VANTIS
BEYOND PERFORMANCE

VF1 FPGA Configuration Guide

INTRODUCTION

Configuration is the process of loading a VF1™ FPGA with a pattern that defines its applications identity. Because the VF1 family uses SRAM configuration memory, configuration must be performed every time the device is powered up. Configuration may also be done at any time that the system designer determines that configuration is needed.

The configuration program is a bitstream generated by Vantis DesignDirect™ place-and-route software. The bitstream is stored in either a host system or in a serial PROM (SPROM). It is downloaded from the host system or SPROM whenever the VF1 FPGA is powered up or whenever the host system initiates the configuration process.

Configuration Overview

Designers have a wide selection of configuration modes that may be used to configure a VF1 FPGA from either a host system or SPROM (Figure 1). If a VCM (Vantis Configuration Memory) SPROM is used, the VF1 FPGA will automatically read its configuration program from the SPROM whenever power is applied. A host system, on the other hand, configures the VF1 FPGA under the control of the host system, and may reconfigure the VF1 FPGA at any time.

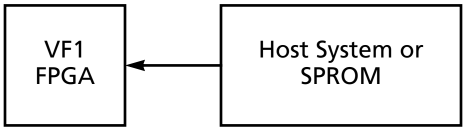


Figure 1. FV1 Device Configuration

21552B-001

The Vantis VF1 FPGA family supports configuration via the JTAG boundary-scan port as well as via a dedicated configuration port using non-JTAG modes. In this discussion, the non-JTAG modes will be covered first, followed by the JTAG mode.

The basic configuration modes are listed below and are described in detail in a following section.

- **Master serial mode.** In this mode, the Master VF1 FPGA is loaded automatically from a companion SPROM whenever power is applied to the VF1 FPGA or when the $\overline{\text{PROGRAM}}$ pin is pulsed. The VF1 FPGA controls the entire configuration process. Multiple VF1 FPGAs may be configured in this mode with the first device configured as a Master, and subsequent devices configured as Slaves.
- **Slave serial mode.** With the exception of JTAG mode, which is described later, slave mode is used whenever a VF1 FPGA is not controlling the configuration process. Usually, slave-mode devices are the second and subsequent VF1 FPGAs when multiple devices are daisy-chained for

Publication# **TN001-1**
Amendment/**0** Issue Date: **November 1998**

21552B-0configuration. VF1 FPGAs may also be configured in Slave mode whenever a host system loads the devices serially and controls the configuration process.

- **Asynchronous peripheral mode.** This mode is used to transfer configuration data a byte at a time from a host system to a VF1 FPGA. The VF1 FPGA provides the configuration clock. The host system monitors a status line to determine when another byte of data can be transferred. The first device in a chain is configured in Asynchronous Peripheral mode and subsequent devices are configured in Slave serial mode.
- **Synchronous peripheral mode.** This mode is similar to asynchronous peripheral mode in that a host system transfers a byte of configuration data at a time, but the host system provides configuration clocks. The VF1 FPGA receives the byte of configuration data, serializes is, and stores it in configuration memory. The first device in a chain is configured in Synchronous Peripheral mode and subsequent devices are configured in Slave serial mode.
- **JTAG mode.** JTAG Mode allows a host system to configure VF1 FPGAs via the VF1 JTAG port. When this mode is used, all VF1 FPGAs in a chain are configured in JTAG mode.

Configuration Program

The configuration program is developed using Vantis' DesignDirect software (Figure 2). Designers select the configuration mode that will be used for their applications and the DesignDirect software generates a bitstream in the appropriate format for that specific mode.

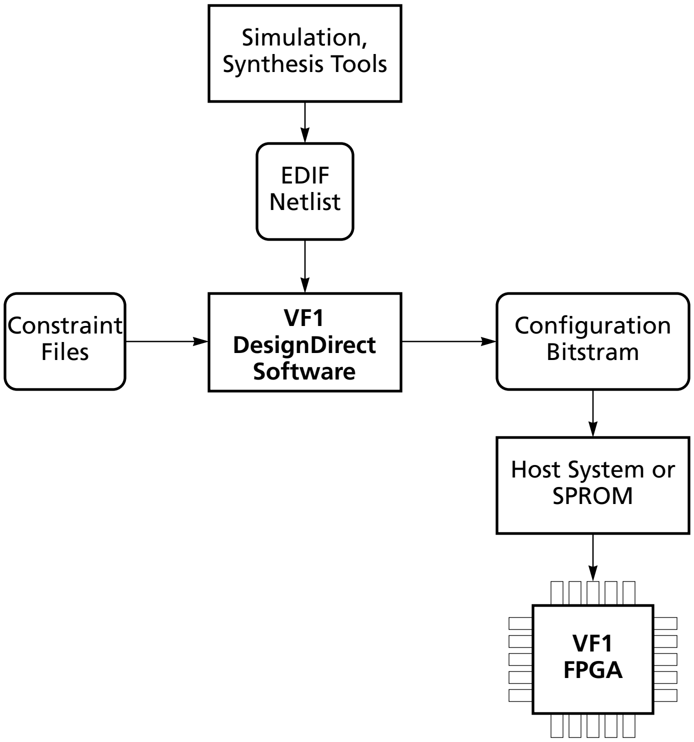


Figure 2. Configuration Program Development

21552B-002

With the exception of some header information, the configuration bitstream is the same regardless of the configuration mode used. The bitstream consists of a header; length count; multiple data frames, each with its own CRC for error detection; and a postamble (Table 1).

Table 1. Bitstream Format

Header	
-Fill bits	11111111
-Preamble	11110010
Length count	Count bits 23:0
Fill bits	11111111

Data frame(s)	Number of frames for each device type:
	VF1012=430 VF1020=546 VF1025=604
	VF1036=720

-Start bit	0
-Data	Configuration data
-CRC bits	Error detection bits
-Stop bits	111
Postamble	11111111

Note:
The total number of bits required to configure a VF1 FPGA depends on the capacity of the device (Table 2).

Table 2. VF1 FPGA Configuration Bits

VF1 FPGA	Configuration Bits
VF1012	245,156
VF1020	385,476
VF1025	474,792
VF1036	669,656

Non-JTAG Overall Timing/Signal Functions

A relatively small number of signals perform the most important configuration functions in non-JTAG configuration modes. Figure 3 is a generic configuration diagram showing the key signals used to interface a VF1 FPGA to either a host system or an SPROM. Other signals take part in various modes (Appendix A, Tables A1 and A2), but the signals shown in Figure 3 and the general timing diagram in Figure 4 are common to all non-JTAG configuration modes:

1. **$\overline{\text{PROGRAM}}$**. A low-to-high transition on the $\overline{\text{PROGRAM}}$ pin starts the configuration sequence.
2. **M[2:0].** Three mode pins select the VF1 configuration mode.
3. **$\overline{\text{INIT}}$**. The $\overline{\text{INIT}}$ signal is pulled low during VF1 initialization prior to configuration, is released during configuration, and is pulled low if a bitstream error occurs during configuration. $\overline{\text{INIT}}$ is an open drain signal that requires a pull-up resistor tied to V_{CC}.
4. **DONE.** The DONE signal goes low during initialization and stays low until the VF1 FPGA is fully configured. DONE is an open drain signal that requires a pull-up resistor tied to V_{CC}.
5. **CCLK.** In Master mode and Asynchronous Peripheral mode, the VF1 FPGA generates its own configuration clock. In these modes, CCLK is an output. In Slave serial mode and Synchronous Peripheral mode, the CCLK configuration clock comes from some other source. In these cases, CCLK is an input.

6. **DIN[7:0].** Configuration data is received on the data in (DIN) pins. In serial configuration modes, all data is received on DIN 0, commonly referred to as DIN. In Synchronous and Asynchronous peripheral modes, data is received in bytes on DIN 0-7.
7. **DOUT.** VF1 FPGAs may be daisy chained for configuration, with all devices configured from a common source. In this case, the lead VF1 FPGA configures itself first and passes subsequent configuration data to the devices that follow via the DOUT pin. The DOUT pin is connected to the DIN pin of the following device.
8. **RDY/BUSY.** When the VF1 FPGA is in one of the Peripheral modes, host systems monitor the RDY/BUSY signal to determine when the device is ready to receive another byte of configuration data.
9. **RDY/BUSY (DIN7).** Instead of monitoring the RDY/BUSY pin, a host system can read the RDY/BUSY status by monitoring the DIN7 pin.
10. **HDC, LDC.** Two status signals, HDC (high during configuration) and LDC (low during configuration) may be monitored during the configuration process. Once configuration is completed, the status of these pins is defined by the configuration pattern and they become user I/O pins.

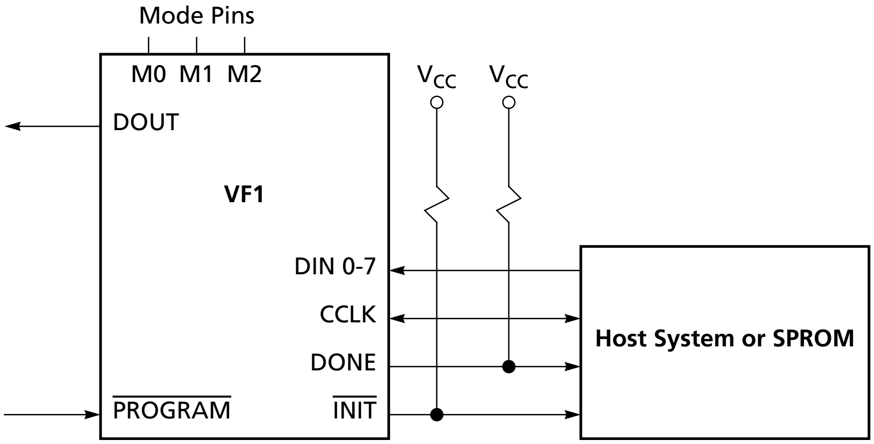


Figure 3. Common Non-JTAG Configuration Signals

Mode Selection

Three multiplexed mode-select pins, M[2:0], determine which mode will be used to configure a VF1 FPGA. Mode signals must be set up before power is applied to the chip and must remain constant until the VF1 FPGA completes the initialization process (while INIT is low). Mode signal setup is usually accomplished by tying the mode pins to the appropriate logic level via pull-up or pull-down resistors. Using resistors rather than tying the signals directly to VCC or GND frees the mode pins for use as I/O pins following the configuration process. Table 3 lists the pin assignments for each mode.

