

Programming ProASIC^{PLUS} Devices in a Mixed Chain

Table of Contents

Introduction	1
Overview	2
Editing a STAPL File	3
CRC Check	3
STAPL File Example	3
Using the FlashPro Software	5
Device.db File Description	6
FlashPro Software Example	7
List of Changes	10

Introduction

The ProASIC^{PLUS}® family of devices, Microsemi second generation Flash field programmable gate array (FPGA), offers enhanced performance over Microsemi ProASIC® family of devices. ProASIC^{PLUS} devices combine the advantages of ASICs with the benefits of programmable devices through non-volatile Flash technology. This combination enables engineers to create high-density systems using existing ASIC or FPGA design flow and tools.

This application note describes the steps to program a ProASIC^{PLUS} device in a mixed chain using a STAPL player or Microsemi FlashPro programmer. A mixed chain of devices refers to a chain of devices from various vendors connected together serially through the JTAG port, as shown in [Figure 1](#). TD0 of the programming header represents the beginning of the chain. The ProASIC^{PLUS} device supports daisy chain programming through the IEEE 1149.1 (JTAG) port.

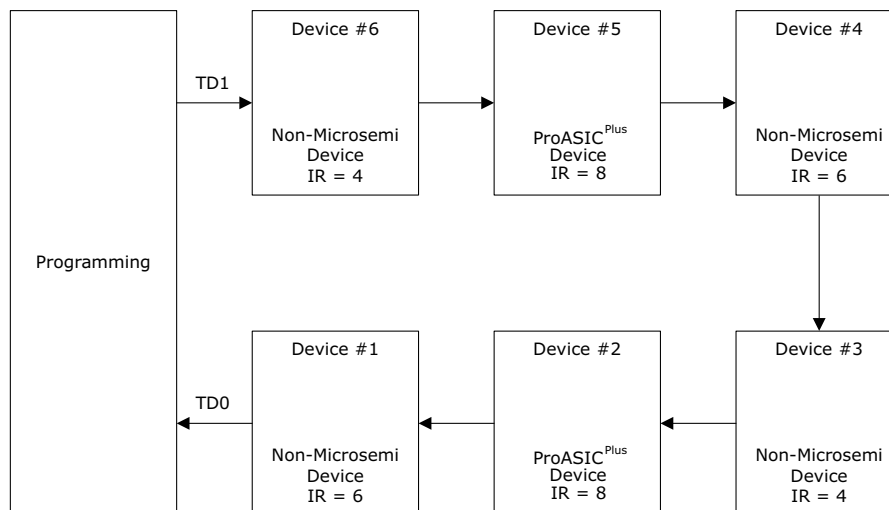


Figure 1 • Programming a Mixed Chain of Devices

Overview

To program a device within a chain of devices, first put the other devices in the chain into bypass mode. Once a device is in bypass mode, its data register length becomes one and does not react to any programming instructions given by the programmer. To put a device into bypass mode, the programmer must know its instruction register length.

When using a standard STAPL player, the STAPL file generated from Microsemi Designer software, must be edited to include the bypass information. When using Microsemi FlashPro programmer, a device data text file in the FlashPro software must be edited to include the bypass information. [Table 1](#) shows product support, and [Figure 2](#) shows the programming flow for the standard STAPL Player and FlashPro.

All the necessary information for editing the STAPL file or FlashPro data file is located in the BSDL file ([Figure 3 on page 3](#)). The instruction length is shown under the INSTRUCTION_LENGTH attribute and the device ID is shown under the IDCODE_REGISTER attribute.

Table 1 • Product Support Information

Programming Platform	Daisy Chain Support through Edited STAPL File	Daisy Chain Support through Device.db File
FlashPro	No	Yes
FlashPro Lite	No	Yes
STAPL/JAM Player	Yes	No
Sculptor II (using STAPL)	Yes	No
Sculptor I and II (using Bitstream)	No	No

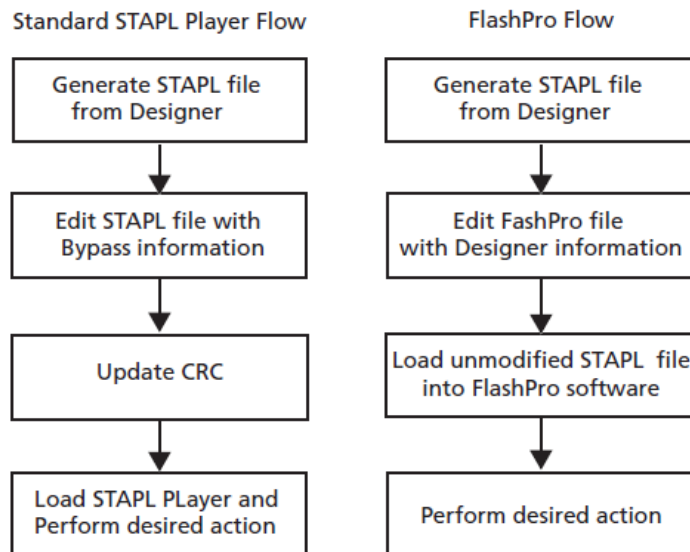


Figure 2 • STAPL Player and FlashPro Programming Flow

Editing a STAPL File

In the INITIALIZE procedure, add the following instructions to the STAPL file, so it can bypass the other devices in the chain.

