Summary

This application note describes the 7 series FPGA Byte-wide Peripheral Interface (BPI) configuration mode with synchronous read and the External Master Configuration Clock (EMCCLK) options. Because the BPI configuration mode with synchronous read and EMCCLK provides the fastest configuration time from a direct FPGA interface, it is referred to as BPI Fast Configuration in this document. The BPI Fast Configuration enables high-capacity nonvolatile storage and decreases the configuration time to less than 8% of the legacy BPI configuration with the asynchronous read. The 7 Series FPGAs Configuration User Guide (UG470) [Ref 1] contains details on FPGA configuration. Familiarity with this document is assumed.

The hardware setup, file generation, and the tool flow to indirectly program a parallel NOR flash through the 7 series FPGA are described. The Virtex®-7 FPGA and 28F00AG18F parallel NOR flash on the VC707 evaluation board are used to demonstrate the flow with the ISE® Design Suite 14.7. For flash programming examples using Vivado® tools, see the UltraScale FPGA BPI Configuration and Flash Programming Application Note (XAPP1220) [Ref 2].

Introduction

Figure 1 shows the two flows that are described in this document:

- 7 series BPI Fast Configuration process
- iMPACT indirect flash programming through the FPGA

Figure 1: BPI Fast Configuration and iMPACT Indirect Flash Programming Flows
BPI Configuration Basics

7 series FPGAs require a configuration bitstream to be delivered at power-up because they are CMOS-configurable latch-based devices. Parallel NOR flash is a popular option for storing and delivering the bitstream because the wide x16 data bus provides faster configuration over SPI flash memory alternatives. Specifications such as PCI Express® with strict power-up requirements need fast configuration like those achieved with the BPI Fast Configuration. In addition to the Fast Configuration, parallel NOR flash offers larger capacity than SPI flash memory. Systems that use parallel NOR flash memory for random-access, nonvolatile application data storage can also benefit from consolidating the configuration storage into a single memory device.

Xilinx also provides the ability to indirectly program parallel NOR flash in-system using the existing configuration connections between the parallel NOR flash and the FPGA. The iMPACT programming tool uses JTAG to configure the FPGA with a pre-made bitstream. This bitstream enables a path between the programming cable and the flash. In a lab environment, this solution provides the flexibility to program new configuration bitstreams into the parallel NOR flash without needing to remove the device from the board.

This application note is divided into the following key sections:

- BPI Configuration Basics, page 2
- Parallel NOR Flash Programming Basics, page 3
- BPI Fast Configuration Implementation, page 3
- BPI Fast Configuration File Generation and Flash Programming, page 11
- Checklist and Debug, page 24

BPI Configuration Basics

Configuration is the process of downloading configuration data into an FPGA using an external source such as a flash device, CPLD, or microprocessor. In the BPI configuration mode, the FPGA supports a direct connection to the address, x16 data bus, and control signals of a parallel NOR flash for extracting a stored bitstream. The 7 series FPGA BPI configuration interface supports two flash read options: asynchronous or synchronous. The BPI configuration mode supports the configuration data download by using either the FPGA CCLK (internally-generated configuration clock) or the EMCCCLK option.

Synchronous Read and EMCCCLK Advantages

BPI Fast Configuration uses the synchronous read and EMCCCLK options. Synchronous read can burst configuration data at frequencies up to 100 MHz after a valid starting address is given. This significantly decreases the configuration time from what can be achieved with the asynchronous read mode. In the legacy asynchronous read mode, the FPGA must increment the address at a frequency less than 10 MHz to support the supported flash read access specifications.
The configuration times can be further decreased by using the EMCCLK option in conjunction with the synchronous read. EMCCLK uses external oscillators that are capable of faster and more precise clock frequency options than provided with the FPGA internal CCLK. This enables more predictable configuration times. For example, if the maximum clock rate for a synchronous read setup is 100 MHz, an external clock of 99 MHz can be used for EMCCLK if the tolerance is 1%. This is an enhancement over the maximum 66 MHz CCLK rate for the same setup. The CCLK maximum is a lower target because of the tolerance $F_{MCCKTOL}$. For example, at a typical $F_{MCCKTOL}$ of ±50% the wide range of 33–99 MHz must be taken into account. Configuration Times, page 10 describes how to estimate the configuration time.

---

### Parallel NOR Flash Programming Basics

Parallel NOR flash memory devices have a single interface for programming and three primary methods to deliver the data to this interface:

- Third-party programmers (off-board programming)
- In-system programming (ISP) with an embedded processor
- Indirect ISP using JTAG or a custom solution

Key factors to consider in selecting the best programming solution for an application are:

- Prototyping or high-volume production current phase
- ISP or off-board programming required
- Direct or indirect programming connectivity available

For example, an off-board third-party programmer such as BP Microsystems should be considered if the most critical factor is to decrease flash programming times for a high-volume production application. Off-board programming can often deliver faster programming times because they can limit overhead by interfacing directly to the flash. This solution can also make use of the enhanced programming higher voltage option.

An application in the low-volume prototyping phase requiring the flash to be programmed onboard would benefit from the iMPACT indirect ISP programming tool. The iMPACT tool can accommodate the testing of multiple design iterations and is extremely useful for debugging in a lab environment, but it is not intended for high volume production programming.