Table 3. Configuration Mode Selection Pins

	M2	M1	M0
Master Serial	0	0	0
Slave Serial	1	1	1
Synchronous Peripheral	0	1	1
Asynchronous Peripheral	1	0	1
JTAG	0	0	1

Initialization

Figure 4 shows the timing relationships that exist between key signals during configuration of a VF1 FPGA, and Table 4 lists specific timing values for each VF1 family member. Initialization begins whenever there is a low-to-high transition on the PROGRAM pin, or when power is applied to the device. When PROGRAM goes low, the VF1 FPGA clears configuration memory and waits for PROGRAM to go high. When PROGRAM goes high, several events occur:

1. The VF1 configuration memory is reset again.
2. The VF1 INIT pin goes low while the device initializes itself for the configuration process.
3. The VF1 examines the mode pins M[2:0] to determine the configuration mode that will be used.
4. The DONE pin goes low, signaling that the VF1 FPGA must be configured. It stays low until the configuration process is completed.
5. The INIT pin goes high when initialization is complete and the VF1 FPGA is ready to receive configuration data.

When the VF1 FPGA is configured from a companion SPROM such as the Vantis Configuration Memory (VCM), the DONE and INIT signals control the operation of the SPROM, as explained in the Master serial mode description that follows. When a host system configures the VF1 FPGA, DONE and INIT provide status information for the host.

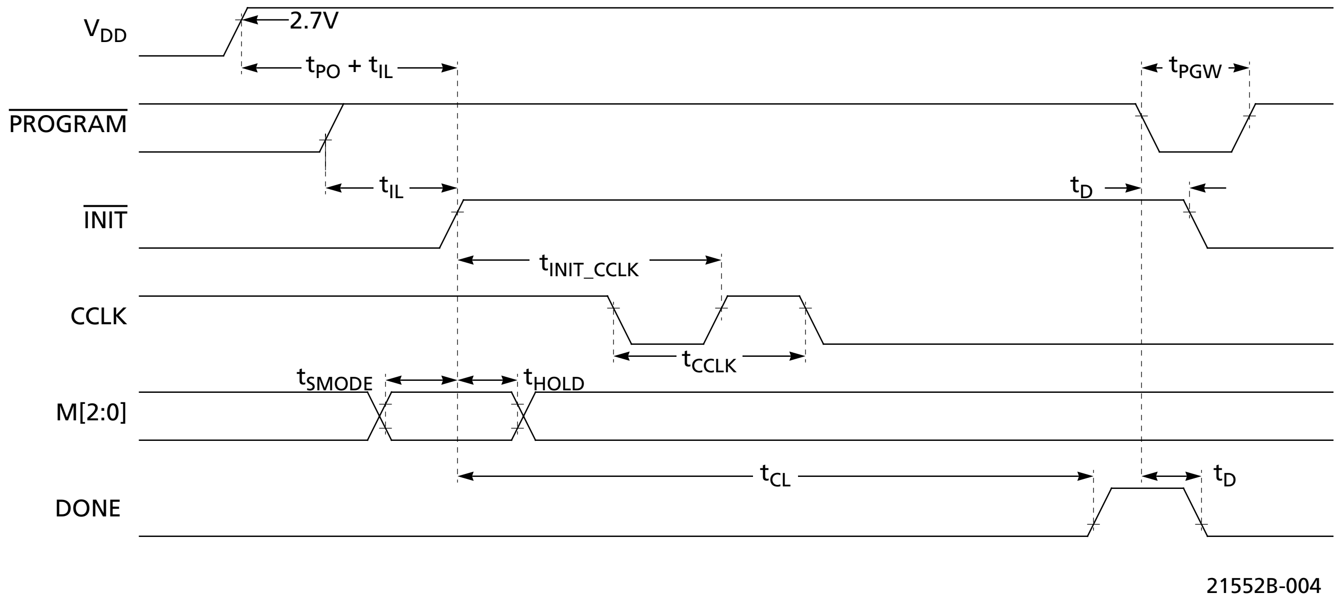


Figure 4. General Configuration Timing Characteristics

Table 4. General Timing Specifications

Parameter	Symbol	Min	Max	Unit
All Configuration Modes				
M[2:0] Setup Time to INIT High	t_{SMODE} (Note 2)	500		ns
M[2:0] Hold Time from INIT High	t_{HMODE}	500		ns
PROGRAM Pulse Width Low to Start Reconfiguration	t_{PGW}	1.5		µs
DONE or INIT Response Time to PROGRAM Pulse	t_{D}		1.0	µs
INIT Timing				
INIT High to CCLK Delay	t_{INIT_CCLK}			
Slave Serial		1.0		µs
Synchronous Peripheral		1.0		µs
Master Serial		200	900	ns
Initialization Latency	t_{IL}			
12K: Master Serial		165	620	µs
Other Modes		30	110	µs
20K: Master Serial		175	650	µs
Other Modes		35	140	µs
25K: Master Serial		180	665	µs
Other Modes		40	150	µs
36K: Master Serial		185	690	µs
Other Modes		50	180	µs
Power-On Reset Delay	t_{PO}			
Master Serial		35	130	ms
Other Modes		9	33	ms
Configuration Latency	t_{CL}			
12K: Master Serial		15	60	ms
Other Modes		25		ms
20K: Master Serial		25	100	ms
Other Modes		40		ms
25K: Master Serial		30	120	ms
Other Modes		45		ms
36K: Master Serial		45	165	ms
Other Modes		65		ms

Notes:

1. *All timing values are based on 1K pullup and 35 pF loading on open-drain outputs INIT and DONE.*
2. *In the case of Master mode, the minimum setup time is 500 ns to INIT High of Slave mode.*

Configuration Program Transfer

Once it is initialized, the VF1 FPGA reads configuration data from an SPROM or a host system. The following sections describe specific methods used to transfer data.

Table 1 describes the format of the configuration bitstream. A header is read first, followed by a bit count of the configuration program. If two or more VF1 FPGAs are configured in a daisy chain, the bit count includes the total of all configuration programs contained in the bitstream.

The bit count is loaded into a register in the VF1 FPGA. Even before the bit count is loaded, the VF1 FPGA begins counting every clock pulse on the CCLK pin. CCLK clocks may be generated internally in a VF1 FPGA (Master serial and Asynchronous peripheral modes) or may be received from some other source (Slave and Synchronous peripheral modes). Regardless of the source, every clock is counted and the count used to determine when configuration is complete. Since the VF1 FPGA reads one bit of data on each clock pulse, the clock pulse count corresponds to the number of data bits read by the device.

The VF1 FPGA loads configuration data a frame at a time. When the number of bits transferred matches the bit count that was read from the bitstream, the VF1 DONE signal goes high, signaling an end to the configuration process. In addition, either of the two status signals, HDC and $\overline{\text{LDC}}$, may be used to monitor the configuration process to determine when the VF1 FPGA is ready to process meaningful data.

Error Detection

During configuration, a VF1 FPGA checks CRC bits frame-by-frame to detect bitstream errors. If no errors occur, a high on the DONE pin signals the successful completion of the configuration process. If DONE is low, indicating that configuration did not complete successfully, the $\overline{\text{INIT}}$ signal may be sampled to help determine the reason. For example, a high $\overline{\text{INIT}}$ may indicate the VF1 FPGA was under-clocked, while a low $\overline{\text{INIT}}$ points to a bitstream error during configuration.

Configuration does not automatically begin again when an error occurs. The $\overline{\text{PROGRAM}}$ signal must be pulsed low to restart the configuration process.

JTAG Considerations in Non-JTAG Configuration Modes

Vantis VF1 FPGAs support JTAG boundary scan testing (IEEE STD 1149.1, IEEE standard test access port and boundary scan architecture). In addition to specifying industry-standard testing procedures, the IEEE specification gives manufacturers the ability to use JTAG ports for other purposes as well. VF1 FPGAs support both board level and device testing using JTAG, and provide a means for configuring VF1 FPGAs via the JTAG port. JTAG configuration is covered in a subsequent section of this document. This section discusses precautions that must be taken when using JTAG testing and non-JTAG configuration.

JTAG boundary scan instructions can control VF1 FPGA I/Os during JTAG testing. In normal mode (after completion of configuration), this does not create a problem because all JTAG instructions are supported. The problem may arise during non-JTAG mode configuration when DIN, DOUT, CCLK, $\overline{\text{INIT}}$, and DONE are sending out or receiving configuration handshaking signals.

The JTAG instructions INTEST, EXTEST or HI-Z, if executed during non-JTAG configuration, the outputs in the boundary scan register may corrupt the configuration process by overriding the configuration signal. Because of this probable conflict, the INTEST, EXTEST, and HI-Z instructions should not be used during non-JTAG mode configuration.

When the VF1 FPGA first powers up, its JTAG engine is initialized at TEST LOGIC RESET until V_{CC} reaches 2.7V and the VF1 internal power-up reset signal is released. This prevents the execution of any JTAG instructions. All the I/Os that are not used for configuration come up in 3-state mode with pull-ups. The boundary scan registers do not control the I/Os at this time.

After the power-up reset delay runs out, the VF1 begins its initialization in preparation for configuration. At this point, it is possible to start running BYPASS and instructions other than INTEST, EXTEST, and HI-Z.

When using non-JTAG configuration and JTAG boundary scan testing, you can handle potential conflicts in one of two ways:

1. Delay boundary scan testing until VF1 configuration is complete. The advantage of this approach is that all JTAG instructions for testing the VF1 FPGA are supported. The disadvantage is that the configuration process uses some board interconnections (those that interface the VF1 FPGA to its configuration source) before they are tested.
2. Use boundary scan to test board-level connections before configuring the VF1 FPGA, and use other JTAG tests following configuration. This approach allows board-level connections to be verified prior to configuration and chip functionality to be tested following configuration. However, it is a little more complex to implement than the first alternative.

Delay Boundary-Scan Testing

Delaying boundary-scan testing until VF1 configuration is complete is simple: monitor the VF1 DONE pin to ensure that configuration is complete before initiating any testing. The testing can begin as soon as the DONE pin goes high.