PREIR: The sum of the instruction register lengths of all the devices in front of the target device.

POSTIR: The sum of the instruction register lengths of all the devices after the target device.

PREDR: The number of devices in front of the target device.

POSTDR: The number of devices after the target device.

The PREIR and POSTIR instructions give the information to the programmer, to put all other devices within the chain into bypass mode. The PREDR and POSTDR instructions give the programmer the proper number of bits to add to the data that is being shifted into the chain. After the devices enter bypass mode, the Data Register becomes 1-bit wide.

CRC Check

After the STAPL file is edited, the CRC of the file changes. Any STAPL compliant player will reject the edited STAPL file, citing a CRC error. There are two methods to resolve this issue. One method is to set the CRC value of the edited STAPL file to 0000. This action disables the CRC check of the STAPL player (per the STAPL spec). The other method is to load the edited STAPL file into Microsemi FlashPro player (v3.1 or newer). The FlashPro player will return a CRC error, but will also recalculate the CRC value and display a new CRC value. Replace the CRC value in the STAPL file with the new recalculated CRC value.

```

attribute INSTRUCTION_LENGTH of A500K050BG272 : entity is 8;
attribute INSTRUCTION_OPCODE of A500K050BG272 : entity is
  "BYPASS      (11111111), " &
  "IDCODE     (00001111), " &
  "EXTTEST    (00000000), " &
  "SAMPLE     (00000001)";
attribute INSTRUCTION_CAPTURE of A500K050BG272 : entity is "XXXXXX01" ;

attribute IDCODE_REGISTER of A500K050BG272 : entity is
  "XXXX00000000110100111000111001111"
  
```

IR Length

Device ID

Figure 3 • BSDL File for an Microsemi ProASIC Device

STAPL File Example

Table 2 lists a STAPL file example of a mixed chain of devices that are connected together (Figure 1 on page 1) for programming Device #5 (APA device). Table 2 lists the values for PREIR, POSTIR, PREDR, and ROSTDR using the variable instructions to calculate the values. The values from Table 2 are then added to the STAPL file (Figure 4 on page 4). After the STAPL file is edited with the correct values, load it into the FlashPro software to determine the new CRC value (Figure 5 on page 4). Then edit the STAPL file with the new CRC value (Figure 7 on page 6).

Table 2 • STAPL File Example Values

Variable	Value	Formula
PREIR	24	Σ IR (Dev1, Dev2, Dev3, Dev4)
POSTIR	4	Σ IR (Dev6)
PREDR	4	Total of four devices in front of device 5
POSTDR	1	Single device after device 5

```

PROCEDURE INITIALIZE USES GV, PARAMETERS, GP, POWER_DOWN;
PREIR 24 ;
POSTIR 4 ;
PREDR 4 ;
POSTDR 1;
IF !USE_RCK THEN FREQUENCY freq*1000000 ;
WAIT RESET, 5 CYCLES;
IRSCAN 8, $0f;
DRSCAN 32, $000000FF, COMPARE $014101cf, $0BFFFFFF, PASS;
IF PASS==1 THEN GOTO idok;
STATUS=6;
CALL POWER_DOWN;
idok:
IRSCAN 8, $09
IRSCAN 8, $0a
IRSCAN 8, $92;
IF USE_RCK THEN DRSCAN 8, B00L (freq-1);
IF !USE_RCK THEN DRSCAN 8, B00L (128+freq-1);
CALL GP;
IRSCAN 8, $d3;
IRSCAN 8, $d8;
IRSCAN 8, $e5;
IRSCAN 8, $c8;
DRSCAN I, #0, CAPTURE PS{};
ENDPROC;

```

Note: The PREIR value is 24; the POSTIR and PREDIR values are four, and the POSTDR value is one.

