Configuring Delta39K™/Quantum38K™ CPLDs

Overview

This application note discusses the configuration interfaces, modes, and processes of the Delta39K™ and Quantum38K™ CPLDs and includes examples of device set-up.

Each member of the Delta39K family is available in volatile and Self-Boot™ packages. An external CPLD Boot EEPROM is used to store the configuration data for the volatile package while an internal Flash memory is embedded in the Self-Boot package. Programming is defined as the loading of a user's design into either the external CPLD Boot EEPROM or the internal Flash device. Configuration, on the other hand, is the loading of a user's design into the volatile die.

The external EEPROM configures the Delta39K through the Serial Configuration port. Both the volatile and Self-Boot packages can also be configured through the JTAG port. The configuration process with the internal Flash is transparent to the user.

The Quantum38K family is architecturally similar to Delta39K but without some of the Delta39K's features. It is also only available in volatile packages. The configuration set-up and processes of the Quantum38K family are the same as those of the volatile Delta39K.

Configuration and Programming Paths

Delta39K devices can be configured in multiple ways (*Figure 1*). The bitstream can be sent through the JTAG port from a PC, through the Serial Configuration port from an external EEPROM, or through the Non-Volatile (NV) port from the internal Flash. For volatile Delta39K and Quantum38K, configuration through the internal Flash is obviously not available.

To program the internal Flash, the bitstream is sent from a PC through the JTAG port into the Delta39K, and is then passed on to the internal Flash though the NV port (*Figure 2*).

The configuration bitstream is in a compressed form when it is stored in an external EEPROM or in the internal Flash. The bitstream is then uncompressed by internal circuitry during configuration.

Configuration Interfaces

Delta39K/Quantum38K has three common configuration signals, a JTAG port, and a Configuration port. Delta39K also has the NV port to interface with the internal Flash. The three common configuration signals are MSEL, Reconfig, and Config_Done.

Common Configuration Signals

MSEL - *Input* - Selects the configuration mode when starting the configuration process. It will configure from the internal Flash when set to LOW/GND and from an external EEPROM when set to HIGH/V_{CCCNFG}. The MSEL level needs to be stable during the configuration process. For Self-Boot devices, MSEL should be connected to GND, as shown in *Figure 6*. For volatile devices, MSEL should be connected to V_{CCCNFG} through a pull-up resistor as shown in *Figure 5*. Note that even if no configuration memory is being used, MSEL must still be connected to a valid logic level and should not be left floating. This applies to all Delta39K devices, including volatile Delta39K devices that will be programmed through JTAG via an embedded processor.

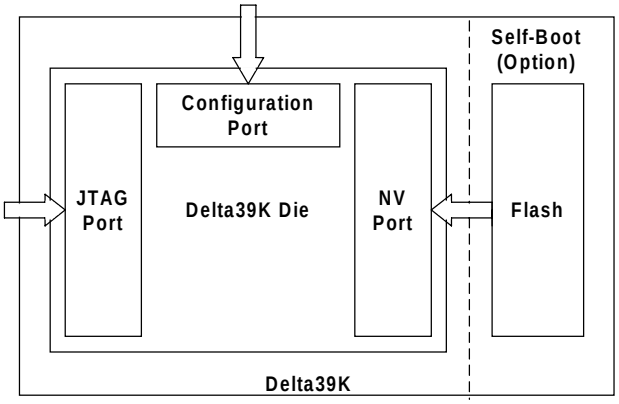


Figure 1. Configuration Paths into the Delta39K™

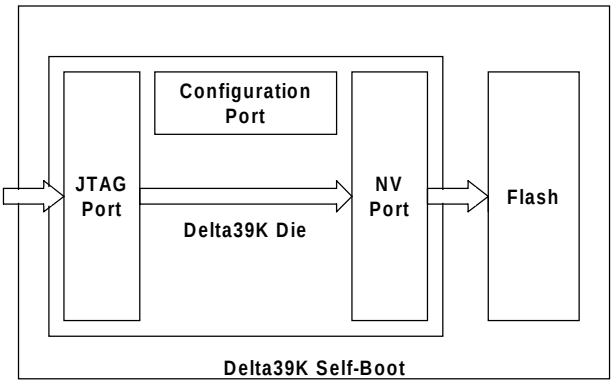


Figure 2. Programming Path into the Internal Flash

Reconfig - *Input* - Initiates reconfiguration of the device when brought from logic level LOW to HIGH. When at a logic level LOW, it suppresses and resets the configuration process. Reconfig must be pulled HIGH for the device's logic to operate ("PAL mode"). The device will not attempt to configure unless Reconfig transitions from LOW to HIGH.

Config_DONE - *Output* - Signal is LOW during the configuration process and then switches to HIGH once the process is completed successfully. This is *not* an open-drain output.

JTAG Port

The JTAG port consists of four signals: TCK, TMS, TDI, and TDO. Data is shifted into the device through TDI and out of the device through TDO. TMS (Test Mode Select) and TCK (Test Clock) are used to control an internal state machine called the Test Access Port (TAP) Controller (Appendix D), which serves as the access point for boundary scan testing, device configuration/verification and non-volatile device programming/verification. For more information on the TAP architecture, refer to IEEE Standard 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture.*

Configuration Port

Four signals, in addition to MSEL and Reconfig, are used to interface with an external EEPROM for configuring the Delta39K:

CCLK - *Output* - Internally generated configuration clock going to the external EEPROM.

CCE - *Output* - A LOW logic level signals the external EEPROM that the Delta39K/Quantum38K is ready to accept configuration data. This signal is used to start and stop serial data shifting into the device.

Reset - *Output* - A HIGH logic level resets the configuration address pointer in the external EEPROM.

DATA - *Input* - A one-bit configuration data input.

Non-Volatile (NV) Port

The NV port is a dedicated port used to interface with the internal Flash configuration memory. It is not directly accessible by the user, but it is indirectly accessible through the JTAG port of a Self-Boot Delta39K.