---

### BPI Fast Configuration Implementation

7 series FPGAs have a configuration interface to third-party parallel NOR flash. The 7 series BPI configuration mode supports asynchronous read as in earlier FPGA families, but to address changing system requirements, new BPI configuration mode features have been added. The 7 series FPGAs are the first Xilinx FPGAs with the built-in ability to synchronously read from select parallel NOR flash memory for configuration. In addition, an EMCCLK option can be enabled to provide a highly accurate configuration clock. These two new features used
together enable significantly faster configuration times than in past generations. The sections that follow discuss the criteria for selecting a flash device and describe the interface and BPI Fast Configuration sequence.

Selecting Parallel NOR Flash

Several factors should be considered when selecting the parallel NOR flash device for a configuration source, such as the storage capacity required by the application, the package type to meet board space requirements, the data bus width for configuration time, and the flash I/O voltage range. See the 7 Series FPGAs Configuration User Guide (UG470) [Ref 1] for information on the bitstream size to determine the minimum flash density required.

Because the flash must be placed into synchronous read mode with a write command sequence, only the Micron P30 and G18F non-multiplexing (Non-MUX) family member devices are supported. This application note focuses on the G18F devices because they are capable of running twice the frequency of the P30 family. Table 1 shows the supported G18F devices.

Table 1: BPI Fast Configuration Parallel NOR Flash

<table>
<thead>
<tr>
<th>Flash Family Members</th>
<th>Density Range</th>
<th>$V_{CC}$/$V_{CCQ}$ (I/O)</th>
<th>iMPACT Indirect Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micron G18F(1)</td>
<td>128 Mb–1 Gb</td>
<td>1.8V/1.8V</td>
<td>See iMPACT manual [Ref 3] for release details.</td>
</tr>
</tbody>
</table>

Notes:
1. Micron Parallel NOR Flash 28F00AG18F had a name change to MT28GU01GAAA1E. There was no functional change to the flash device, but the address signal label names have changed slightly. For this application note, the 28F00AG18F signal names are used.

Voltage Compatibility

The parallel NOR flash I/O must support a voltage that is compatible with the connected FPGA pins. The dedicated configuration pins in bank 0 and the dual-purpose pins in banks 14 and 15 are used by the BPI Fast Configuration interface. These pins need to receive the same $V_{CCO}$ voltage supply to ensure a consistent I/O voltage for the BPI configuration interface. In addition, the JTAG pins are in bank 0 and follow the requirements for this bank. The configuration bank voltage select pin (CFGBVS) is set High in order to set the configuration-related I/O for 3.3V/2.5V operation, or set Low to set the configuration pins for 1.8V operation. This application note focuses on the Virtex-7 FPGA whose I/O are limited to 1.8V or lower standards on banks 14 and 15. In the Virtex-7 FPGA, the CFGBVS pin is set to Low to support BPI Fast Configuration. For additional details on CFGBVS, refer to the 7 Series FPGAs Configuration User Guide (UG470) [Ref 1].
BPI Fast Configuration Interface

Figure 2 shows associated signals for the BPI Fast Configuration interface. Refer to BPI Fast Configuration Sequence, page 9 for details on the configuration process steps for this mode.

![BPI Fast Configuration Interface Diagram]

**Note:** The RS[1:0] and EMCCLK signals are optional connections for the BPI configuration mode highlighted in this application note.

Table 2 describes the BPI Fast Configuration interface signals and their function in the configuration process. The connections to the parallel NOR flash memory are also provided.

<table>
<thead>
<tr>
<th>7 Series FPGA Pin Name</th>
<th>Type</th>
<th>Dedicated or Dual-Purpose</th>
<th>Description</th>
<th>Micron Flash Signal Connection (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCLK</td>
<td>Input or Output</td>
<td>Dedicated</td>
<td>Configuration clock output. This pin is the initial configuration clock source for all configuration modes except JTAG. CCLK is an output in master BPI configuration mode. During the BPI flash asynchronous read mode, CCLK does not directly clock the parallel NOR flash but is used internally by the FPGA to generate the address and sample read data. During the BPI flash synchronous read mode, CCLK must be directly connected to the parallel NOR flash to clock the data out sequentially.</td>
<td>CLK (Clock)</td>
</tr>
</tbody>
</table>
**Table 2: 7 Series FPGA BPI Configuration Signal Descriptions (Cont’d)**