Figure 4 • New Values Added to the STAPL File

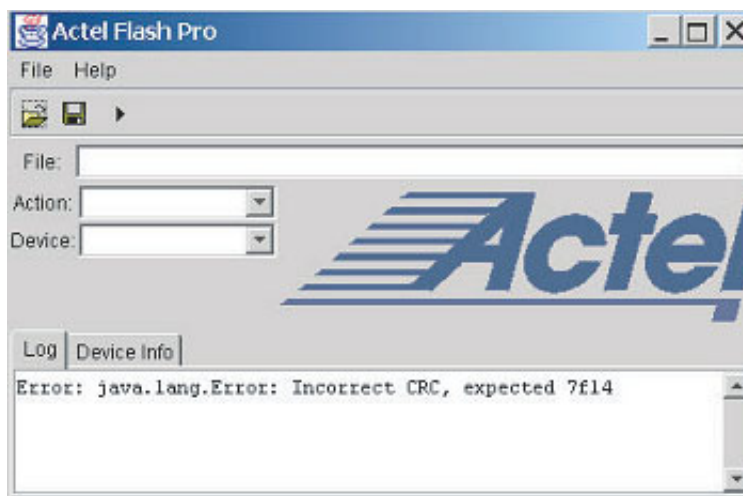


Figure 5 • CRC Value from the FlashPro Software

Once the STAPL file is edited, it can be used on most STAPL players except FlashPro.

Using the FlashPro Software

Because of the Analyze Chain feature in the FlashPro software, an edited STAPL file cannot be used with the FlashPro software to program a ProASIC^{PLUS} device within a mixed chain. The FlashPro software has its own database file (`Device.db`) that contains all the information to bypass a non-Microsemi device. Any new non-Microsemi device that a user requires support for must have the necessary information entered into this `Device.db` file.

The `Device.db` file contains the following information for each device:

- Device ID
- Device name
- Instruction register length

```
PROCEDURE POWER_DOWN USES GV;  
IRSCAN 8 $0b  
WAIT 5 USEC;  
STATE RESET;  
EXIT STATUS;  
ENDPROC;
```

```
CRC 7f14; ← Change CRC to a new value (last line  
of the STAPL file).
```

Figure 6 • Edited STAPL File with New CRC Value

```

<BANK NUMBER = "1">
  <ENTRY ID="47" COMPANY= "ACTEL ">
    <DEVICE PART= "20BD" NAME = "SX32P" IR = "5" />

    <DEVICE PART= "22BD" NAME = "SX32P" IR = "5" />
    <DEVICE PART= "40B2" NAME = "SX72A" IR = "5" />
    <DEVICE PART= "40B4" NAME = "SX08A" IR = "5" />
    <DEVICE PART= "40B8" NAME = "SX16A" IR = "5" />
    <DEVICE PART= "40BD" NAME = "SX32A" IR = "5" />

  </ENTRY>
  <ENTRY ID = "73" COMPANY= "Xilinx" />
  <ENTRY ID= "110" COMPANY= "ALTERA" />

</BANK>

<BANK NUMBER = "2">
  <ENTRY ID = "103" COMPANY= "ACTEL" >
    <DEVICE PART = "01A7" NAME = "A500K050" IR = "8" />
    <DEVICE PART = "02CA" NAME= "A500K130" IR= "8" />

    <DEVICE PART = "034C" NAME= "A500K180" IR= "8" />
    <DEVICE PART = "03CF" NAME= "A500K270" IR= "8" />
    <DEVICE PART = "0C10" NAME= "A61PA750" IR= "8" />
    <DEVICE PART = "19C4" NAME= "APA075" IR= "8" />
    <DEVICE PART = "1A46" NAME= "APA150" IR= "8" />
    <DEVICE PART = "1A08" NAME= "APA300" IR= "8" />
    <DEVICE PART = "1B08" NAME= "APA450" IR= "8" />
    <DEVICE PART = "138C" NAME= "APA600" IR= "8" />
    <DEVICE PART = "1410" NAME= "APA750" IR= "8" />
    <DEVICE PART = "1594" NAME= "APA1000" IR= "8" />

  </ENTRY>

</BANK>

<BANK NUMBER = "3">
</BANK>

```

Figure 7 • Device.db. File Example

Device.db File Description

The `Device.db` file includes the device ID, device name, and the instruction register length.

Device ID

A device ID is 32 bits wide and divided into five sections:

- **LSB (0-bit):** The first bit in the Device ID is always a one.
- **Manufacturing ID (1-7 bits):** The manufacturing ID is assigned by the JEDEC office. Within the `device.db` file, the manufacturing ID is referred as the Entry ID and it is represented in decimal form.
- **Bank Value (8-11 bits):** The bank value allows multiple vendors to have the same value assigned to the manufacturing ID. Bits 0000 represent bank one.
- **Part Number (12-27 bits):** The part number is assigned by the manufacturer. The part number is represented in Hex.
- **Version Number (28-31 bits):** The version number is assigned by the manufacturer

A complete listing of all bank value and manufacturing IDs can be found at:
<http://www.jedec.org/download/search/jep106L.pdf>

Device Name

The device name is the name that will appear after the FlashPro completes the Analyze Chain function.