Power Supplies

There are six types of power supplies in Delta39K to allow independent and flexible configuration. They are V_{CC} (Core), V_{CCIO} (IO Cells), V_{CCPRG} (NV port and internal Flash), V_{CCCNFG} (Configuration port), V_{CCJTAG} (JTAG port), and V_{CCPLL} (PLL block). For PAL (Operation) mode power supply set-up, please refer to the Delta39K/Quantum38K family data sheet.

There is no required power-up sequence of these power supplies for successful configuration. However, all power supplies except IO power supplies have to be up and stable before the device will start configuring. The power supplies are connected to the internal POR circuitry and trigger the POR operation when all power supplies reach at least 1.8V. Even when the design does not use the PLL feature of Delta39K, V_{CCPLL} will still need to be powered. The same rule applies to the other power supplies.

While V_{CCPRG} only needs to be 1.8V for it to trigger the POR circuitry, it needs to be 3.3V for successful operation of the internal Flash memory in Self-Boot Delta39K devices. For Delta39K and Quantum38K volatile devices, V_{CCCNFG} must be supplied by 3.3V, since the CPLD Boot EEPROM is a 3.3V device. V_{CCJTAG} should be supplied according to the I/O interface of the JTAG port: 1.8V for LVCMOS18 operation, 2.5V for LVCMOS2 operation, 3.3V for LVCMOS/LVTTL operation. The I/O standard of the JTAG port is not "configurable"—it is set by the supply voltage. If V_{CCJTAG} is 3.3V, use the C3ISR cable or the UltraISR™ cable. If V_{CCJTAG} is 1.8V or 2.5V, use the C3ISR cable. Note that a chain of JTAG devices must have the same supply voltage.

Configuration Data Files

When compilation and fitting of a design to a Delta39K device is successful, *Warp*® generates a configuration data file in the project directory. The data file is easily recognized by the HEX extension.

The HEX file is compressed configuration data in Intel HEX format. It is used by the ISR software to program the internal Flash and/or configure the volatile die, and is used by the CPLD Boot EEPROM software to configure the external Boot EEPROM.

The external CPLD Boot EEPROM used to store configuration data for the Delta39K/Quantum38K volatile packages is programmed via a two-wire interface through Cypress's CYDH2200E CPLD Boot EEPROM Programming Kit. For more information on how to program the CPLD Boot EEPROM, please refer to the data sheet titled "CYDH2200E CPLD Boot EEPROM Programming Kit." For more information on the architecture and timing specification of the CPLD Boot EEPROM, refer to the data sheet titled "CPLD Boot EEPROM."

For more information on how to use ISR software, please refer to the ISR-related application notes available at Cypress Semiconductor's website (www.cypress.com).

Table 1 shows the uncompressed configuration bitstream length and the corresponding recommended EEPROM. Note that in all cases the Atmel equivalent part (AT17LV) may be used instead of the Cypress part (CY3LV).

Table 1. Uncompressed Configuration Bitstream Length

Device	Bitstream Length	Memory Device Recommendation
Delta39K30 Quantum38K30	751,088	CY3LV512, AT17LV512
Delta39K50 Quantum38K50	751,088	CY3LV512, AT17LV512
Delta39K100 Quantum38K100	1,497,296	CY3LV010, AT17LV010
Delta39K165	3,268,640	CY3LV002, AT17LV002
Delta39K200	3,268,640	CY3LV002, AT17LV002

Note that, in every case, the recommended memory device does not contain enough capacity to store the entire uncompressed bitstream required to configure the Delta39K device. However, remember that the Delta39K external CPLD Boot

EEPROM configuration interface expects a compressed bitstream, and not an uncompressed bitstream. The compression ratio is effective enough to allow the use of a smaller CPLD Boot EEPROM than would be required for the uncompressed bitstream.

Intel HEX Format

The general record format in an Intel HEX file is shown in *Table 4*. Each byte is represented by two ASCII characters and the RECORD MARK is always represented by a colon.

RECORD MARK ':'	RECLEN	LOAD OFFSET	RECTYPE	INFO/ DATA	CHECKSUM
1 char	2 chars	4 chars	2 chars	2n chars	2 chars
	1 byte	2 bytes	1 byte	n bytes	1 byte

Figure 3. Intel HEX Format

RECORD MARK - Always a colon (:), it signals the start of a record line.

RECLEN - The number of bytes, between 0 and 255 bytes, contained in the INFO/DATA field.

LOAD OFFSET - 16-bit offset address for the data byte(s) in the INFO/DATA field.

RECTYPE - Record Type: 00 for DATA, 01 for END OF FILE.

DATA - Contains a pair of ASCII characters to represent each byte of data specified in the RECLEN field.

CHECKSUM - The 2's Complement of the 8-bit addition of the bytes in RECLEN, LOAD OFFSET, RECTYPE, and INFO/DATA.

Example

Below is an example of an Intel HEX file that contains two sets of data. The first set has 4 bytes of data (0x05, 0xFA, 0x39, 0x4D) and the second set has 2 bytes of data (0x55, 0xAA). The third line is the EOF record.

:0400000005FA394D77

:0200040055AAFB

:00000001FF

How To Calculate The Checksum

Add 1 byte of data at a time and negate the end result to give the final checksum value. Below is the calculation for the first data set of the above example.

0x04 + 0x00 + 0x00 + 0x00 = 0x04

0x04 + 0x05 = 0x09

0x09 + 0xFA = 0x03

0x03 + 0x39 = 0x3C

0x3C + 0x4D = 0x89

–(0x89) = 0x77

Warp HEX File Format

The HEX format output by *Warp* for Delta39K configuration is a specific application of the general Intel HEX format.

Table 2. *Warp* HEX File Format

Offset (bytes)	Size (bytes)	Name	Comment
0	4	Signature	This value is used to identify the header
4	4	Length	Length of the header (*not* including signature or this length field)
8	4	Version	Header version (currently 0x1)
12	64	Order Code	Space-padded string containing targeted device order code for this HEX file
76	19	Date	Date the file was created
95	1	DTCB	Debug and Test Configuration Byte
96+ 1026n	1024	Data	Configuration data for CPLD
1120+ 1026n	2	CRC	Checksum to ensure configuration data payload transmission is correct

In *Table 2* above, the equations to determine offset for the last two rows are for n=0,1,2,3...

