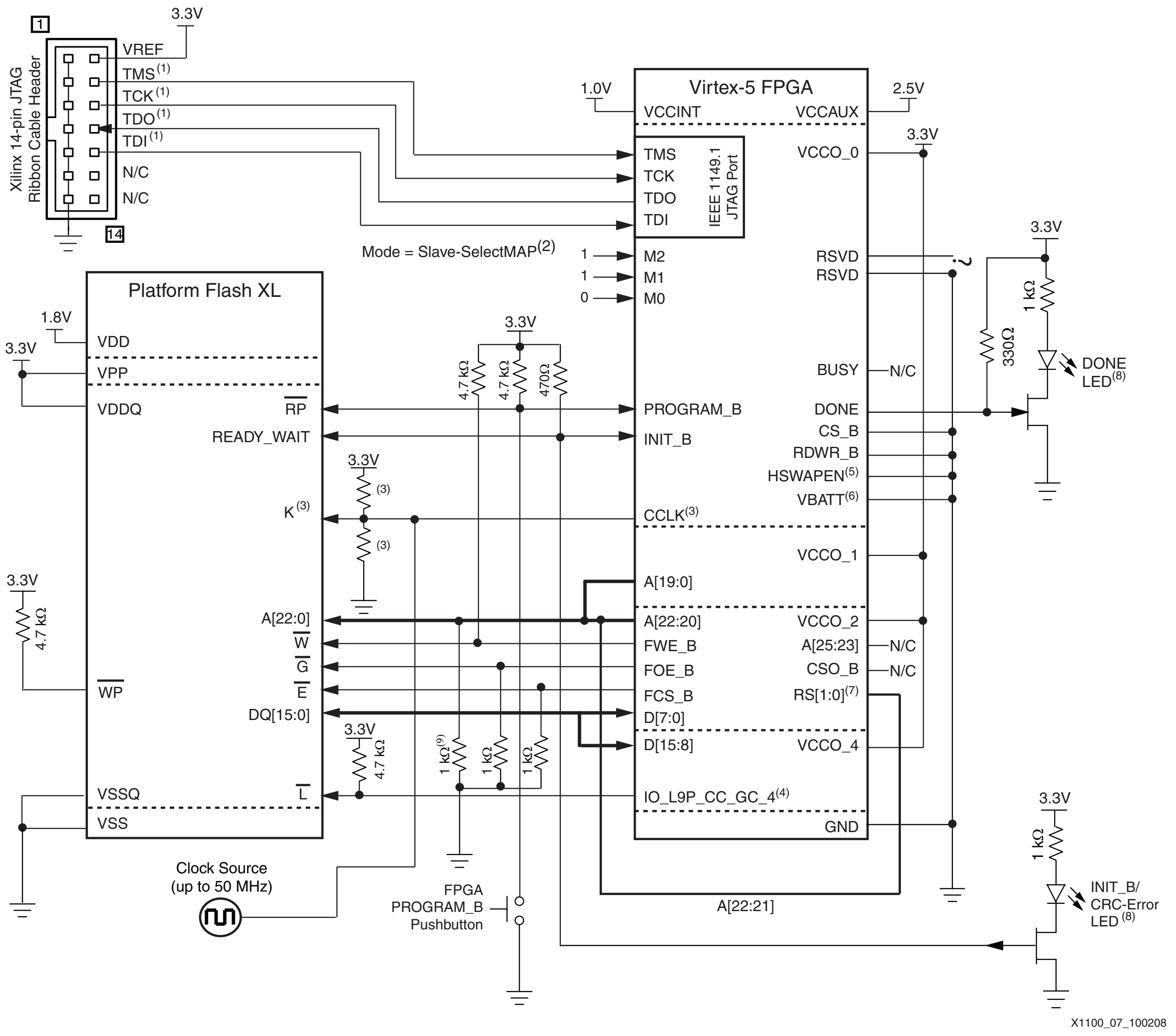


Figure 7: **Virtex-5 FPGA Slave SelectMap Configuration Mode with Platform Flash XL**

Notes pertaining to Figure 7:

1. The JTAG connections are shown for a simple, single-device JTAG scan chain. When multiple devices are on the JTAG scan chain, the proper IEEE Std 1149.1 daisy-chain technique to connect the JTAG signals must be used. The TCK signal integrity is critical for JTAG operation. The TCK signal must be appropriately routed, terminated, and if necessary, buffered to ensure signal integrity for the devices in the JTAG scan chain.
2. The FPGA mode (M[2:0]) pins are shown set to Slave SelectMAP mode (`110`). The implementation of a board-level option that enables the user to change the FPGA mode pins to JTAG mode (`101`) is strongly recommended to enable full JTAG-based debug capability for the FPGA during design prototyping.