<table>
<thead>
<tr>
<th>7 Series FPGA Pin Name</th>
<th>Type</th>
<th>Dedicated or Dual-Purpose</th>
<th>Description</th>
<th>Micron Flash Signal Connection[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DONE</td>
<td>Bidirectional, Open-drain, or Active</td>
<td>Dedicated</td>
<td>Active-High signal indicating configuration is complete: 0 = FPGA not configured 1 = FPGA configured</td>
<td>NC</td>
</tr>
<tr>
<td>INIT_B</td>
<td>Bidirectional</td>
<td>Dedicated</td>
<td>From power-on reset or PROGRAM_B reset, INIT_B is driven Low, indicating that the FPGA is initializing (clearing) its configuration memory. Before the mode pins are sampled, INIT_B is an open-drain, active-Low input and can be held Low to delay configuration. After the mode pins are sampled, the INIT_B output indicates whether a CRC error occurred during configuration or a readback CRC error occurred after configuration (when enabled): 0 = CRC or IDCODE error (DONE is Low) or readback CRC error (DONE is High and readback CRC is enabled) 1 = No CRC error, initialization is complete (needs an external pull-up resistor to VCCO_0)</td>
<td>/RST (Reset)</td>
</tr>
<tr>
<td>M[2:0]</td>
<td>Input</td>
<td>Dedicated</td>
<td>The mode pins determine the configuration mode. M[2:0]=010 for Master BPI mode.</td>
<td>NC</td>
</tr>
<tr>
<td>PROGRAM_B</td>
<td>Input</td>
<td>Dedicated</td>
<td>Active-Low asynchronous full-chip reset.</td>
<td>NC</td>
</tr>
<tr>
<td>A[28:00]</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Address output. Signals are used to increment the flash address and in synchronous read mode, these signals send the flash read configuration register content that set the flash in synchronous mode.</td>
<td>A[n:1] (Address inputs)</td>
</tr>
<tr>
<td>ADV_B</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Active-Low address valid output. Required for the synchronous read option in Master BPI Configuration mode. This signal is needed to tell when an address is valid in synchronous read mode. The signal also needs to be driven Low for asynchronous read mode.</td>
<td>/ADV (Address valid)</td>
</tr>
<tr>
<td>CSO_B</td>
<td>3-state output</td>
<td>Dual-purpose</td>
<td>Parallel daisy chain active-Low chip select output during configuration. Otherwise, this output is 3-stated. Not used in single FPGA applications.</td>
<td>NC</td>
</tr>
<tr>
<td>D[15:00]</td>
<td>Bidirectional</td>
<td>Dual-purpose</td>
<td>This x16 data bus is sampled by the rising edge of the FPGA CCLK. Data is read from the flash on this bus and commands to write to the flash read configuration register are sent on this bus when a synchronous read command is seen by the configuration controller.</td>
<td>DQ[15:0] (Data)</td>
</tr>
</tbody>
</table>
Figure 3 shows the BPI Fast Configuration example implementation on the VC707 evaluation board. The required signal connections to the parallel NOR flash are given in the figure. Signal connections for the JTAG port connections to support the iMPACT indirect flash programming are included.

<table>
<thead>
<tr>
<th>7 Series FPGA Pin Name</th>
<th>Type</th>
<th>Dedicated or Dual-Purpose</th>
<th>Description</th>
<th>Micron Flash Signal Connection(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMCCCLK</td>
<td>Input</td>
<td>Dual-purpose</td>
<td>External Master Configuration Clock input. This option is used for the BPI Fast Configuration mode. An external clock is supplied on this input and the FPGA configuration controller switches over to use this clock instead of the CCLK (internal configuration clock) after the EMCCCLK command is read from the bitstream header. The EMCCCLK enables more predictable configuration times because the clock tolerance is determined by the external oscillator selected.</td>
<td>NC</td>
</tr>
<tr>
<td>FCS_B</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Active-Low flash chip select output. This output is actively toggled during configuration.</td>
<td>/CE</td>
</tr>
<tr>
<td>FOE_B</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Active-Low flash output enable. This output is actively toggled during configuration.</td>
<td>/OE</td>
</tr>
<tr>
<td>FWE_B</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Active-Low flash write enable. This output is actively toggled during configuration.</td>
<td>/WE</td>
</tr>
<tr>
<td>PUDC_B</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Controls I/O (except bank 0 dedicated I/O) pull-up resistors during configuration. This pin must be externally terminated: 0 = Pull-up during configuration 1 = 3-state during configuration</td>
<td>NC</td>
</tr>
<tr>
<td>RS[1:0]</td>
<td>Output</td>
<td>Dual-purpose</td>
<td>Revision Select pins. Used for multi-bitstream applications to select between revisions and provide fallback capability. RS[1:0] are actively driven Low to load the fallback bitstream when a configuration error is detected. See the 7 Series FPGAs Configuration User Guide (UG470) [Ref 1] for additional details on revision management.</td>
<td>A[n:n–1]</td>
</tr>
</tbody>
</table>

Notes:
1. Refer to the Micron G18F flash data sheet [Ref 4] for flash signal descriptions and details.
2. The use of the revision select pins (RS[1:0]) is an optional feature used in cases where bitstream revision management is required.
Notes relevant to Figure 3:

1. The 7 series FPGA JTAG signals TCK, TMS, TDI, and TCK are not required for the BPI Fast Configuration interface, but these signals are necessary for the iMPACT indirect flash programming solution. The JTAG interface is also a popular debug interface used on many application setups. Refer to BPI Fast Configuration File Generation and Flash Programming, page 11 for steps to program the flash indirectly.
2. The EMCCCLK maximum frequency is dependent on the target flash and FPGA. See *Configuration Times, page 10* to determine the maximum frequency for a given setup. The VC707 board uses an 80 MHz clock source.

3. RS[1:0] are optional revision management pins. In applications that do not require revision control, the required address pins are connected from FPGA A[28:0] to the flash A[n:1] and RS[1:0] is a no connect.

4. The 7 series FPGA VCCO_0 supply must be compatible with the VCCQ on the parallel flash.

5. The Micron 28F00AG18F has Write Protect (/WP) and Wait signals that should be tied appropriately to allow for indirect flash programming and configuration.