Debug and Test Configuration Byte (DTCB)

The DTCB contains internal configuration settings as well as the configuration clock wait-state value. The default value of the bits should not be changed except bits 4, 5, and 6 that users may manually modify. Bit 6 is the Even Parity bit of bits 0 through 6.

Table 3. Debug and Test Configuration Byte

Bit	Default	User	Description
7(MSB)	0	N	Reserved
6	0	Y	Even Parity Check
5	0	Y	Wait State 1
4	0	Y	Wait State 0
3	0	N	Reserved
2	0	N	Enables Ring Oscillator
1	0	N	Reserved
0(LSB)	0	N	Enables CRC

Compressed Bitstream Format

The configuration data sent to the Delta39K volatile configuration interface (into the "DATA" input of the Delta39K device) is sent with the least significant bit of the DTCB first.

The compressed bitstream format (the data that is sent to the Delta39K device) follows the following sequence:

1. DTCB (1 Byte of data, sent LSB/bit 0 first)
2. Data (1024 Bytes of data)
3. CRC (2 Bytes of data)

   (Repeat 2 and 3 until all configuration bits are covered)

When the configuration bitstream does not fully utilize the final 1024 bytes of data, all data bits after the last configuration bit will be set to 0.

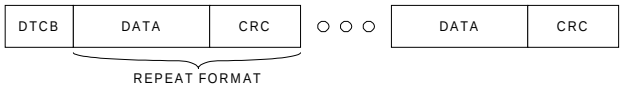


Figure 4. Compressed Bitstream Format

Warp HEX File Example

Following is an example HEX file created by Warp6.2 when a 39200 device is targeted:

```
:020000040000FA
:10000000050219125700000001000000637933391E
:10001000323030763230382D3138316E74632020F2
:10002000202020202020202020202020202020D0
:10003000202020202020202020202020202020C0
:10004000202020202020202020202030392F3167
:10005000382F323030312031383A30343A303700AE
:10006000FFFFFFFFFFFFFFD08FFFFF1BFE80FFE8BA
:1000700047FE3FFFFF83FE827FE37FD0108FFFFFAD
.
.
.
:10AFF00026D83EE04053B43D78A26D83EE04053B75
:10B0000043D78A26D83EE04053B43D78A26D83EE04
:10B0100004053B43D78A26D83EE04053B43C0000A9
:10B0200000000000000000000000000000000020
:10B0300000000000000000000000000000000010
.
.
.
:10B1200000000000000000000000000000000001F
:08B13000000000000CEA4A5
:00000001FF
```

The CPLD Boot EEPROM Programming Kit (CYDH2200E) comes with the Configurator Programming Software (CPS). This software uses the HEX file created by *Warp* to program the CPLD Boot EEPROM. In the example HEX file above, the "data payload" of the HEX file is indicated by the area of the file that is in **bold**.

In the example above, the first line of the HEX file is a "data segment" type of record. The data record on this line, and all other records with a RECTYPE of 0x04, should be ignored. The configuration bitstream is a subset of the dataspace described by all data bytes in the *Warp* HEX file. Some bitstreams are larger than the 16-bit (512-kbit) address space allowed by the Intel HEX format; if the compressed bitstream contains more than 2^16*8 bits, the LOAD OFFSET will become 0x0000 again. This follows standard industry practice for describing a configuration bitstream with the Intel HEX format.

The start of the data payload for the Intel HEX format is the second line. The :10 at the start of the line indicates a data payload size of 16 (0x10) bytes. The next two bytes are the load offset of the data payload. In the first line, the load offset is 0x0000, because this is the start of the data payload. The load offset is measured in "bytes" for the Warp HEX file. The next byte is the RECTYPE field, as described in *Figure 3*. In the example above, this is 0x00, indicating a data record. The 16 bytes (32 hex characters) of the data payload then follow, followed by the one-byte checksum.

The entire data payload of the HEX file, however, does not become part of the bitstream. *Table 2* describes the format of the HEX file output from *Warp*6.2. Note that the data payload that becomes part of the bitstream starts with the DTCB.

In the Warp HEX File Example above, the "signature" mentioned in *Table 2* is the first four bytes of the data payload. The signature is not underlined, and is **05021912**. The next four bytes (eight hex characters) are the length of the header, not including the signature or the length field itself. In the example above, the length is underlined and is **57000000**. This translates to decimal "87" (since the least significant byte is first). The length field is followed by the header version, which is not underlined and is **01000000**. This represents decimal "1". The next 64 bytes are the order code. This starts with the two underlined bytes **6379** and ends with the two underlined bytes **2020** that are immediately prior to the two bytes **3039**. The next byte (pair of hex characters) is the DTCB, which is described in *Table 3*. In the example above, the DTCB is **00**, which represents CRC check on, ring oscillator on, no wait states, and even parity.

Warp-HEX-File-to-Bitstream Example

When the HEX file data is converted to a bitstream, the data is streamed to the Delta39K/Quantum38K device MSB-first. That is, the first data sent to the Delta39K device for the *Warp* HEX Example above are:

00000000 11111111 11111111 11111111 ...

(Read the data stream above left-to right to determine the bit ordering.)

The first eight bits are the DTCB, and after that are the data bits for the device configuration.

For another example, consider a configuration bitstream with the same header, but a different DTCB and data payload:

.
.
.

```
:10005000382F323030312031383A30343A303731 7D
:10006000010203040506070809 0A0B0C0D0E0F1008
```

.
.
.

In this example, the first data that will be streamed out of the CPLD Boot EEPROM to the Delta39K device are:

00110001 00000001 00000010 00000011...