Be careful during boundary scan testing to ensure that testing does not re-trigger the configuration process. If the test process loads a “0” on the VF1 $\overline{\text{PROGRAM}}$ pin, a low-to-high transition will occur on the pin when testing is complete and the pin is allowed to go high again, thus initializing the VF1 and triggering a new configuration cycle. To avoid this problem, load a “1” on the $\overline{\text{PROGRAM}}$ pin to ensure that the signal remains high during boundary scan testing. Safe state information for the $\overline{\text{PROGRAM}}$ pin is included in the BSDL files for the VF1 FPGAs.

Boundary Scan Testing Before Configuration

Executing the JTAG boundary scan instructions prior to configuration allows board traces to be verified before they are used in the configuration process. To do boundary scan testing before configuration, follow these steps:

1. Hold the VF1 $\overline{\text{PROGRAM}}$ pin low after power-up, thereby holding the VF1 in the WAIT state and delaying initialization.
2. Perform boundary scan testing. After boundary scan testing is completed, set the JTAG engine to TEST LOGIC RESET. At this point, the boundary scan registers no longer control the I/Os. All the I/O’s not used for configuration go into 3-state mode again, just as they were before the boundary scan test.
3. Toggle the $\overline{\text{PROGRAM}}$ pin to high to start initialization and configuration.
4. Once the VF1 FPGA is configured, additional JTAG tests may be conducted, provided care is taken to ensure that the $\overline{\text{PROGRAM}}$ pin is held high throughout the testing process.

Master Serial Mode

Master serial mode is most often used to configure VF1 FPGAs automatically on power up (Figure 5). Figure 6 and Table 5 show timing information. In this mode, the VF1 FPGA is connected to a VCM SPROM. On power up, or on command from a host system, the VF1 FPGA reads its configuration program from the VCM SPROM.

Note:

1. *The VCM SPROM RESET polarity must be programmed for OE high and $\overline{\text{RESET}}$ low.*

A simple four-wire interface connects the VF1 FPGA with the VCM SPROM:

- The VF1 DIN pin connects with the VCM SDATA (Serial data) pin.
- The VF1 CCLK pin connects with the VCM SCLK (Serial clock) pin.
- The VF1 DONE pin connects with the VCM $\overline{\text{CS}}$ (Chip select) pin.
- The VF1 $\overline{\text{INIT}}$ pin connects with the VCM OE (Output enable) $\overline{\text{RESET}}$ (Reset) pin.

If the $\overline{\text{PROGRAM}}$ pin is tied to V_{CC}, configuration starts automatically on power up. If $\overline{\text{PROGRAM}}$ is not tied to V_{CC}, the host system must cause a low-to-high transition on the $\overline{\text{PROGRAM}}$ pin to initiate the configuration process as described earlier.

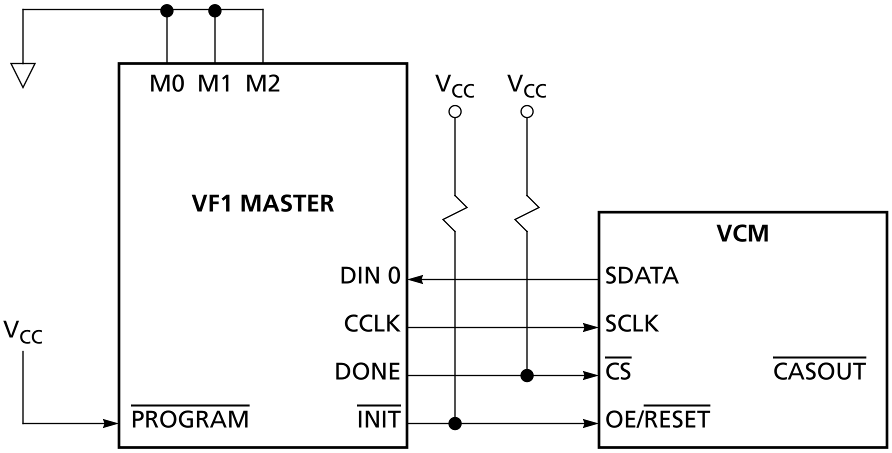


Figure 5. Master Serial Mode

21552B-005

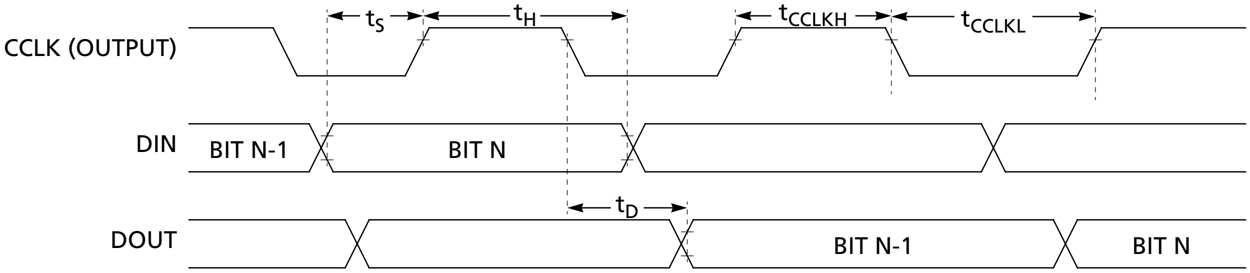


Figure 6. Master Serial Mode Timing Relationships

21552B-006

Table 5. Master Serial Mode Timing Values

Master Serial Mode	Symbol	Min	Max	Unit
DIN Setup Time	t_S	20		ns
DIN Hold Time	t_H	0		ns
CCLK High Time	t_{CCLKH}	33.5	100	ns
CCLK Low Time	t_{CCLKL}	33.5	100	ns
CCLK Period	t_{CCLK}	67	200	ns
CCLK Frequency	f_C	5	15	MHz
CCLK to DOUT	t_D		30	ns

When the $\overline{\text{INIT}}$ pin goes high, the transfer of the configuration bitstream from the VCM SPROM to the VF1 FPGA begins. A low on the VCM $\overline{\text{CS}}$ and a high on its OE/$\overline{\text{RESET}}$ pins cause the VCM to place the first bit of the configuration bitstream on its SDATA pin. The VF1 FPGA reads this bit on the rising edge of the first CCLK clock pulse. The same CCLK pulse signals the VCM to increment its internal address counter and place the second bit on the SDATA pin. This process continues until the configuration process is completed or is interrupted.

Slave Serial Mode

The primary difference between Master mode and Slave mode is the CCLK signal. CCLK is an input for VF1 FPGAs in Slave mode while it is an output in Master mode. Slave serial mode is used to configure VF1 FPGAs in daisy chains that are headed by Master devices (Figure 7), or when directly configured in serial mode by a host system (Figure 9). They may also follow Asynchronous Peripheral Mode devices (Figure 12) or Synchronous Peripheral Mode devices (Figure 10) in daisy chains. Figure 8 and Table 6 show Slave mode timing.

The Master-driven Slave serial mode is covered first, followed by Host-driven Slave serial mode. Other uses are covered in subsequent sections.

Master-Driven Slave Serial Mode

Figure 7 shows three VF1 FPGAs reading their configuration data from two SPROMs. The first VF1 FPGA in the chain is configured in Master mode while the second and third devices are configured in Slave mode.

The overall function of Master-driven slave serial mode is straightforward. The Master device reads the configuration bitstream from the SPROMs and configures itself. Once it is configured, it puts overflow configuration data on its DOUT pin where it is read by the first Slave VF1 FPGA. The first Slave device configures itself, and then puts overflow configuration data on its DOUT pin where it is read by the second Slave device. This process continues until the Master and all Slave devices in a chain have been configured. The following paragraphs describe the process in more detail.

Note:

Two SPROMs are shown in Figure 7. A single VCM (Vantis Configuration Memory) SPROM with a 1 Mbit memory capacity can hold the configuration programs for two VF1025 or VF1020 devices, four VF1012 devices, or one VF1036 device, as explained in the Vantis Configuration Memory data sheet. When the configuration program exceeds the capacity of one VCM SPROM, two or more may be cascaded to provide the necessary storage capacity. Refer to the Vantis Configuration Memory Data Sheet for details on how to cascade VCM SPROM devices.

The interface between the Master VF1 FPGA and the SPROMs is identical to that described in Master serial mode, above. The Master VF1 controls the configuration process by controlling the CCLK configuration clock