**BPI Fast Configuration Sequence**

The BPI Fast Configuration sequence described in this section occurs automatically when the bitstream has been generated for the synchronous read mode and the mode pins M[2:0] are set to the BPI configuration mode. Refer to *BPI Fast Configuration File Generation and Flash Programming, page 11* for the steps to generate the bitstream.

In the BPI configuration mode, the 7 series FPGA always begins reading from the parallel NOR flash in asynchronous read mode. The FPGA increments the address bus from a given start address, and the parallel NOR flash sends back the bitstream data. The 7 series FPGA reads the bitstream header to determine the flash read mode selected for reading the configuration data. When a synchronous command is read in the bitstream header, the FPGA configuration controller performs an asynchronous write to the Read Configuration register (RCR) of the connected flash to set the synchronous mode and latency bits. After the flash RCR has been successfully written, the FPGA controller then initiates synchronous read and reads the bitstream data contents. After configuration completes, the flash is left in synchronous read mode.

The timing waveform in *Figure 4* shows the three steps to initiate the BPI configuration synchronous read.

![BPI Fast Configuration Timing Waveform](image)
The first step is reading the bitstream in asynchronous mode to determine the target mode. After the INIT_B signal is released and the control signals FCS_B, FOE, and ADV_B are asserted with a valid address A[28:00] being incremented, data is captured from the flash on the data bus D[15:0]. The bitstream header is read to determine which read mode is targeted by the design.

In the second step, the FPGA enables the flash synchronous mode by writing to the read configuration register of the flash. To write the flash synchronous mode bit, the FPGA issues an asynchronous write sequence. To perform the write operation, the FPGA asserts the FCS_B and FWE_B while the INIT_B and FOE_B are deasserted. During the write operation, the address and data are latched on the rising edge of FWE_B or FCS_B, whichever occurs first. The FPGA issues the flash configuration register write sequence of two write cycles. The first cycle has the Read Configuration register (RCR) data on A[15:0] and command 0x60 on the data bus. The second cycle has the RCR data on A[15:0] and the command 0x03 on the data bus. The RCR values are different for the Micron G18F and the P30 devices and is determined by the BitGen option -g BPI_sync_mode. Refer to Table 4 for options.

In the final step, the FPGA finally switches from the asynchronous read to synchronous read protocol and reinitiates the bitstream read data contents. This sequence is implemented by asserting the FCS_B and the FOE_B signals and having ADV_B asserted for one cycle with a valid address. The configuration data is then burst from the flash and read back by the FPGA.

It is important to understand that the flash is left in the same read mode that is used for configuration. For example, the flash is left in asynchronous read mode after the FPGA is configured in the asynchronous read mode. Alternatively, the flash is left in synchronous read mode after the FPGA is configured in the synchronous read mode.

**Configuration Times**

The BPI Fast Configuration setup in this application note uses EMCCCLK. To calculate the maximum EMCCCLK frequency, both the parallel NOR flash clock to out specification and the FPGA setup data sheet specifications are used. In addition, board delay needs to be considered. The maximum supported EMCCCLK frequency (F_EMCCCLK) is specified in the 7 series FPGA data sheets DS181, DS182, and DS183 [Ref 5] [Ref 6] [Ref 7] and must not be exceeded. An estimation for the maximum BPI Fast Configuration EMCCCLK can be calculated with Equation 1.

\[
\text{Maximum Frequency} = \frac{1}{\text{Flash clock to out (TCHQV)} + \text{FPGA data setup (TBPIDCC)} + \text{Board delay}}
\]  

Equation 1

**Note:** With a 28F00AG18F flash clock-to-out specification of T\text{CHQV} = 5.5\,\text{ns} and a Virtex-7 XC7VX485T FPGA data setup of T\text{BPIDCC} = 4\,\text{ns}, the maximum EMCCCLK would be capped by the data sheet specification F\text{EMCCCLK} if board delay was negligible.

The BPI Fast Configuration time on the VC707 board is given compared to the BPI Configuration mode with asynchronous read and CCLK on the same board in Table 3. The calculation used to estimate the configuration time is given in Equation 2.

\[
\text{Configuration time} = \frac{\text{bitstream size}}{\text{configuration clock frequency} \times \text{data bus width}}
\]  

Equation 2
For the asynchronous read calculation, additional parameters must be considered such as the FPGA $T_{BPICCO}$ + Flash $T_{ACC}$ + FPGA data setup $T_{BPIDCC}$, which determines the maximum clock frequency. After the maximum clock frequency is known, the CCLK tolerance ($F_{MCCKTOL}$) must be considered when using the internal configuration clock CCLK instead of the EMCCCLK. Although the 28F00AG18F flash is capable of running at slightly more than 9 MHz in asynchronous read mode, the $F_{MCCKTOL}$ limits the CCLK frequency setting to 6 MHz typical (or a range of 3 MHz–9 MHz).

### BPI Fast Configuration File Generation and Flash Programming

The ISE design tools create 7 series FPGA bitstreams and flash programming files, and indirectly program parallel NOR flash devices. Figure 5 highlights the steps (Design Entry, BitGen, PROMGen, and iMPACT) where special options are required to properly generate files for the BPI Fast Configuration mode and program the flash. Refer to the Virtex-7 FPGA VC707 Evaluation Board demonstrations for reference design examples using this flow and options at www.xilinx.com/support/documentation/vc707_14-1.htm.