Instruction Register Length

The instruction register length is the length of the instruction register for a particular device. When the FlashPro programmer performs the analyze chain function, it will compare the device IDs read back from the chain to those stored in the `device.db` file to obtain the devices instruction register length.

FlashPro Software Example

FlashPro software example using the chain in [Figure 1 on page 1](#) with more information assigned to each device are as follows:

Device #6 is a Cisco device with the following:

- ID code: 0000,0010001010110001,0011,0001010,1
- Bank number: 3
- Part number: 22B1
- Entry ID (manufacturing ID): 10
- Device Name: Router_IC

Device #4 is a TI device with the following information:

- ID code: 0000,0011001010110101,0000,0010111,1
- Bank number: 1
- Part number: 32B5
- Entry ID (manufacturing ID): 23
- Device Name: DSP_IC

Device #3 is an Atmel device with the following information:

- ID code: 0000,1010011010101101,0000,0011111,1
- Bank number: 1
- Part number: A6AD
- Entry ID (manufacturing ID): 31
- Device Name: EPROM

Device #1 is a Xilinx device with the following information:

- ID code: 0000,0110010010100101,0000,10010001,1
- Bank number: 1
- Part number: 64A5
- Entry ID (manufacturing ID): 73
- Device Name: Spartan

A description of the ID code is represented in the "Device ID" section, and the ID code of Device #6 is listed in [Table 3](#).

Table 3 • ID Code Description

Bit(s)	Description
0000	Version number
0010001010110001	Part number
0011	Bank value
0001010	Manufacturing ID
1	Always one

The Bank number, part number, and Entry ID are calculated using the descriptions from [Table 2 on page 3](#). Then, open the `device.db` file using any text editor. Enter the information of the four non-Microsemi devices into the file (see [Figure 8](#)).

```

<BANK NUMBER = "1">
  <ENTRY ID="47" COMPANY= "ACTEL ">
    <DEVICE PART= "20BD" NAME = "SX32P" IR = "5" />

    <DEVICE PART= "22BD" NAME = "SX32P" IR = "5" />
    <DEVICE PART= "40B2" NAME = "SX72A" IR = "5" />
    <DEVICE PART= "40B4" NAME = "SX08A" IR = "5" />
    <DEVICE PART= "40B8" NAME = "SX16A" IR = "5" />
    <DEVICE PART= "40BD" NAME = "SX32A" IR = "5" />

  </ENTRY>
  <ENTRY ID = "73" COMPANY= "Xilinx" />
  Device # 1 → <DEVICE PART = "65A5" NAME = Spartan" ir= "6" />

  </ENTRY>
  <ENTRY ID = "110" COMPANY= "ALTERA" />
  Device # 4 → <ENTRY ID= "23" COMPANY= "TI" />
  <DEVICE PART= "32B5" NAME = "DSP_IC" IR= " 6 " />
  </ENTRY>
  Device # 3 → <ENTRY ID= "31" COMPANY "ATMEL" />
  <DEVICE PART= "A6AD" NAME ="EPROM" IR ="4" />
  </ENTRY>

</BANK>

<BANK NUMBER = "2">
  <ENTRY ID = "103" COMPANY= "ACTEL" >
    <DEVICE PART = "01A7" NAME = "A500K050" IR = "8" />
    <DEVICE PART = "02CA" NAME= "A500K130" IR= "8" />

    <DEVICE PART = "034C" NAME= "A500K180" IR= "8" />
    <DEVICE PART = "03CF" NAME= "A500K270" IR= "8" />
    <DEVICE PART = "0C10" NAME= "A61PA750" IR= "8" />
    <DEVICE PART = "19C4" NAME= "APA075" IR= "8" />
    <DEVICE PART = "1A46" NAME= "APA150" IR= "8" />
    <DEVICE PART = "1A08" NAME= "APA300" IR= "8" />
    <DEVICE PART = "1B08" NAME= "APA450" IR= "8" />
    <DEVICE PART = "138C" NAME= "APA600" IR= "8" />
    <DEVICE PART = "1410" NAME= "APA750" IR= "8" />
    <DEVICE PART = "1594" NAME= "APA1000" IR= "8" />

  </ENTRY>

</BANK>

<BANK NUMBER ="3">
  Device # 6 → <ENTRY ID= "10" COMPANY= Cisco" >
    <DEVICE PART= "22b1" NAME= "ROUTER_IC" IR="4" />
  </ENTRY>

</BANK>

<BANK NUMBER = "4">
</BANK>

```

Figure 8 • Programming Mixed Chain Device Example

List of Changes

The following table shows important changes made in this document for each revision.

Revision	Changes	Page
Revision 1 (June 2016)	Non-technical updates.	N/A
Revision 0 (August 2003)	Initial release.	N/A



Power Matters.™

Microsemi Corporate Headquarters

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

www.microsemi.com

© 2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.