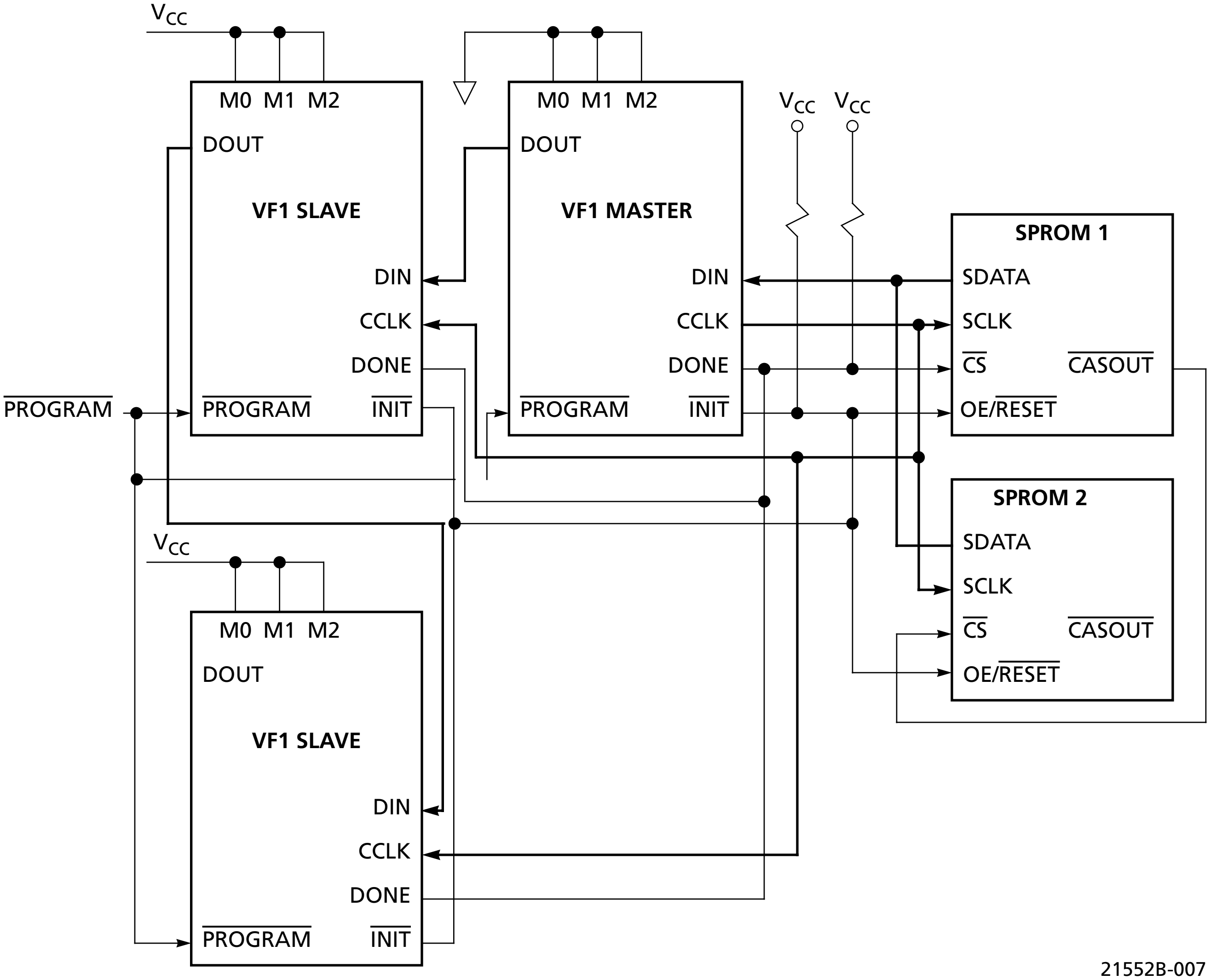


Figure 7. Slave Serial Mode, Master-Driven

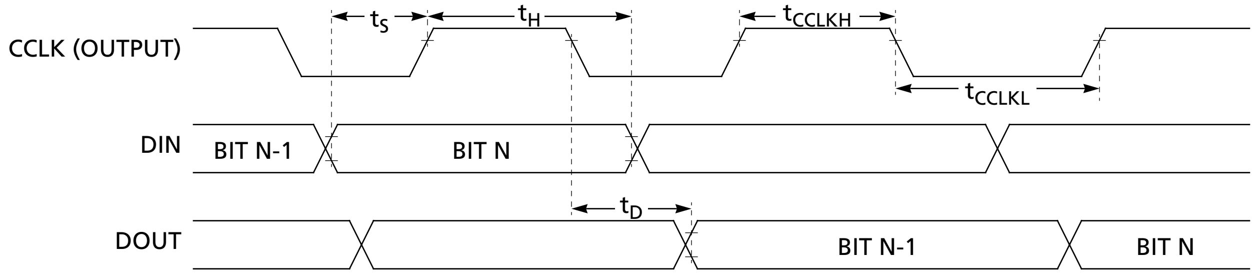


Figure 8. Slave Serial Mode Timing Relationships

21552B-008

Table 6. Slave Serial Mode Timing Values

Slave Serial Mode	Symbol	Min	Max	Unit
DIN Setup Time	t_S	20		ns
DIN Hold Time	t_H	0		ns
CCLK High Time	t_{CCLKH}	50		ns
CCLK Low Time	t_{CCLKL}	50		ns
CCLK Period	t_{CCLK}	100		ns
CCLK Frequency	f_C		10.0	MHz
CCLK to DOUT	t_D		30	ns

The DONE and $\overline{\text{INIT}}$ signals of all three devices are tied together to ensure that the configuration process does not begin until all VF1 FPGAs have completed the initialization process. The $\overline{\text{INIT}}$ signal is applied to the OE/$\overline{\text{RESET}}$ input of both SPROMs, while the DONE signal is applied to the $\overline{\text{CS}}$ input of the first SPROM only. The first SPROM's $\overline{\text{CASOUT}}$ (Cascade out) signal is applied to the $\overline{\text{CS}}$ input of the second SPROM.

The CCLK clock output from the Master VF1 is applied to the SCLK inputs of both SPROMs and to the CCLK pins of the Slave VF1 FPGAs.

The SDATA outputs of both SPROMs are tied together to provide a common DIN data input to the Master VF1 FPGA. The DOUT output pin of the Master device is applied to the DIN of the first Slave device, and the DOUT output of the first Slave is applied to the DIN input of the second Slave device. The same connections are used for any subsequent Slave devices in the chain.

When the INIT signal goes high, configuration proceeds as described in Master serial mode, with the following additions and exceptions:

1. The bit count that the Master device reads early in the configuration cycle is a count of the complete bitstream, including the configuration programs for all VF1 FPGAs in the chain.
2. All VF1 FPGAs receive every CCLK clock pulse and begin counting pulses with the first pulse received. The pulse count, therefore, will be the same in each VF1 FPGA throughout the configuration cycle.
3. The Master VF1 FPGA reads the header bytes and configuration bit count from the SPROMs, stores the bit count, and passes the information on to the first Slave VF1 FPGA via the Master's DOUT pin. The header and bit count are delayed by 1.5 clock cycles as they pass through the Master VF1 FPGA.
4. Once the header and bit count have been transmitted, the Master VF1's DOUT pin goes high and remains high until it is ready to transmit configuration data to the first Slave VF1 FPGA.
5. The first Slave VF1 FPGA reads the header bytes and bit count, stores the bit count in its internal counter, and passes the same header and bit count on to the next Slave VF1 FPGA. The Header and bit count are delayed by an additional 0.5 clock cycle as they pass through the first Slave VF1 FPGA. Once the header and bit count have been transmitted, the first Slave VF1's DOUT pin goes high and remains high until it is ready to transmit configuration data to the second Slave VF1 FPGA.
6. The second and subsequent Slave VF1 FPGAs read the header and bit count and store the bit count in their internal counters.
7. Following receipt of the header and bit count, the Master device receives configuration frames from the SPROM, performs a CRC check on each frame, and uses these frames to configure itself.
8. Once the Master VF1 is fully configured, it sends subsequent configuration frames to the first Slave device via the Master's DOUT pin. The start bit of the first configuration frame that the Slave device sees causes the Slave device to receive the frame, perform CRC checks, and use the frame data to configure itself. The Slave device continues to receive and store frames until it is fully configured.
9. When a Slave device is fully configured, it sends subsequent configuration data to down-stream Slave devices via its DOUT pin. This process continues until all Slave devices have been configured.

When the final Slave device is fully configured, the bit counter in each VF1 FPGA should match the bit count that was received from the SPROM early in the bitstream. When the counts match, all VF1 FPGAs will allow their DONE outputs to go high, thus signaling the end of the configuration process and removing the CS signal from the SPROMs. At the same time, the VF1 FPGA I/O pins are enabled and the FPGA begins to perform the functions that are defined by the configuration bitstream.

Error Detection

Each VF1 FPGA checks incoming configuration frames for errors as it receives them. If any VF1 FPGA detects a bitstream error, it pulls its INIT signal low, terminating the configuration process. Just as in Master serial mode, a pulse on the PROGRAM pins of all devices restarts the configuration process.

Application Notes

Host-Driven Slave Serial Mode

A system designer can eliminate configuration SPROMs from a system by loading VF1 FPGAs from a host system (Figure 9). Configuring VF1 FPGAs from a host system allows the host to precisely control the configuration process, and to reconfigure the VF1 FPGAs in the system at any time.

The configuration diagram shown in Figure 9 is similar to the Master-driven slave serial mode described above, with these differences:

1. The SPROMs are replaced by a host system.
2. All VF1 FPGAs are configured in Slave mode, with no Master mode device in the chain.
3. The host system provides the configuration clock for all VF1 FPGAs in the chain.
4. The host system controls the $\overline{\text{PROGRAM}}$ signal.