3. CCLK signal integrity is critical. the CCLK signal must be appropriately routed and terminated to ensure good signal integrity at the XCF128X K pin and at the FPGA CCLK pin.
4. The iMPACT software requires the Virtex-5 FPGA's IO_L9P_CC_GC_4 connection to the device's L pin to support JTAG-based indirect programming.
5. The FPGA HSWAPEN pin is tied to ground in this sample schematic. HSWAPEN can alternatively be tied High. Review the FPGA data sheet for the effect of the alternate HSWAPEN setting.
6. The Virtex-5 FPGA does not support AES decryption in the 16-bit-wide configuration mode shown in this sample schematic. Thus, the VBATT decryptor key battery power supply is unused and is tied to GND.
7. The RS[1:0] pins are connected to Platform Flash XL high-order address pins for MultiBoot reconfiguration.
8. DONE LED lights when DONE is High. INIT_B/CRC-Error LED lights when INIT_B is Low. The LED circuits and pull-up values must be adjusted for desired lighting results.
9. Each Platform Flash XL address pin requires a separate pull-down resistor to GND to ensure the XCF128X flash latches the zero address at the start of configuration.

Table 7: **Platform Flash XL and Virtex-5 FPGA Address Signals and Descriptions**

Platform Flash XL		**Virtex-5 FPGA**			
Pin Name	**Signal Description**	**Pin Name**	**Signal Description**	**Function**	
				During SelectMap Configuration	**During iMPACT Indirect Flash Programing**
A[20:0]	Address Inputs	ADDR[20:0]	Address Output	A[20:0] pins are 3-stated.	Address outputs.
A[21]	High Order Address Pins	A[21] and RS[0]	Revision select pins. Used for multiple bitstream applications.	RS[1:0] pins are 3-stated during the initial configuration. RS[1:0] pins are driven by the user through ICAP during MultiBoot configuration. A[22:21] pins are 3-stated	RS[1:0] pins are 3-stated. A[22:21] pins are address outputs.
A[22]		A[22] and RS[1]			

Creating a PROM file for MultiBoot Applications

This section details the flow to create a PROM file for MultiBoot applications using the Xilinx ISE software (specifically, PROMGen) in Version 10.1.03. PROMGen is the software program used to format single or multiple BIT files into a PROM file and can be run as a standalone program from the UNIX/DOS command prompt. For more information on the Command-Line mode usage and options, refer to the PROMGen Chapter in the *Xilinx Development System Reference Guide*.

Preparing a MultiBoot PROM File Using the ISE PROMGen Command-Line Software

PROMGen takes multiple FPGA bitstream (`.bit`) files as input and, with the appropriate options, generates a memory image file for the data array for a Platform Flash XL. The output memory image file format is chosen via a PROMGen command-line option. Typical file formats include Intel Hex (`.mcs`) and Motorola Hex (`.exo`).

Note: The Platform Flash XL uses the Intel Hex (`.mcs`) format.

The PROMGen command-line to generate an mcs-formatted file with multiple bitstreams for a 16-MB (or 128-Mb) Platform Flash XL used in BPI-UP mode is:

```
promgen -p mcs -o MulitBoot_File.mcs -s 16384 -data_width 16 -u 0 rev0.bit
-u 200000 rev1.bit
```

The **-p** mcs option specifies Intel Hex (`.mcs`) output file format. The **-o MultiBoot.mcs** specifies output to a file named `MultiBoot.mcs`. The **-S 16384** specifies a PROM file page size in kilobytes. The **-u 0** option specifies the data for first bitstream to start at address zero and fill the data array in the up direction. The `rev0.bit` is the first bitstream file, and `rev1.bit` the second. The **-u 200000** option specifies the data for second bitstream to start at address `0x0200000` and fill the data array in the up direction (see Table 6).

Reference Design Files

The reference design files are available for download at:
https://secure.xilinx.com/webreg/clickthrough.do?cid=112420

The reference design matrix, which provides additional information about the design, is shown in Table 8.

Table 8: **Reference Design Matrix**

Parameter	**Description**
Developer Name	Xilinx
Target Devices (stepping level, ES, production, speed grades)	Virtex-5 FPGA
Source Code Provided	Yes
Source Code Format	Verilog and VHDL
Design Uses Code/IP from an Existing Reference Design/Application Note, Third Party, or CORE Generator™ software	No
Simulation	
Functional Simulation Performed	Yes
Timing Simulation Performed	No
Testbench Used for Functional Simulations Provided	Yes
Testbench Format	VHDL
Simulator Software Used/Version (e.g., ISE software, Mentor, Cadence, other)	ModelSim SE
SPICE/IBIS Simulations	No
Implementation	
Synthesis Software Tools Used/Version	XST
Implementation Software Tools Used/Versions	ISE software version 10.1, Service Pack 3
Static Timing Analysis Performed	Yes
Hardware Verification	
Hardware Verified	Yes

Conclusion

This application note enables the user to use the MultiBoot feature in Virtex-5 FPGAs to reconfigure the FPGA with a different bitstream stored in the Platform Flash XL.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
11/06/08	1.0	Initial Xilinx release.

Notice of Disclaimer