**Figure 5:** ISE Design Tools Overview

This section provides setup and tool flow examples for the following target:

- ISE tools, version 14.7
- VC707 board (Virtex-7 XC7VX485T FPGA with 28F00AG18F Parallel NOR flash and USB cable)
Design Entry Preparation for EMCCLK

Applications that use the BPI Configuration EMCCCLK option must ensure that the I/O standard is defined on the EMCCCLK dual-purpose pin. If the VC707 Virtex-7 FPGA design uses the EMCCCLK after configuration, you need to include this user constraint option to enable the 80 MHz EMCCCLK:

```
NET "EMCCCLK" LOC = AP37 | IOSTANDARD = LVCMOS18;
```

Alternately, when the EMCCCLK is used only for the configuration clock and is not used after configuration, you need to define the EMCCCLK voltage by entering the following constraint in the PCF file:

```
CONFIG CONFIG_VOLTAGE = 1.8;
```

Preparing a Bitstream for BPI Configuration

This section provides the options needed to create a bitstream that can be used with the fastest BPI Configuration mode using the synchronous read. Table 4 provides the recommended options for the VC707 board and descriptions of the features related to the BPI Configuration mode. Options not specified in the BitGen command use the defaults.

**Table 4: BitGen Configuration Options for BPI Configuration Mode with Synchronous Read**

<table>
<thead>
<tr>
<th>BitGen Command Line</th>
<th>Description</th>
</tr>
</thead>
</table>
| `-g StartupClk:Cclk` | Option: StartupClk: Cclk|UserClk|JtagClk  
| Default: Cclk  
| Specifies the FPGA CCLK as the clock that drives the FPGA start-up sequence. The Cclk option is required for successful BPI configuration. |
| `-g Compress` | Option: Compress  
| Default: Off  
| Specifies that FPGA bitstream file compression is enabled. This is an optional switch but recommended to reduce both indirect flash programming and FPGA configuration time. Compression can significantly reduce the file size, but it is nondeterministic and varies with the user design. |
| `-g ConfigRate:3` | Option: ConfigRate:3|6|9|12|16|22|26|33|40|50|66  
| Default: 3  
| Specifies a nominal CCLK frequency of 3 MHz. The FPGA CCLK controls the FPGA BPI configuration timing. This application note assumes that EMCCCLK is utilized so the default is required and only used for the reading of the initial bitstream header. For designs that do not utilize the EMCCCLK, the ConfigRate option can be increased to a frequency that meets the maximum clock frequency supported for the FPGA and flash taking into account the FMCKTOL tolerance. |
BitGen Command Line Example

The bitstream generation tool BitGen can be run from a command line. An example for the VC707 target with the BPI Configuration synchronous mode is shown here:

```
bitgen.exe -w -g StartUpClk:CClk -g Compress -g ConfigRate:3 -g BPI_sync_mode:Type1 -g ExtMasterCclk_en:div-1 -g BPI_page_size:1 -g BPI_1st_read_cycle:1 design.ncd design.bit
```

Note: The VC707 board has an oscillator connected to the EMCCCLK FPGA I/O of 80 MHz so the div-1 option can be used without exceeding the flash or FPGA maximum synchronous frequency.

### BitGen Graphical User Interface Example

The bitstream can also be generated from the ISE tools Project Navigator flow (Figure 6). In the Project Navigator, the defaults are used together with these options:

- Under the Generate Programming File process, right-click to get the process properties:
  - For the Property Display Level, select **Advanced** to show all possible options.
  - Under General Options, the `-g Compress` switch is optional, but it is selected because it helps to reduce the programming and configuration time.
Under the Configuration Options category, ensure that the options shown in Figure 7 are selected to generate a bitstream for BPI configuration with synchronous read capability using the external master configuration clock.

Figure 6: General Options for BPI Fast Configuration Mode Bitstream Generation
Figure 7: Configuration Options for BPI Fast Configuration Mode Bitstream Generation
Creating the Parallel NOR Flash Programming File

The ISE tools PROMGen or iMPACT PROM File Formatter creates flash files. These ISE tools take an FPGA bitstream (.bit) and generate a flash file (.mcs) that can be used to program the parallel NOR flash. The options to create the file are described in Table 5.

Table 5: PROMGen Options

<table>
<thead>
<tr>
<th>PROMGen Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-p &lt;format&gt;</td>
<td>File format. The most common format is Intel hex (.mcs).</td>
</tr>
<tr>
<td>-c FF</td>
<td>Flash fill value.</td>
</tr>
<tr>
<td>-s &lt;size&gt;</td>
<td>Flash size in kilobytes (must be a power of 2). If not specified, this defaults to the size of the file(s).</td>
</tr>
<tr>
<td>-o &lt;filename&gt;</td>
<td>Output flash file name.</td>
</tr>
<tr>
<td>-u &lt;hexaddr&gt; &lt;file[.bit]&gt;</td>
<td>Loads the .bit file from the specified starting address in an upward direction. This option must be specified immediately before the input bitstream file.</td>
</tr>
<tr>
<td>-data_width</td>
<td>Specifies the data width of the targeted flash. Bit/byte ordering should be changed between x8 and x16 in the device bitstream depending on the target flash data bus width. The x16 ordering is the only mode supported for BPI Configuration interface synchronous read.</td>
</tr>
<tr>
<td>-w</td>
<td>Overwrite existing output file.</td>
</tr>
<tr>
<td>-data_file</td>
<td>Specify the starting address, direction, and data files to add into the flash file.</td>
</tr>
<tr>
<td>-bpi_dc</td>
<td>Selects a daisy-chain output from the first FPGA. Parallel is the only valid option for the 7 series FPGA BPI Fast Configuration mode.</td>
</tr>
</tbody>
</table>