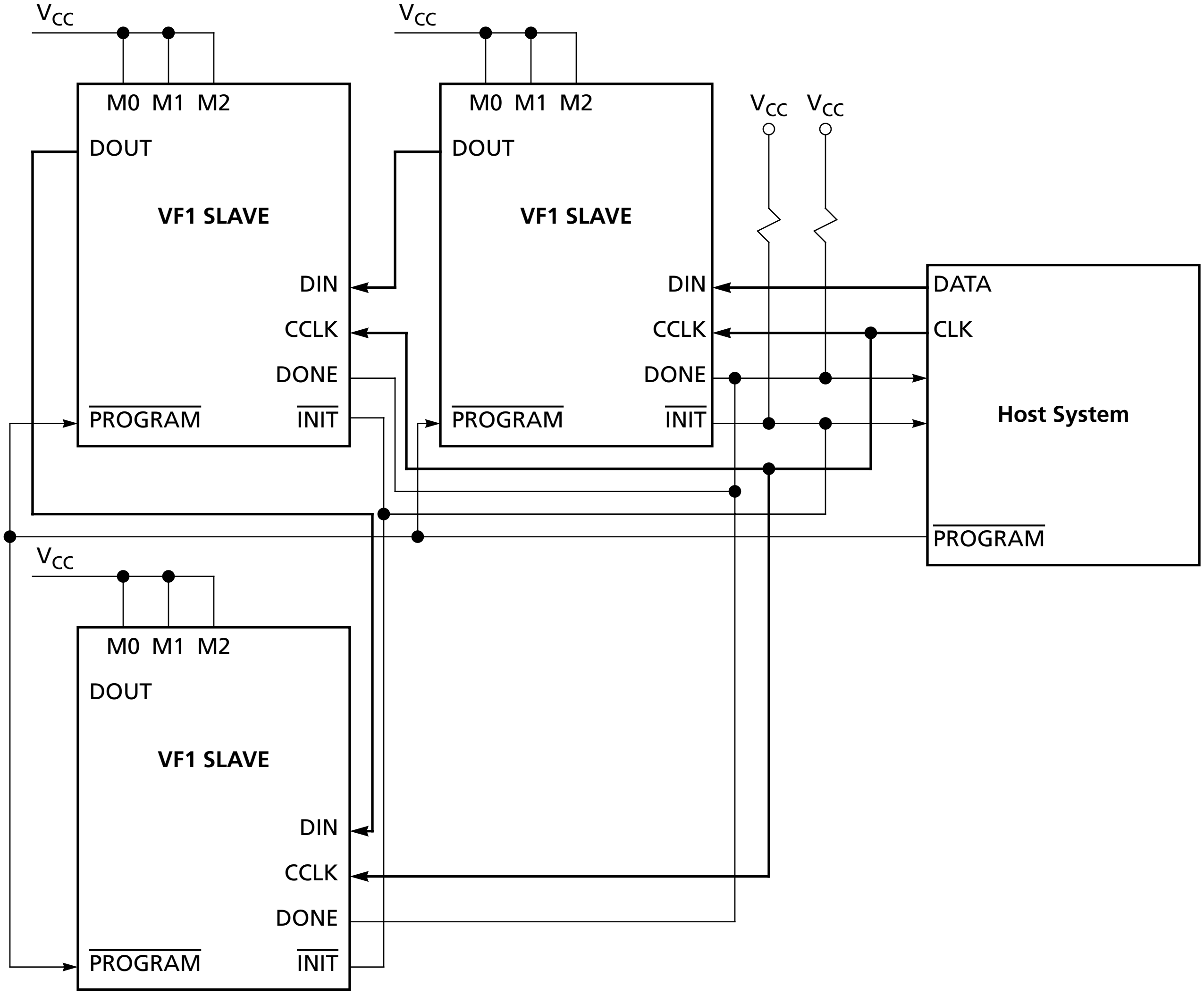


Figure 9. Slave Serial Mode, Host-Driven

21552B-009

The configuration process is similar to the Master-driven Slave serial mode with these exceptions:

1. The host system initiates configuration by the by pulsing the PROGRAM signal low. Even on power up, the VF1 FPGAs waits for the host system to initiate configuration.
2. The VF1 DONE and INIT signals provide status information to the host system. Host systems may also monitor the VF1 HDC (High during configuration) and LDC (Low during configuration) status signals during the configuration process, but the states of HDC and LDC are undefined following configuration.

Byte-Wide Data Transfers

Two configuration modes, Synchronous Peripheral mode and Asynchronous Peripheral mode, support byte-wide data transfers between a host system and VF1 FPGA.

When a VF1 FPGA receives a byte of configuration data, it serializes the byte and processes it as if it were received serially. Byte-wide transfers, therefore, offer another design alternative but not a means of speeding up the configuration process.

Synchronous Peripheral Mode

In Synchronous Peripheral mode, the host system sends byte-wide configuration data and the configuration clock (CCLK) to the VF1 FPGA. The VF1 FPGA receives the byte-wide data on its DIN 0-7 pins. If there are two or more VF1 FPGAs in a daisy chain, the first is configured in Synchronous Peripheral mode and all subsequent devices in Slave mode (Figure 10). Figure 11 and Table 7 show timing information.

Configuration proceeds as follows:

1. The host system initiates the configuration process by pulsing the VF1 PROGRAM pin low.
2. The first data byte is clocked into the VF1 FPGA on the rising edge of the second CCLK pulse after INIT goes high (Figure 11). Bytes are then clocked in on every eighth CCLK pulse.
3. The VF1 RDY/BUSY signal acknowledges the loading of the byte by going high for one CCLK period on the same clock that loaded the byte.
4. To complete the shifting of the last byte into configuration memory, the host system must continue providing CCLK pulses after the last byte is loaded. One way to do this is to continue providing CCLK pulses until the DONE pin goes high.
5. In daisy-chain configurations, the first VF1 FPGA in the chain loads itself and then presents serial data on its DOUT pin. The data appears on DOUT 1.5 CCLK cycles after it is loaded in parallel, which means that DOUT changes on a falling CCLK edge and the next VF1 FPGA loads data on the next rising edge.

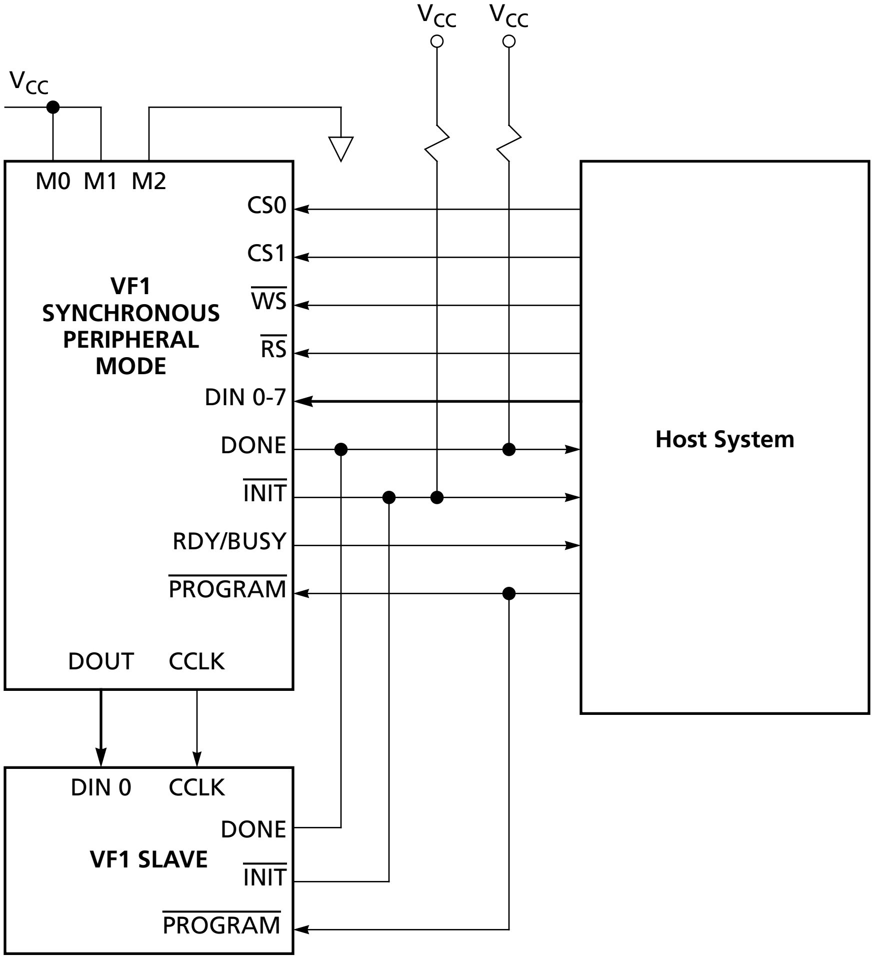


Figure 10. Synchronous Peripheral Mode

21552B-010

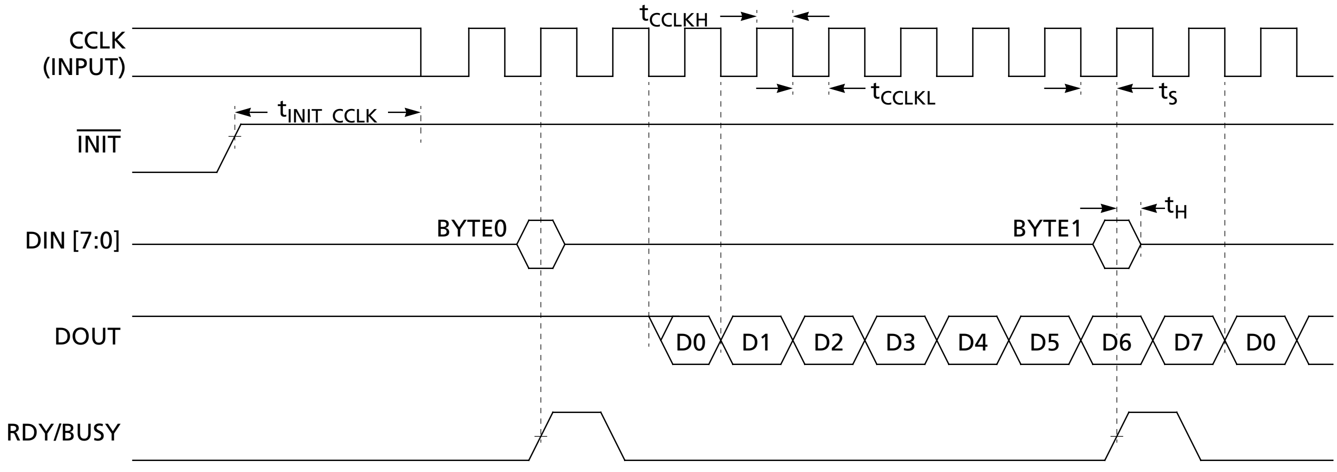


Figure 11. Synchronous Peripheral Mode Timing Relationships

21552B-011

Table 7. Synchronous Peripheral Mode Timing Values

Synchronous Peripheral Mode (Figure D)	Symbol	Min	Max	Unit
DIN Setup Time	t_S	20		ns
DIN Hold Time	t_H	0		ns
CCLK High Time	t_{CCLKH}	50		ns
CCLK Low Time	t_{CCLKL}	50		ns
CCLK Period	t_{CCLK}	100		ns
CCLK Frequency	f_C		10.0	MHz
CCLK to DOUT	t_D		30	ns

Asynchronous Peripheral Mode

In Asynchronous peripheral mode, a host system sends a byte of configuration data to a VF1 FPGA, or to the lead VF1 FPGA in a daisy chain (Figure 12). The VF1 FPGA receives the byte and generates its own clock to serialize the configuration data. The VF1 RDY/$\overline{\text{BUSY}}$ signal provides handshaking between the host system and the VF1 FPGA.