The first eight bits are the DTCB, and after that are the data bits for the device configuration. Note the Intel HEX Format data is shifted out MSB-first, but the first bit of the DTCB that is shifted out is the LSB of the DTCB. Therefore a DTCB of 0x31 in the Intel HEX Format file corresponds to an invalid DTCB with the following characteristics:

CRC enabled (bit 0 = 0)
Reserved bit not set (bit 1 = 0)
Ring Oscillator disabled (bit 2 = 1)
Reserved bit set (bit 3 = 1)
Wait State bit 0 not set (bit 4 = 0)
Wait State bit 1 not set (bit 5 = 0)
Even parity check bit not set (bit 6 = 0)
Reserved bit set (bit 7 = 1)

Approximating Compressed Bitstream Length

To determine the approximate size of the compressed bitstream in a *Warp* HEX file, multiply the file size in bytes (reported by Windows Explorer) by three.

For example, the CY39200 *Warp* HEX file used in the example above has a file size reported by Windows Explorer of 311,971 bytes. The approximate number of bits in this file is therefore 311,971 * 3 = 935,913. The actual number of configuration bits in this example file is 886,472. The approximation routine will always overestimate the number of bits required.

Delta39K Gating of CCLK

The HEX file created by *Warp* is the compressed bitstream, and the bitstream stored in the CPLD Boot EEPROM is the compressed bitstream. The Delta39K requires multiple configuration clocks to internally create the uncompressed bitstream when it is expanding a compressed code to its uncompressed equivalent. As a result, the external configuration clock is suppressed during these events. Monitoring the configuration clock (the pin labeled CCLK) during configuration will reveal that clock pulses are sent to the CPLD Boot EEPROM in small pulse trains. The “dead space” is the space required for the Delta39K to decompress the received configuration data.

Configuration Time

The total time that Delta39K and Quantum38K devices take to be fully configured includes decompression overhead. The devices have an internal oscillator that generates the CCLK signal used both in the configuration process through the internal Flash and external EEPROM. CCLK operates between 2.02 MHz with 3 wait states and 8.06 MHz with 0 wait states. *Table 4* shows the approximate configuration times of CY39100 and CY38100 devices. Future versions of *Warp* will provide the ability to specify the wait state configurations and directly control the configuration clock.

Table 4. Configuration Time

Interface Ports	Wait State (DTCB[5:4])	CCLK (MHz)	Approximate Config Time (CY39100, CY38100)
Serial Configuration and Non-Volatile	0	8.06	200 ms
	1	4.03	384 ms
	2	2.69	568 ms
	3	2.02	756 ms
JTAG (ISR S/W)	N/A	~0.6	8 sec.

To calculate the approximate configuration time of the other devices, use the following formula:

Config Time = (Total Config Bits/CCLK) * 1.08 Eq. 1

Programming Time

ISR takes approximately 60 seconds to program the internal Flash. This programming time also varies depending on the PC. For more information on programming the CPLD Boot EEPROM, please refer to the CYDH2200E CPLD Boot EEPROM Programming Kit User’s Guide.

Configuration Sequence

Powering-up the system will initiate the configuration process when the Reconfig input is driven HIGH. After the system is powered-up, there are two ways to initiate the Delta39K/Quantum38K configuration process:

1. ISR instructions through the JTAG port
2. Toggle Reconfig

Toggling the Reconfig signal will only work when the JTAG TAP Controller (Appendix C) is in the Test-Logic-Reset state. So, ISR must first complete its operation. If ISR issues an instruction anytime during the configuration process, the configuration process will cease and execute the latest JTAG instruction.

If MSEL is LOW, the device will attempt to configure from the NV Configure mode. If MSEL is HIGH, the device will attempt to configure from the external CPLD Boot EEPROM interface. Config_Done will be LOW during configuration and will switch to HIGH when the process is complete.

If a CRC error is encountered, the internal configuration circuitry will restart the configuration process immediately.

NV Configure Mode

NV Configure loads configuration bits from the internal Flash device to the memory cells of the Self-Boot Delta39K device. This is done via the NV port. The configuration process is transparent to the user.

Master Serial Mode

This mode loads configuration bits from an external EEPROM to the Delta39K or Quantum38K device through the Configuration port. To stop (or prevent) the configuration process, hold the Reconfig signal LOW. To start the configuration process, toggle the Reconfig signal from LOW to HIGH. The clock signal for the external EEPROM and the internal circuitry is generated by the Delta39K or Quantum38K and is output as the CCLK signal. After receiving the rising edge on Reconfig, the device will set the Config_Done, CCE and Reset signals to LOW. It will then activate the CCLK. After all configuration data have been shifted in, the device will deactivate the CCLK and set CCE, Reset, and Config_Done High.

Hardware Setup

The hardware set-ups described below apply to CY39100VXXXB and all other Delta39K/Quantum38K devices.

Volatile

For the volatile package, use a pull-up resistor to connect the Ready signal of the EEPROM to V_{CCCNFG} and a capacitor to GND (*Figure 5*). You can also connect a separate 10-pin header to allow the CPLD Boot EEPROM to be reprogrammed with the CPLD Boot EEPROM Programming kit, CYDH2200E. The external Boot EEPROM, such as the CY3LV, should be set as a Slave since Delta39K generates the CCLK signal. The programmable RESET option of the CY3LV needs to be set such that logic HIGH resets the Boot EEPROM.

During power-on reset, the READY pin of the CPLD Boot EEPROM will send out a LOW signal to the Reconfig pin of the Delta39K/Quantum38K device. This will hold the CPLD in a reset state and prevent it from attempting to configure. After the power supplies have met the power-on reset requirements described in the section “Power Supplies” above, the READY pin will be three-stated by the CPLD Boot EEPROM. The 4.7-kΩ pull-up resistor will bring the READY net to a logic high once the 1 µF capacitor is charged. The diode in *Figure 5* prevents any external signal (during normal operation or power-up) that drives the Reconfig line LOW from also activating the SER_EN input of the CY3LV Boot EEPROM.