PROMGen Command Line

The ISE tools PROMGen is executed from a command line. Here is an example PROMGen command line:

```
Promgen -w -p mcs -c FF -bpi_dc parallel -data_width 16 -o flash_file_name -s 131072 -u 00000000 design.bit
```

This command line generates a parallel NOR file with these options:

- format = mcs
- data bus = 16
- flash size = 1 Gb (131,072 x 1,024 bytes x 8 bits = 1,073,741,824)
- BPI from address 0
- output file = design.mcs
- fill value = FF
- input bitstream = design.bit
**PROM File Formatter GUI Flow Example**

Within the iMPACT tool, a flash file can also be generated using these steps:

1. Open iMPACT and double-click the Create PROM File (PROM File Formatter) iMPACT Flow in the upper left box.

2. A wizard display guides you through the next steps (Figure 8):
   a. Under “Step 1. Select Storage Target” select **BPI Flash > Configure Single FPGA** and click the green arrow.
   b. Under “Step 2. Add Storage Device(s)” select the target FPGA (e.g., **Virtex-7**) and the storage device (bytes) (128M for the 28F00AG18F) and then click **Add Storage Device**. Click the green arrow.
   c. Under “Step 3. Enter Data” select the output file name, output location, and flash properties. Click **OK**.

3. When prompted, click **OK** and add the target user design bitstream (.bit) file. For a single design image, click **No** at the next prompt that is used for multiple designs. Click **OK** to confirm the completion of the design file entry and again at the summary of usage.

4. After successfully running a project, click **Generate File...** to create the flash programming image.
Virtex-7 FPGA VC707 Board Setup for iMPACT Indirect Flash Programming

Refer to the VC707 Evaluation Board for the Virtex-7 FPGA User Guide (UG885) [Ref 8] for detailed information on features and board usage. The basic board setup to program the parallel NOR flash with iMPACT and to properly configure the Virtex-7 FPGA is briefly highlighted below:

1. Ensure the defaults for the Mode pins M[2:0] and the two upper address lines A[25:24] tied to Revision Select RS[1:0] are set as shown in Figure 9.

2. Power on the VC707 board.

3. Connect a USB Type-A to Micro-B cable to the USB JTAG (Digilent) connector on the VC707 board or connect a Xilinx USB II cable to the header.

Cable Options

The iMPACT programming tool uses a Xilinx Platform Cable USB II cable or the Digilent JTAG SMT1 module (on board) to indirectly program the VC707 parallel NOR flash using the Virtex-7 FPGA JTAG interface. Table 6 lists the cables and the maximum supported frequency for the iMPACT indirect programming solution.

<table>
<thead>
<tr>
<th>Cables</th>
<th>Interface</th>
<th>BPI Indirect Programming Max Frequency Supported (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xilinx Platform Cable USB II</td>
<td>USB</td>
<td>12</td>
</tr>
<tr>
<td>Digilent JTAG SMT1 Module (Onboard)</td>
<td>USB</td>
<td>30</td>
</tr>
</tbody>
</table>

Refer to the specific USB cable data sheet for additional information on the JTAG signal connectivity and header details.
**iMPACT Indirect Parallel NOR Flash Programming**

The iMPACT tool is used to indirectly program the parallel NOR flash. Because parallel NOR flash memory devices do not have a JTAG interface, extra logic is required to serve as a bridge between the iMPACT tool and the flash. For this programming solution, iMPACT configures the FPGA with a bitstream that serves as a bridge between the IEEE 1149.1 (JTAG) bus and the parallel NOR flash bus interface. This solution is referred to as “indirect programming” because the flash is not directly programmed from iMPACT but is programmed through the FPGA. The basic setup of the iMPACT indirect programming solution is shown in Figure 10 for the Virtex-7 FPGA VC707 evaluation board.

![Figure 10: iMPACT Indirect Flash Programming Setup](image)

The Virtex-7 XC7VX485T device and the Micron 28F00AG18F parallel NOR flash on the VC707 evaluation board are used to demonstrate the iMPACT programming flow. Before beginning the programming steps, the basic board and cable setup details provided in Virtex-7 FPGA VC707 Board Setup for iMPACT Indirect Flash Programming, page 18 should be followed.

After the board is set up, the onboard flash can be programmed without having to remove it from the board. The 7 series FPGA must first be configured using JTAG with the iMPACT pre-made bitstream that bridges the programming cable to the flash. The existing contents of the FPGA are cleared in order to load the iMPACT pre-made bitstream. The bitstream design 3-states all unused I/Os that are not part of the parallel NOR interface. Be aware of this step and ensure that this does not have any undesired effect on other devices attached to the FPGA in the application. If there are unused I/Os that should not be floating, then external resistor termination would be required. The bitstream design is integrated into the iMPACT programming tool and automatically loaded in the background during an indirect flash operation. The core is for iMPACT runtime use and the source code is not available for design integration. After the bitstream design is loaded, the iMPACT tool issues the command sequence to the flash based on the target device and operation selected.