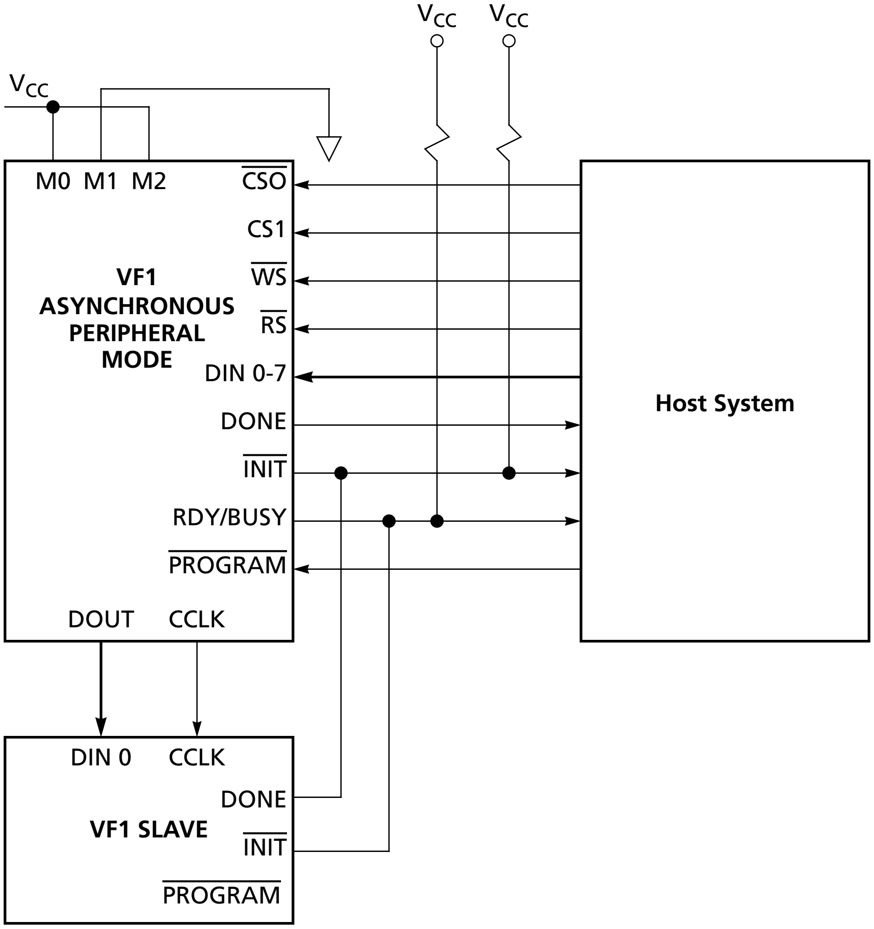
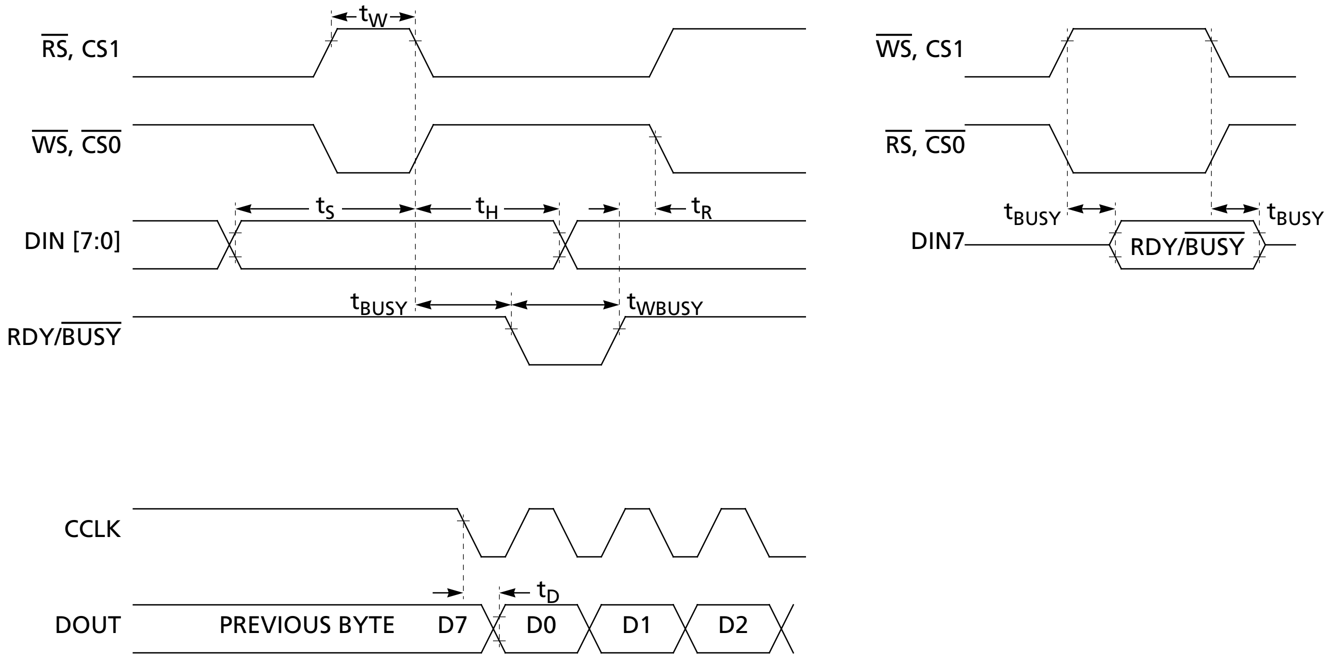


Figure 12. Asynchronous Peripheral Mode

21552B-012



$\overline{\text{RS}}$, CS1
$\overline{\text{WS}}$, $\overline{\text{CS0}}$
DIN [7:0]
RDY/$\overline{\text{BUSY}}$
t_W
t_S
t_H
t_R
t_{BUSY}
t_{WBUSY}
$\overline{\text{WS}}$, CS1
$\overline{\text{RS}}$, $\overline{\text{CS0}}$
DIN7
t_{BUSY}
t_{BUSY}
RDY/$\overline{\text{BUSY}}$
CCLK
DOUT
PREVIOUS BYTE
D7
D0
D1
D2
t_D

Figure 13. Asynchronous Peripheral Mode Timing Relationships

21552B-013

Table 8. Asynchronous Peripheral Mode Timing Values

Asynchronous Peripheral Mode	Symbol	Min	Max	Unit
$\overline{\text{RS}}$ $\overline{\text{WS}}$ $\overline{\text{CS0}}$ CS1 Pulse Width	t_W	100		ns
DIN[7:0] Setup Time	t_S	20		ns
DIN[7:0] Hold Time	t_H	0		ns
RDY/$\overline{\text{BUSY}}$ Delay	t_{BUSY}		30	ns
RDY/$\overline{\text{BUSY}}$ Low	t_{WBUSY}	2	9	CCLK Periods
Earliest $\overline{\text{WR}}$ After End of Busy	t_R	0		ns
CCLK to DOUT	t_D		30	ns

Four signals control the writing of byte-wide data into the lead VF1 FPGA: $\overline{\text{WS}}$ (write select), $\overline{\text{CS0}}$ (chip select 0), $\overline{\text{RS}}$ (read select), and CS1 (chip select 1). As stated earlier, RDY/$\overline{\text{BUSY}}$ provides handshaking. Figure 12 shows signal connections while Figure 13 and Table 8 show timing relationships:

1. As in all other non-JTAG modes, configuration is initiated by pulsing the VF1 FPGA(s) $\overline{\text{PROGRAM}}$ signal low. The VF1 DONE and $\overline{\text{INIT}}$ signals provide initialization and configuration status information to the host system.
2. When the VF1 FPGA is ready to receive a data byte, its RDY/$\overline{\text{BUSY}}$ signal goes high.
3. The host system places the data byte on the VF1 DIN 0-7 pins, usually via a microprocessor bus. The host selects the VF1 FPGA by setting $\overline{\text{CS0}}$ low, CS1 high, and $\overline{\text{RS}}$ high. The host then pulses

$\overline{\mathrm{WS}}$ low. The VF1 reads the data byte on the rising edge of the $\overline{\mathrm{WS}}$ pulse. The VF1 uses both a holding register and a shift register for data bytes. A byte is received in the holding register and transferred to the shift register when the shift register is empty.

4. When the VF1 reads the byte, the RDY/$\overline{\mathrm{BUSY}}$ signal goes low. The VF1 FPGA transfers the byte from the holding register where it was received into a shift register where the byte is serialized for VF1 configuration.
5. The RDY/$\overline{\mathrm{BUSY}}$ signal goes high again when the VF1 is ready to receive another byte. If the shift register is empty when the holding register is loaded, the byte is transferred on the next clock and the RDY/$\overline{\mathrm{BUSY}}$ signal goes high immediately. If the shift register is not empty, the RDY/$\overline{\mathrm{BUSY}}$ signal stays low until the shift register is empty and the byte waiting in the holding register is transferred.

The designer can eliminate one status line by monitoring the RDY/$\overline{\mathrm{BUSY}}$ signal on the microprocessor bus rather than at the RDY/$\overline{\mathrm{BUSY}}$ output on the VF1 FPGA. Enabling the chip selects ($\overline{\mathrm{CS0}}$ low and CS1 high), and selecting read mode ($\overline{\mathrm{WS}}$ high and $\overline{\mathrm{RS}}$ low) places the RDY/$\overline{\mathrm{BUSY}}$ status on the VF1 DIN7 pin.

JTAG Mode

In JTAG mode, VF1 FPGAs are configured via the JTAG pins TCLK, TMS, TDI, and TDO (Figure 14). Figure 15 and Table 9 show timing information. Three instructions in addition to the standard JTAG instructions support JTAG the configuration mode:

- PROG_MODE. This instruction places the VF1 FPGA in programming mode. Scan cells control the VF1 I/Os during this instruction.
- PROGRAM. Once the VF1 FPGA is in programming mode, the PROGRAM instruction shifts configuration data into the VF1 FPGA. Scan cells control VF1 I/Os.
- VERIFY. After configuration this instruction is used to read back all configuration, VGB and I/O flip-flops, and embedded SRAM bits in the device.

A host system such as a microprocessor controls the configuration of the VF1 FPGA or devices and supplies configuration data. The host also provides the configuration clock TCLK.

If two or more VF1 FPGAs are to be configured, they are arranged in a daisy chain. Data is applied to the TDI pin of the first device and the TDO pin of that device is connected to the TDI pin of the next device. The TMS and TCLK signals from the host are applied to all VF1 FPGAs in parallel.