Pin 10 of the CPLD Boot EEPROM ISP connector is connected to the programmer through a standard 10-pin header. Mechanically, this is the same header that is used for the ISR connection to the Delta39K/Quantum38K CPLD. The “programmer” side of the cable has pin 10 grounded. This forces both SER_EN and the Reconfig pin to be in a logic LOW state when the programmer cable is connected to the ISP header. This will turn on the programming mode of the CPLD Boot EEPROM and hold the Delta39K/Quantum38K device in reset. That is, the logic inside the Delta39K/Quantum38K device will not function (the device will not be in “PAL” mode) while the Boot EEPROM programming cable is connected to the header. After the data is programmed into the Boot EEPROM, removing the Boot EEPROM programming cable will cause SER_EN to go HIGH due to the pull-up resistor on SER_EN. Provided the Boot EEPROM is ready (i.e., the READY output of the Boot-PROM is three-stated), the pull-up resistor on the Reconfig net will allow the Reconfig signal to be pulled HIGH, causing the Delta39K/Quantum38K device to start configuration. For the hardware set-up for volatile CY39100VXXX and CY39100VXXXA devices, please refer to Appendix E.

Self-Boot

In the Self-Boot device, the MSEL signal can be connected directly to GND (*Figure 6*). This will select the internal Flash memory as the configuration source. The DATA pin should be pulled HIGH or LOW and the CCLK, CCE, and Reset pins can be left unconnected. If an external EEPROM is to be used with a Self-Boot part as a configuration source, set up the connection the same way as in the volatile device hardware set-up.

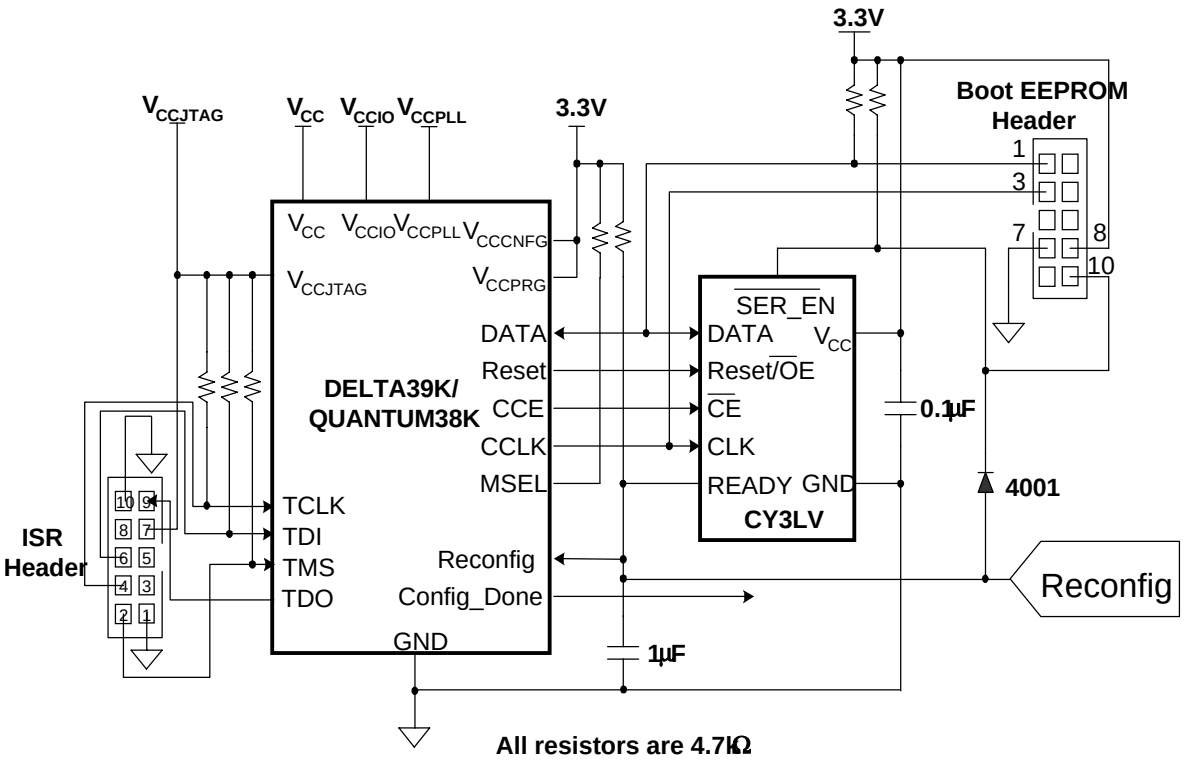


Figure 5. Volatile Delta39K/Quantum38K Device Hardware Set-up

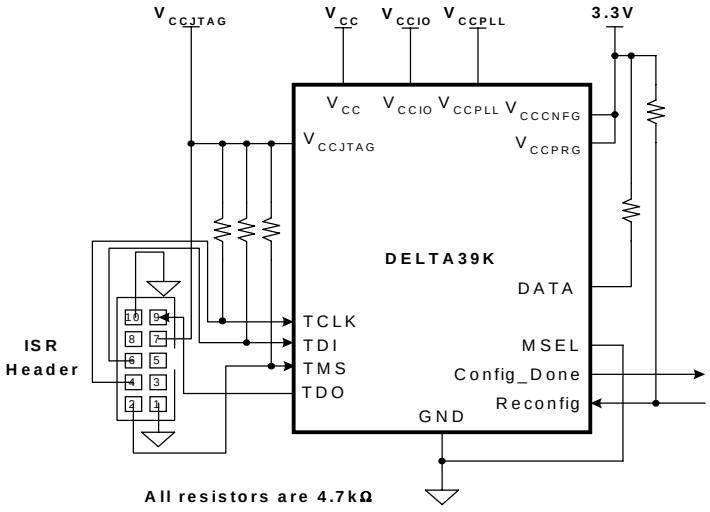


Figure 6. Self-Boot Delta39K Hardware Set-up (including CY39100V388 and CY39100V388A)