The iMPACT tool can be run in batch mode or through the GUI. The iMPACT steps to program the flash indirectly are shown in the next section.
**iMPACT Batch Example to Indirect Program Parallel NOR Flash**

This section shows popular operations and options for running iMPACT in batch mode with a command file.

```
iMPACT -batch design.cmd
```

Sample design.cmd contents:
- setMode -bs
- setCable -port auto
- Identify -inferir
- identifyMPM
- attachflash -position 1 -bpi "28F00AG18F"
- assignfiletoattachedflash -position 1 -file "designname.mcs"
- Program -p 1 -dataWidth 16 -rs1 25 -rs0 24 -bpionly -e -loadfpga
- closeCable
- Quit

Optional instructions include flash verify and readback commands:
- BlankCheck -p 1 -dataWidth 16 -rs1 25 -rs0 24 -bpionly -e -loadfpga
- Verify -p 1 -dataWidth 16 -rs1 25 -rs0 24 -bpionly -e -loadfpga

**iMPACT Graphical User Interface Example Steps to Program Parallel NOR Flash**

1. Open the iMPACT design tool standalone or from within ISE tools Project Navigator by selecting the **Configure Target Device** process. In iMPACT, double-click the **Boundary Scan** selection under the iMPACT Flows in the upper left of the screen.
2. Select the **Initialize Chain** icon to identify the target device on the board.
3. iMPACT prompts you to assign files. Click **No**.
4. Right-click the **SPI/BPI?** dotted box above the FPGA icon and select **Add SPI/BPI Flash**. When prompted, assign the target flash image.
5. Select **BPI PROM** and **28F00AG18F** for the 1 Gb parallel NOR flash that is onboard the VC707 board. Ensure that the bus data width is set to **16** and that the RS[1:0] selection is **25:24**. Click **OK**.

**Note:** On the VC707 board, the iMPACT core drives the FPGA RS0 and RS1 signals with this selection because they are the FPGA pins connected to the upper flash address bits A24 and A25, respectively. The RS[1:0] pins are optional and used for external revision control if multiple designs are stored in a single flash. If the RS[1:0] pins are not utilized by a design for revision control, this option during programming should be set to **Not Used**.
6. After the parallel NOR flash target has been selected, right-click the flash icon above the FPGA icon to choose the operation to perform on the flash or to select the operation properties. Click the **Program** operation which by default always performs a design-specific erase, program, and verify.

**Note:** In a design-specific erase, the flash contents are erased in the address space addressed by the targeted programming file to the nearest flash boundary. To erase the entire flash, the **Set Erase Properties...** selection should be chosen and changed to **Full Chip Erase**.
After issuing the flash operation, the iMPACT tool first downloads the pre-made bitstream and verifies some common parameters for density and family in the common flash interface (CFI). An example iMPACT log display for the CFI query on the VC707 28F00AG18F is shown below:

```
Populating BPI common flash interface ...
Common Flash Interface Information Query completed successfully.
INFO:Cae - Common Flash Interface Information from Device:
INFO:Cae - Verification string: 51 52 59
INFO:Cae - Manufacturer ID: 89
INFO:Cae - Vendor ID: 00
INFO:Cae - Device Code: 1b
Reset Core
Using x16 mode ...
Set Data Width
Using RS Pins [25,24]...
```

**Figure 12:** Indirect Flash Operation
iMPACT Indirect Flash Operation Time Estimates

Using iMPACT 14.7 with the VC707 board, some reference sample operation times are provided for compressed and uncompressed XC7VX485T bitstreams using the board's embedded USB cable at a default of 10 MHz (Table 7). These values are for reference only and not guaranteed timing.

<table>
<thead>
<tr>
<th>Image</th>
<th>Times (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Erase</td>
</tr>
<tr>
<td>XC7VX485T Sample Compressed 15 Mb</td>
<td>12</td>
</tr>
<tr>
<td>XC7VX485T Uncompressed 157.4 Mb</td>
<td>82</td>
</tr>
<tr>
<td>1 Gb Full Image</td>
<td>103</td>
</tr>
</tbody>
</table>

Effect on the Rest of the System from iMPACT Pre-Made Bitstream

The iMPACT JTAG-to-BPI pre-made bitstream is loaded into the FPGA before performing the target flash operation. The JTAG-to-BPI bitstream 3-states all unused I/Os. If the FPGA I/O is controlling any other devices, pull-ups or pull-downs should be added to disable the device during programming if a floating I/O will cause undesired behavior.

When the iMPACT bitstream is loaded, it drives the FPGA signals as required by the flash operation. The signals driven include the interface signals A[28:00], D[15:00], FCS_B, FWE_B, FOE_B, ADV_B, and optionally RS[1:0]. The I/O standard used by the design is compatible with the CFG8VS 1.8V or 3.3V selection. When the pre-made bitstream controller is not active, these flash interface signals are 3-stated and should be externally terminated as needed by the user application.