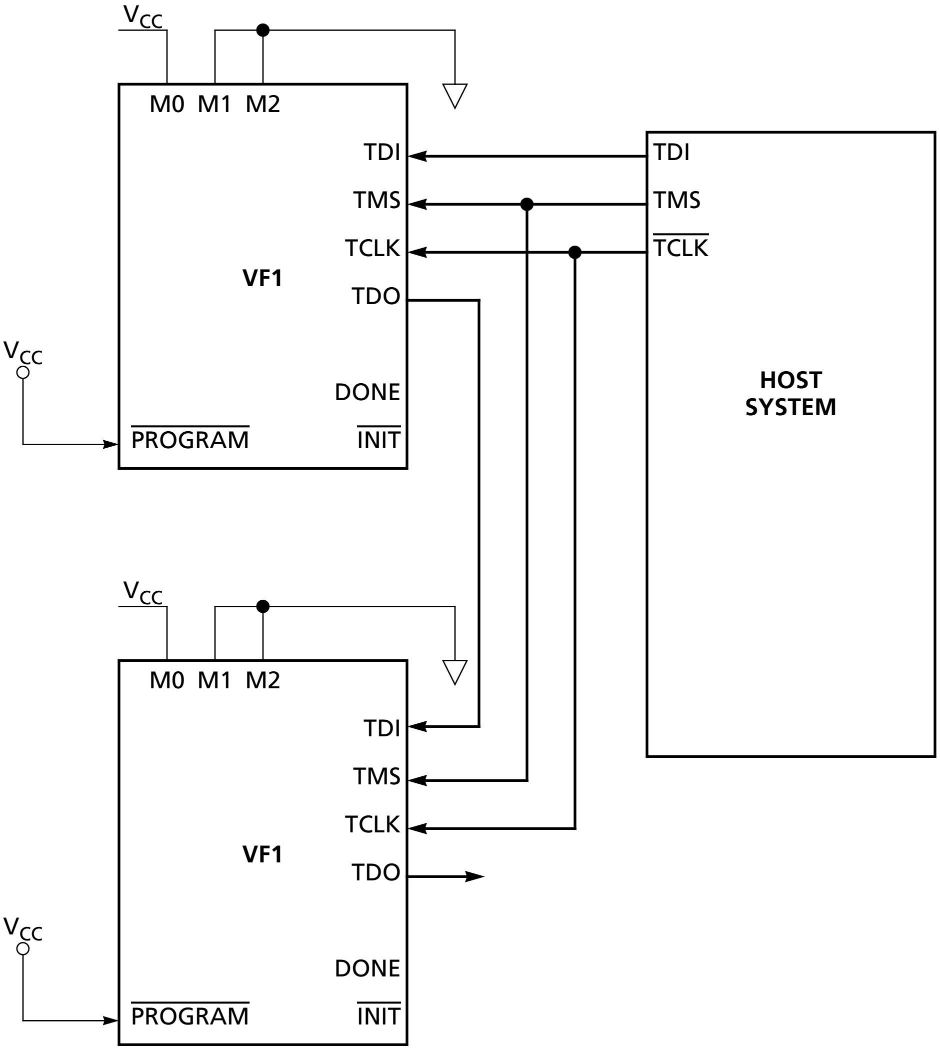


Figure 14. JTAG Mode

21552B-014

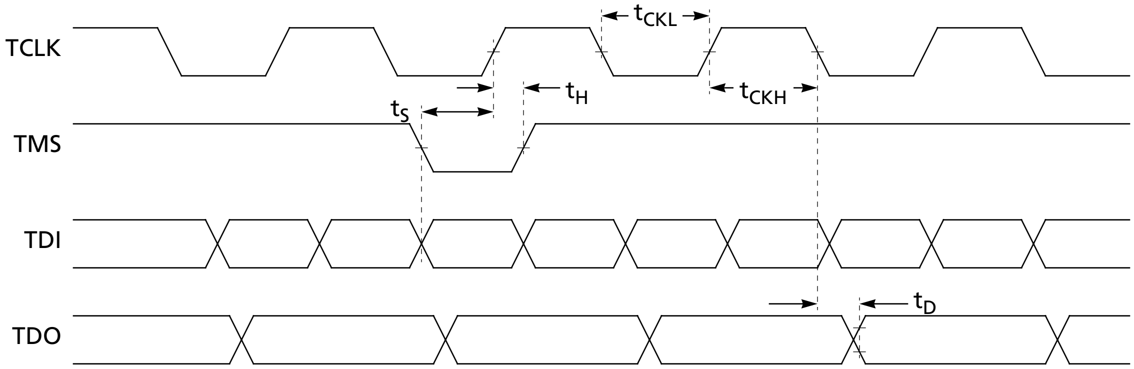


Figure 15. JTAG Mode Timing Relationships

21552B-015

Table 9. JTAG Mode Timing Values

JTAG Mode	Symbol	Min	Max	Unit
TDI/TMS to TCK Setup Time	t_S	40		ns
TDI/TMS Hold Time from TCK	t_H	40		ns
TCK Low Time	t_{CKL}	50		ns
TCK High Time	t_{CKH}	50		ns
TCK to TDO Delay	t_D		30	ns
TCK Frequency	t_{CKF}		10.0	MHz

DONE and $\overline{\text{INIT}}$

During non-JTAG configuration the $\overline{\text{INIT}}$ pin is an open-drain status pin that can be used to delay the start of configuration when it is driven low. In JTAG mode, however, this pin may be driven low or HIZ by the boundary scan register when executing PROG_MODE or PROGRAM instructions and override the non-JTAG functions of the pin.

To avoid having the boundary scan settings interfere with configuration, driving the $\overline{\text{INIT}}$ pin low in JTAG mode has no effect on holding off configuration. Therefore, the rules for using DONE and $\overline{\text{INIT}}$ when configuring two or more VF1 FPGAs are different in JTAG mode than in non-JTAG modes:

- Tying $\overline{\text{INIT}}$ pins together during JTAG configuration mode is not allowed.
- Tying DONE pins together in JTAG configuration mode is not allowed.

Configuration

The JTAG state machine automatically goes to TEST LOGIC RESET upon power up. Execution of JTAG configuration instructions can start after a time delay of t_{PO}+t_{IL} (see Figure 4) when V_{CC} reaches 2.7 volts. This time delay period guarantees that the configuration memory is initialized. JTAG configuration uses the following instruction flow and pin settings:

1. Settings during power-on reset are $\overline{\text{PROGRAM}}$ = High. M0=1, M1=0, M2=0. VF1 I/O pins will come up in 3-state mode except $\overline{\text{INIT}}$ and DONE.
2. After V_{CC}=2.7 volts, wait until Tpo+Til time limit has passed.
3. Apply SAMPLE/PRELOAD instruction to load the UPDATE registers in the boundary scan cells.
4. Apply PROG_MODE instruction. At this point, VF1 I/O pins, including $\overline{\text{INIT}}$ and DONE, are controlled by boundary scan cells.
5. Apply PROGRAM instruction. At this point, VF1 I/O pins, including $\overline{\text{INIT}}$ and DONE, are controlled by boundary scan cells.
6. Go to RUN TEST IDLE. Wait 1 msec.
7. Shift in configuration data by applying TCLK clocks. The total number of t_{CLKs} = Total number of bits in the bitstream. Following the shifting in of all configuration data, VF1 cells will be in the modes defined by the configuration program.
8. Exit PROGRAM instruction and go to TEST LOGIC RESET.
9. When the first VF1 FPGA is configured, put it in BYPASS. If a second VF1 FPGA is to be configured, put any intermediate system devices in BYPASS and configure the second VF1 FPGA.

Put the second VF1 FPGA in BYPASS. Continue this process until all VF1 FPGAs have been configured.

10.The DONE pin may optionally be sampled to confirm successful configuration. If DONE is high, configuration was successful. If DONE is low, an error occurred or the device was not completely configured. Configuration must be repeated.

Vantis Programming Software

Vantis offers VantisPRO™ programming software that provides for the configuration, reconfiguration and readback of VF1 FPGAs using the JTAG mode. Given the configuration of the JTAG chain the VF1 FPGA is in, and the name of the bitstream file, the software can configure the VF1 FPGA through the parallel port on a personal computer. Additionally, the software can also be used to generate the files necessary to configure the VF1 FPGAs using automated test equipment such as full-scale board test systems or JTAG test equipment. VantisPro can generate configuration files for the HP3070, Teradyne test systems and Genrad test systems. Also, it generates the Serial Vector Format (SVF) files commonly used by JTAG test equipment software. The SVF files contain a simple instruction set that tells the hardware the state the JTAG state machine should be in and the data that should be shifted into or out of the part(s).

Vantis also offers a version of VantisPRO software that can be used by an embedded microprocessor to configure the devices in the JTAG mode. This software uses a compressed form of the SVF file generated by VantisPRO software to configure or readback a device. Information about VantisPRO software can be found in the separate document entitled, *MACH ISP Manual.*

Readback

The VF1 readback capability allows designers to read the configuration program back from the VF1 FPGA and compare it with the program that was written into the device. During configuration, a VF1 FPGA is set for one of three possible readback options:

- **Read Disable** (the default state). No readbacks are allowed.
- **Read Once.** One readback is allowed, then readbacks are disabled.
- **Read on Command**. Unlimited readbacks are allowed.

The readback selection is made when the DesignDirect software prepares the configuration pattern. During design development, the Read on Command option is normally used. In production systems, the Read Once option may be used if a host system will verify configuration, or the Read Disable option may be used if the VF1 configuration error detection logic is deemed adequate for reliable operation.

Readback Data

During readback, configuration data is read along with the data in the VF1 internal registers so that the Readback bitstream is not identical to the original bitstream in those particular locations reserved for the registers' internal nodes. In addition, the fill bits, preamble, length counter, and postamble are not read. DesignDirect software provides a mask that allows the readback data to be compared with the original bitstream, or to allow the content of internal registers to be separated from the configuration bitstream for analysis.

The VF1 family provides two readback methods:

- **JTAG readback.** Configuration data is read via the JTAG port using JTAG commands (Figure 14). This method is available following either JTAG or non-JTAG configuration. JTAG readback can be accomplished using the Vantis programming software described earlier.
- **Non-JTAG readback.** Configuration data is enabled with the RTRIG pin and read from the RDO pin (Figure 16. Figure 17 and Table 10 show readback timing information. This readback method is enabled only after non-JTAG configuration.