APPENDIX A - Timing Diagrams and Parameters

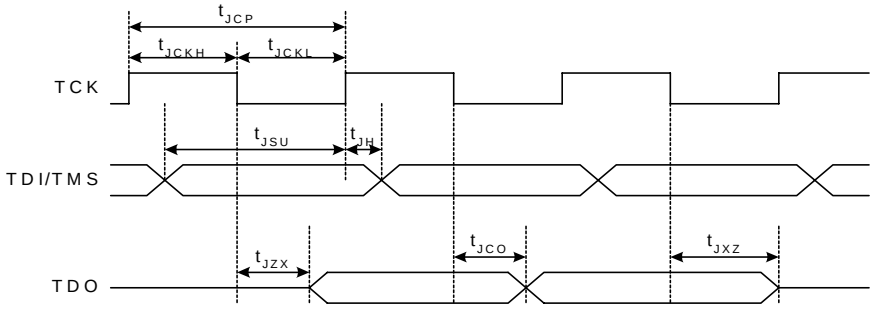


Figure 7. JTAG Timing Diagram

Table 5. JTAG Timing Parameters (Max. operating frequency at 20 MHz)

Parameter	Definition	Min.	Max.	Unit
t_{JCKH}	TCK Clock High Time	25		ns
t_{JCKL}	TCK Clock Low Time	25		ns
t_{JCP}	TCK Clock Period	50		ns
t_{JSU}	JTAG Port Set-up Time (TDI/TMS)	10		ns
t_{JH}	JTAG Port Hold Time (TDI/TMS)	10		ns
t_{JCO}	JTAG Port Clock to Output (TDO)		20	ns
t_{JXZ}	JTAG Port Valid Output to High Impedance (TDO)		20	ns
t_{JZX}	JTAG Port High Impedance to Valid Output (TDO)		20	ns

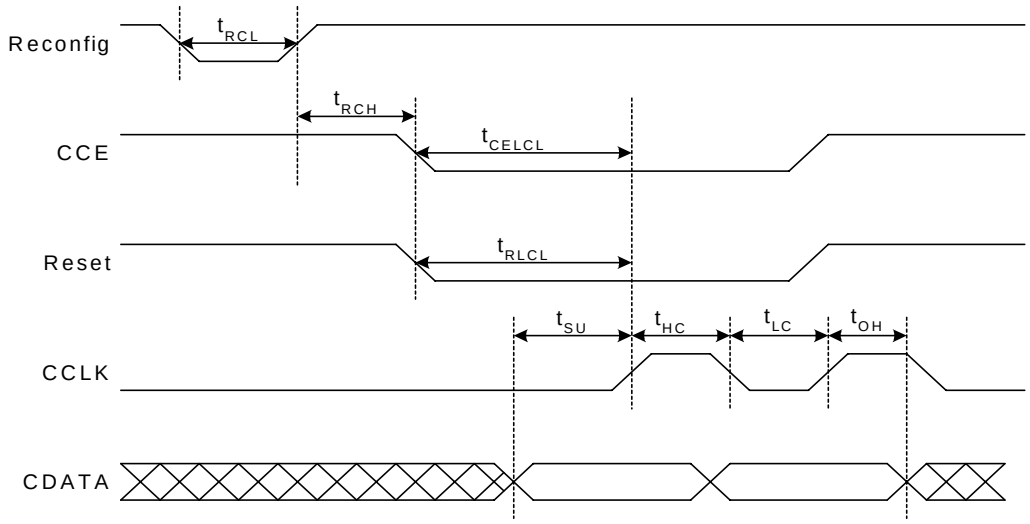
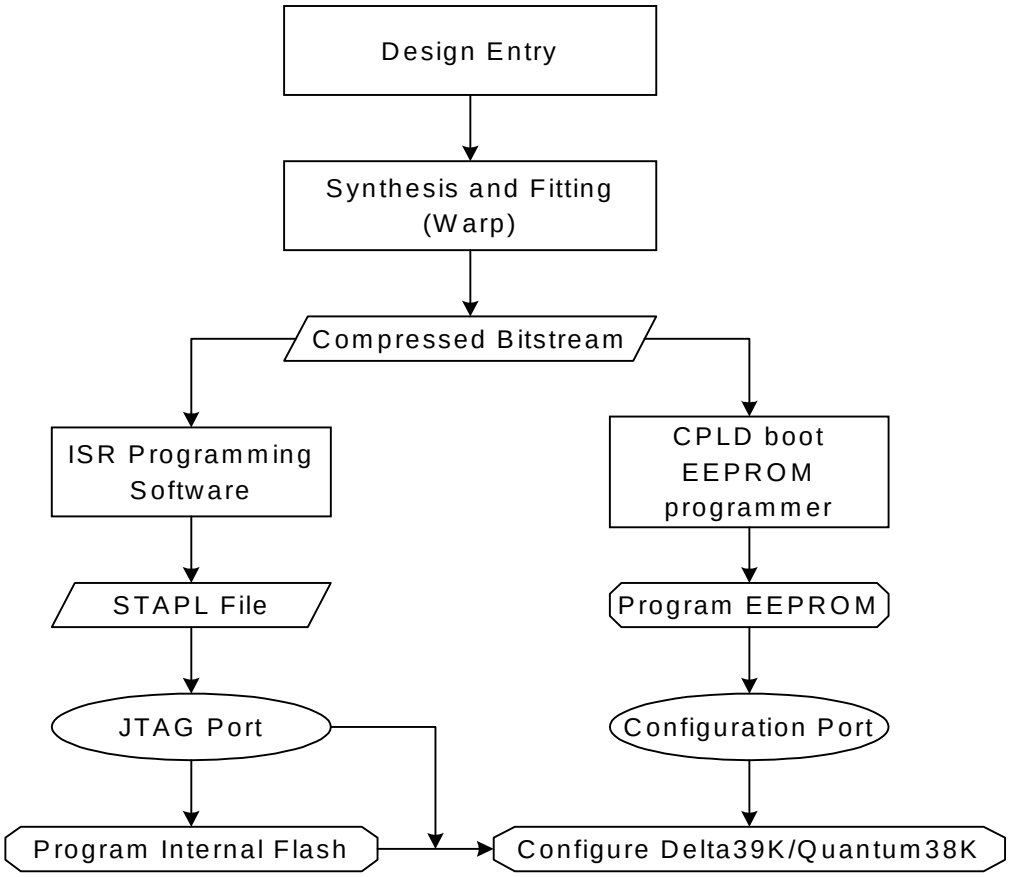