In applications using the RS[1:0] revision select capability, the RS[1:0] pins from the FPGA are tied to the flash upper address signals. Although the design might not have the address pins connected, the core drives both the RS[1:0] and A[28:00] pins. For example, in a 1 Gb flash device, if the upper address lines (A[25:24]) are associated with RS[1:0], both the FPGA A[25:24] signals and the RS[1:0] signals are driven by the core during the flash operation.
Checklist and Debug

This section summarizes the considerations and common debug steps for the BPI Configuration mode and indirect flash programming.

Configuration

• 7 series FPGA Status register data can be read in iMPACT via JTAG. For example, if the DONE and INIT signals are Low, this register captures the specific error conditions that can help identify the type of failure. In addition, the Status register allows you to verify the Mode pin settings M[2:0] and the bus width detect (for x16, this should be CFG BUS [25:24]=01).

• A bitstream generated for asynchronous mode and the internal configuration clock default can be a useful debug tool if a configuration failure is encountered using the synchronous read. The asynchronous read bitstream test helps to ensure that the interface connectivity on the board is correct and that there is not a maximum clock frequency violation. A new asynchronous read bitstream with the `-g ExtMasterCclk_en:Disable` and `-g BPI_sync_mode:Disable` options would be generated for this test.

• Virtex-7 FPGAs fallback to the default Master BPI with asynchronous read when the fallback is enabled. For applications that use the synchronous read and fallback options, the slower frequency requirements of asynchronous read need to be considered for the implementation.

• When RS[1:0] pins are used, the FPGA bitstream image must be less than 1/4 of the flash density to successfully configure in asynchronous read mode. When using the RS pins, the upper two flash address pins are held statically by the RS[1:0]. This allows for easy revision selection between up to four regions, however, it also does not enable the address to count beyond the quarter of the flash targeted by the static RS[1:0] pins.

• For FPGA designs that do not access the parallel NOR flash, the following is a list of recommendations for maintaining the flash in a safe, low-power condition while the FPGA user design is running:
  - Drive the FPGA FCS_B pin High
  - Drive the FPGA FOE_B pin High
  - Drive the FPGA FWE_B pin High

File Generation

• RTL entry: Ensure that the EMCCCLK I/O standard is defined. See Design Entry Preparation for EMCCCLK, page 12.

• BitGen usage: Ensure that the Synchronous mode and EMCCCLK are enabled with `-g BPI_sync_mode:Type1 -g ExtMasterCclk_en:div-1`. Optionally, for faster configuration and programming times, use the `-g compress` option. Also, ensure that the ConfigRate option does not exceed the maximum supported frequency supported by the target flash and FPGA for asynchronous read mode.
• PROMGen usage: Ensure that the option -data_width 16 data bus width is selected as described.

**iMPACT Indirect Programming**

• Erase operation: Flash devices are nonvolatile devices and must be erased before programming.

• BlankCheck operation: This operation can be used for verifying a prior erase operation.

• Maximum cable speed for flash operations: Ensure that the maximum cable speed for indirect programming is not exceeded. See Cable Options, page 18.

• Revision select pin usage RS[1:0]: These pins support an optional feature. If this feature is used instead of the FPGA upper two address bits for revision control, the option should be selected in iMPACT.

---

**Conclusion**

The 7 series BPI Fast Configuration uses the synchronous read and EMCCLK options to significantly decrease the required FPGA configuration time. This mode assists designers with meeting strict power-up requirements of popular standards such as PCIe. The ability to in-system program the parallel NOR flash for this solution provides an additional system value for the 7 series FPGA.

---

**References**

This following references provide additional information useful to this application note:

1. 7 Series FPGAs Configuration User Guide (UG470)
2. UltraScale FPGA BPI Configuration and Flash Programming Application Note (XAPP1220)
3. iMPACT Manual
4. Micron StrataFlash Embedded Memory Data Sheet: www.micron.com/~/media/documents/products/data-sheet/nor-flash/parallel/g18/256_512_1gb_g18_it.pdf
5. Artix-7 FPGAs Data Sheet: DC and Switching Characteristics (DS181)
6. Kintex-7 FPGAs Data Sheet: DC and Switching Characteristics (DS182)
7. Virtex-7 FPGAs Data Sheet: DC and Switching Characteristics (DS183)
8. VC707 Evaluation Board for the Virtex-7 FPGA User Guide (UG885)
9. Virtex-7 FPGA VC707 Evaluation Kit website
10. ISE Design Suite software manuals
11. **iMPACT Flash Memory Device Support Table**

---

**Revision History**

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/25/2012</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
<tr>
<td>05/31/2012</td>
<td>1.0.1</td>
<td>Updated PDF document properties.</td>
</tr>
<tr>
<td>05/10/2013</td>
<td>1.1</td>
<td>Updated description after Figure 10. Updated first paragraph of Effect on the Rest of the System from iMPACT Pre-Made Bitstream.</td>
</tr>
<tr>
<td>02/09/2015</td>
<td>1.2</td>
<td>Updated for ISE Design Suite 14.7. Revised Figure 4.</td>
</tr>
</tbody>
</table>

---

**Please Read: Important Legal Notices**

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults; Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at [http://www.xilinx.com/legal.htm#tos](http://www.xilinx.com/legal.htm#tos); IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at [http://www.xilinx.com/legal.htm#tos](http://www.xilinx.com/legal.htm#tos).

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.

© Copyright 2012–2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.