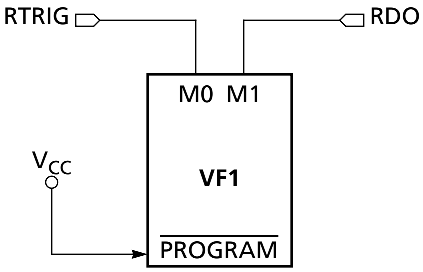


Figure 16. Non-JTAG

21552B-016

Non-JTAG Readback

Non-JTAG readback is enabled following any non-JTAG configuration modes (Master Serial, Slave Serial, Asynchronous Peripheral, and Synchronous Peripheral), provided one of a readback option was selected when the configuration bitstream was generated.

- Apply a low-to-high edge on RTRIG pin and hold RTRIG high.
- Apply four dummy clocks to the CCLK pin.
- Subsequent CCLKs clock out configuration data on RDO. The number of clocks required is: 4 dummy CCLKs + [1 CCLK + Total # of bits in a Frame + 3 CCLKs] times the number of Frames.
- Non-JTAG Readback can be halted by setting RTRIG pin low and after three positive CCLK edges.
- Start bits and stop bits are read as “Don’t cares.”

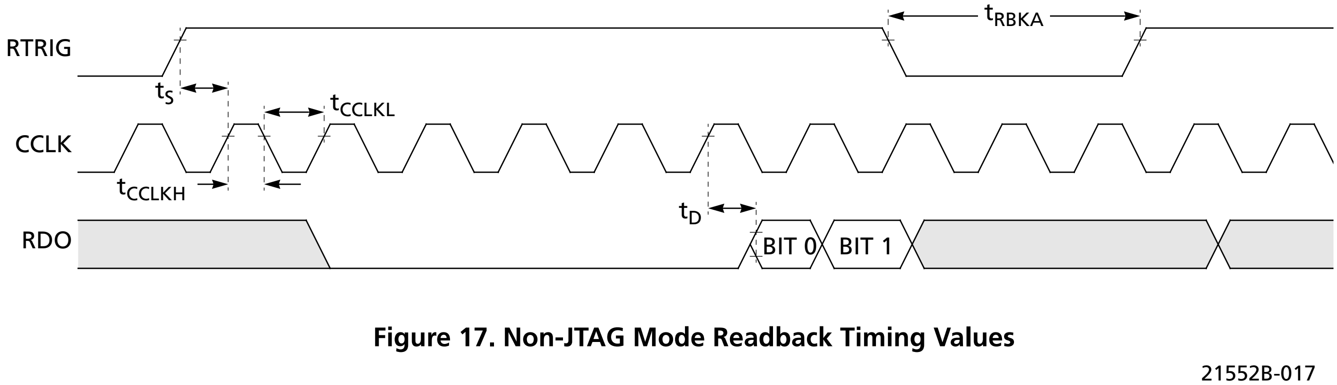


Figure 17. Non-JTAG Mode Readback Timing Values

21552B-017

Table 10. Non-JTAG Readback Timing Values

Readback	Symbol	Min	Max	Unit
RTRIG to CCLK Setup Time	t_S	40		ns
RTRIG Low Width to Abort Readback	t_{RBKA}	3		CCLK
CCLK Low Time	t_{CCLKL}	50		ns
CCLK High Time	t_{CCLKH}	50		ns
CCLK Frequency	f_C		10.0	MHz
CCLK to RDO Delay	t_D		30	ns

APPENDIX A

Configuration Signals

Table 11 describes the functions of all configuration signals. Table 12 summarizes the signals used in each configuration mode. The individual mode descriptions that follow describe how the signals are used in each mode.

Table 11. Configuration Signal Descriptions

M0/RTRIG **M1/RDO** **M2**	**Three multiplexed I/O pins that select the configuration mode.** During configuration, these pins are input pins and are sampled right after initialization to determine the configuration mode. Once configuration is complete, M0 and M1 can be used as RTRIG (Read trigger) and RDO (Read data out) for non-JTAG read-back.
$\overline{\text{PROGRAM}}$	**A dedicated input pin that initiates configuration.** A low level clears the configuration memory and puts the VF1 FPGA into a WAIT state. The MODE pins are sampled. A low-to-high transition clears the configuration memory once more and starts the configuration process. If this pin is high during power up, the device will skip the WAIT state after clearing the configuration memory and will go directly into configuration mode.
$\overline{\text{INIT}}$	**A multiplexed, open-drain I/O pin that indicates initialization status.** A low $\overline{\text{INIT}}$ when $\overline{\text{PROGRAM}}$ is high indicates initialization is not complete and the device is not ready to receive data for configuration. A high $\overline{\text{INIT}}$ (when DONE is low) indicates that initialization is complete and no configuration bitstream errors have occurred. For non-JTAG configuration modes, holding the $\overline{\text{INIT}}$ pin low externally will delay configuration.
DONE	**A dedicated open drain pin that signals when configuration is done.** A low output indicates the VF1 FPGA is in configuration mode. A high output indicates configuration is done and all the I/Os are enabled for normal operation. For non-JTAG configuration modes, enabling of all the I/Os in different devices can be synchronized by tying all the DONE pins together. Enabling the synchronization capability is the default condition, but can be disabled by the configuration bitstream.
CCLK	**A dedicated I/O pin for configuration clock input or output.** In the Master and Asynchronous peripheral modes, this pin is the clock output from an internal oscillator. In the Slave mode and Synchronous Peripheral mode, this pin receives a clock from the Master VF1 FPGA or from a host source. In Master serial mode, this signal clocks data from a companion SPROM such as a VCM SPROM.
DOUT	**A multiplexed I/O pin to pass configuration data from the first VF1 FPGA in a chain to subsequent devices.** During configuration, this is an output pin for sending overflow configuration data to daisy-chained devices.
DIN[7:0]	**Seven multiplexed I/O pins for byte-wide data input.** During Synchronous and Asynchronous Peripheral modes, these input pins receive parallel configuration data. During non-JTAG serial modes, DIN0 functions as the data in (DIN) pin.
RDY/$\overline{\text{BUSY}}$	**A multiplexed I/O Ready/$\overline{\text{Busy}}$ status pin.** This pin indicates when it is appropriate to write another byte of configuration data into the VF1 FPGA during Peripheral mode configuration.
TDI, TCLK, TMS, TDO	**TDI, TCLK, and TMS are dedicated input pins; TDO is a dedicated output pin.** These pins are used for JTAG boundary scan functions and for programming VH1 devices in JTAG mode.
$\overline{\text{CS0}}$, CS1, $\overline{\text{WS}}$, $\overline{\text{RS}}$	**Multiplexed I/O pins used in Asynchronous Peripheral mode.** These four pins are used for controlling configuration data entry in Asynchronous Peripheral mode. A write cycle is initiated by simultaneously asserting $\overline{\text{CS0}}$, CS1, and $\overline{\text{WS}}$, and deasserting $\overline{\text{RS}}$. A low to high transition on $\overline{\text{CS0}}$ or $\overline{\text{WS}}$ or a high to low transition on CS1 or $\overline{\text{RS}}$ loads the data on DIN[7:0] into the VF1. A low on $\overline{\text{RS}}$ and high on $\overline{\text{WS}}$ while $\overline{\text{CS0}}$ and CS1 are asserted changes DIN7 into a status pin that outputs the same signal as the RDY/BUSY pin.
HDC	**A multiplexed I/O status pin that is High During Configuration.**
$\overline{\text{LDC}}$	**A multiplexed I/O status pin that is Low During Configuration.**

Table 12. Signals Used in Each Configuration Mode

Master Serial	Slave Serial	Synchronous Peripheral	Asynchronous Peripheral	JTAG	User Operation
M0 (I)	M0 (I)	M0 (I)	M0 (I)	M0 (I)	(I/O)/RTRIG
M1 (I)	M1 (I)	M1 (I)	M1 (I)	M1 (1)	(I/O)/RDO
M2 (I)	M2 (I)	M2 (I)	M2 (I)	M2 (I)	(I/O)
$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)
$\overline{\text{INIT}}$ (OD)	$\overline{\text{INIT}}$ (OD)	$\overline{\text{INIT}}$ (OD)	$\overline{\text{INIT}}$ (OD)	$\overline{\text{INIT}}$ (OD)	(I/O)
DONE (OD)	DONE (OD)	DONE (OD)	DONE (OD)	DONE (OD)	DONE (OD)
HDC (O)	HLC (O)	HLC (O)	HLC (O)		(I/O)
$\overline{\text{LDC}}$ (O)	$\overline{\text{LDC}}$ (O)	$\overline{\text{LDC}}$ (O)	$\overline{\text{LDC}}$ (O)		(I/O)
CCLK (O)	CCLK (I)	CCLK (I)	CCLK (O)		CCLK (I)
TDI (I)	TDI (I)	TDI (I)	TDI (I)	TDI (I)	TDI (I)
TCLK (I)	TCLK (I)	TCLK (I)	TCLK (I)	TCLK (I)	TCLK (I)
TMS (I)	TMS (I)	TMS (I)	TMS (I)	TMS (I)	TMS (I)
TDO (O)	TDO (O)	TDO (O)	TDO (O)	TDO (O)	TDO (O)
DOUT (O)	DOUT (O)	DOUT (O)	DOUT (O)		(I/O)
DIN0 (I)	DIN0 (I)	DIN0 (I)	DIN0 (I)		(I/O)
		DIN1 (I)	DIN1 (I)		(I/O)
		DIN2 (I)	DIN2 (I)		(I/O)
		DIN3 (I)	DIN3 (I)		(I/O)
		DIN4 (I)	DIN4 (I)		(I/O)
		DIN5 (I)	DIN5 (I)		(I/O)
		DIN6 (I)	DIN6 (I)		(I/O)
		DIN7 (I)	DIN7 (I)		(I/O)
		RDY/BUSY(O)	RDY/BUSY(O)		(I/O)
			$\overline{\text{CS0}}$ (I)		(I/O)
			CS1 (I)		(I/O)
			$\overline{\text{WS}}$ (I)		(I/O)
			$\overline{\text{RS}}$ (I)		(I/O)

Note:

I = Input

O = Output

OD = Open Drain

I/O = Input/Output