Figure 8. Configuration Port (Master Serial) Timing Diagram

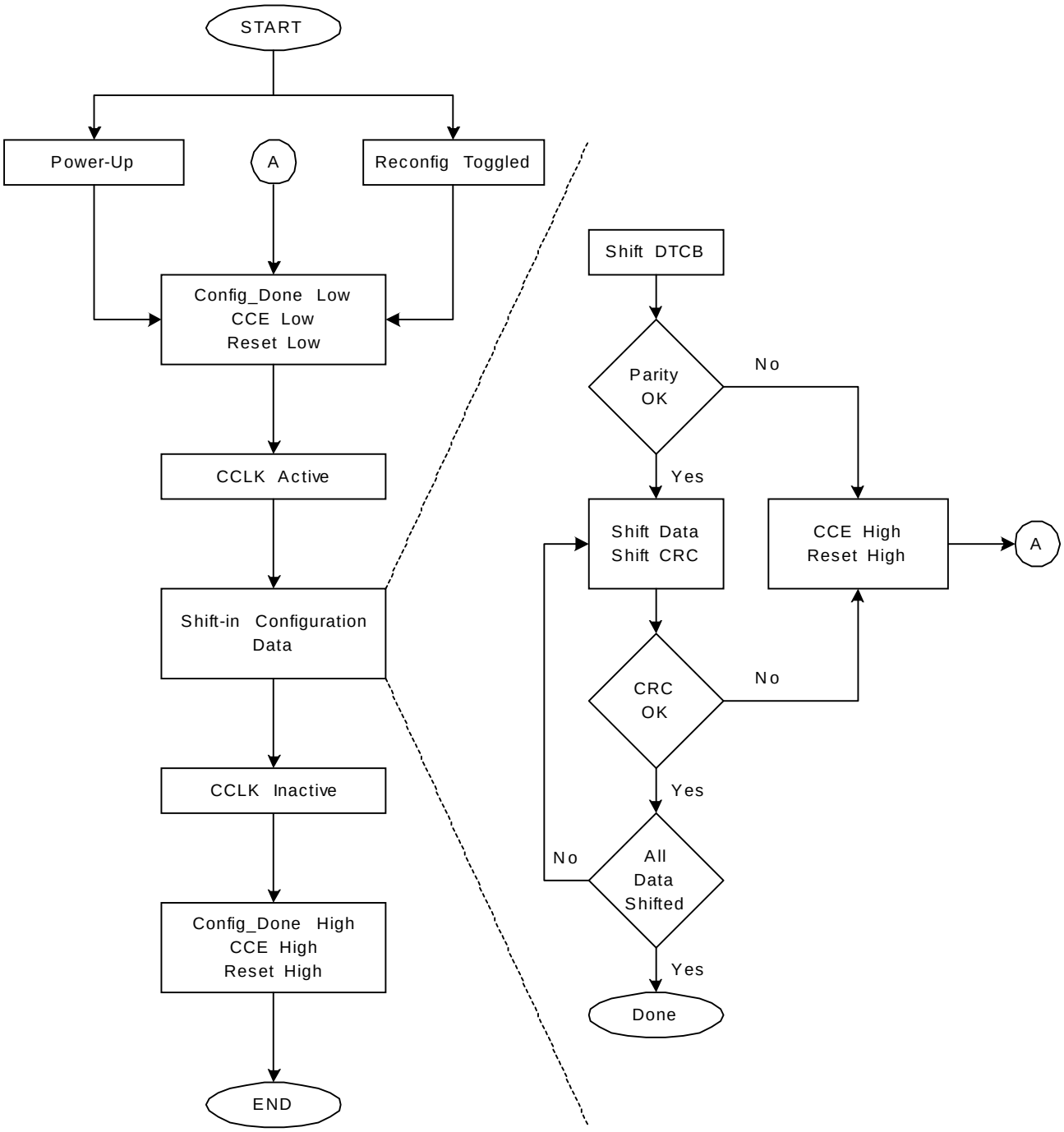
Table 6. Configuration Port (Master Serial) Read Cycle Timing Requirements

Parameter	Definition	Min.	Max.	Unit
t_{RCL}	Reconfig Low Time	200		ns
t_{RCH}	Reconfig High Time		200	ns
$t_{CELCL/RLCL}$	CCE/Reset Low to First CCLK Positive Edge		3.5	µs
t_{SU}	CDATA Set-up to CCLK	20		ns
t_{OH}	CDATA Hold from CCLK	0		ns
$t_{HC/LC}$	CCLK - High/Low time	60	250	ns

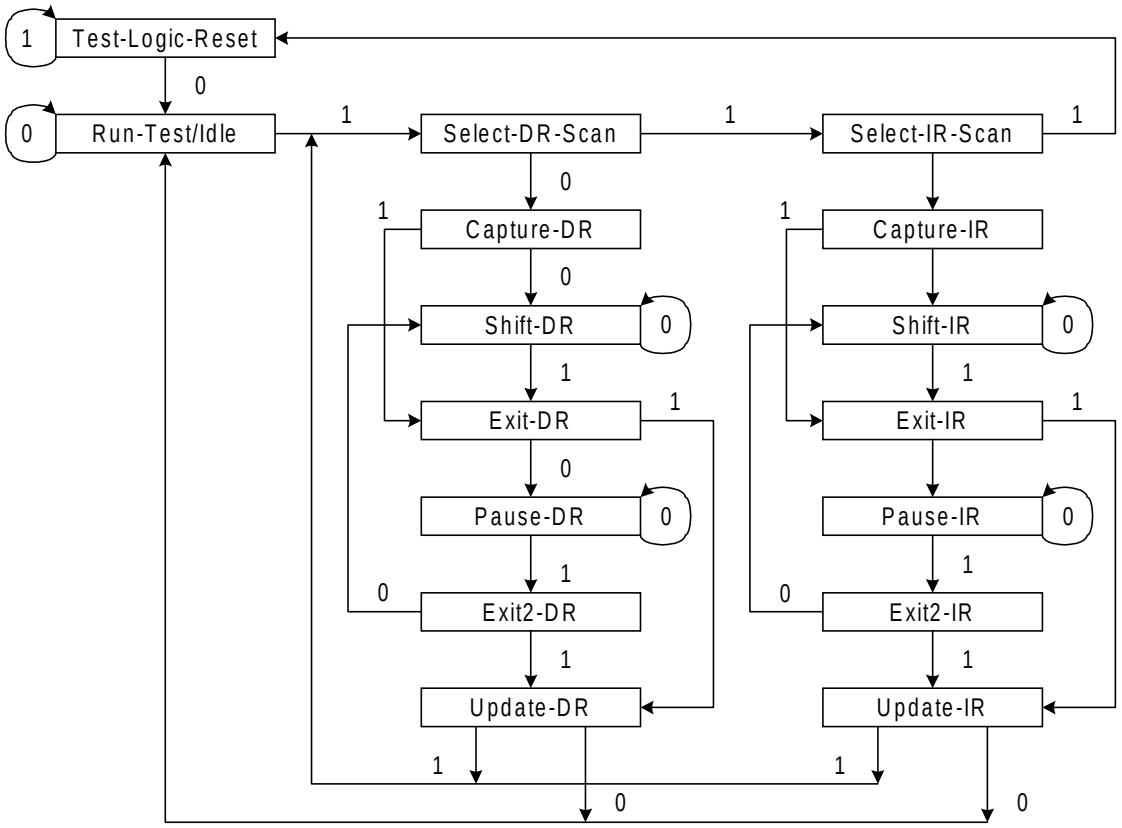
APPENDIX B - Delta39K/Quantum38K Design Flow



APPENDIX C - Internal Flash and External EEPROM Configuration Flow

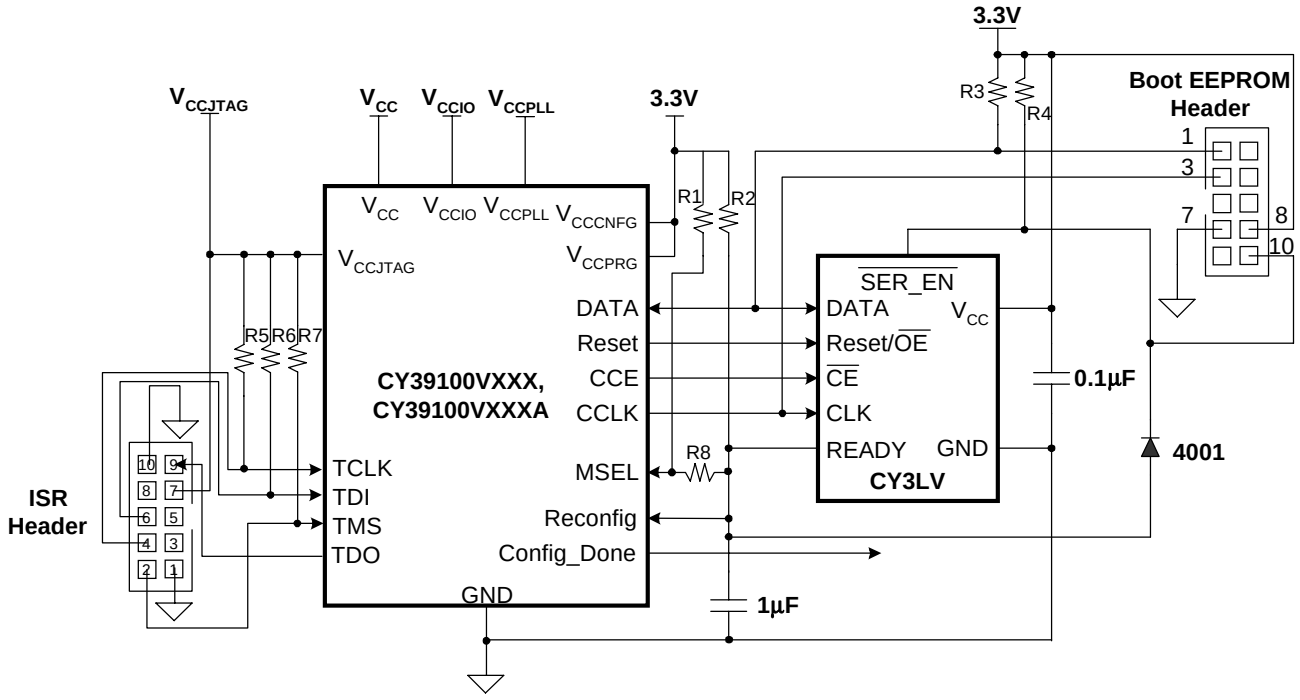


APPENDIX D - JTAG TAP Controller



Transitions based on TMS on the rising edge of TCK

APPENDIX E - Volatile CY39100VXXX and CY39100VXXXA Board Set-up



Follow the population guidelines below to allow migration from CY39100VXXX or CY39100VXXXA to CY39100VXXXB. For further information on CY39100VXXX, CY39100VXXXA, and CY39100VXXXB please refer to the Delta39K Datasheet Errata page.

CY39100VXXX and CY39100VXXXA:
R1: Not Populated
R2-R7: 4.7kΩ
R8: 0Ω

CY39100VXXXB:
R1-R7: 4.7kΩ
R8: Not Populated

The intent of the schematic above is to allow a designer using CY39100V208 or CY39100V208A to lay out their board once by providing a migration path from CY39100V208/CY39100V208A to CY39100V208B. The PCB of a design using CY39100V208/CY39100V208A should be laid out to match the schematic above (the schematic for the CY39100V208B and all other Delta39K family members can be found on page 7 of this application note). The population guidance on the left should be followed when the devices used are CY39100V208/CY39100V208A. Once the CY39100V208B is received, the same PCB layout can be used — except this time the resistors are populated differently. The effect of the population change is to have Reconfig separate from MSEL for CY39100V208B and to have Reconfig shorted to MSEL for CY39100V208 and CY39100V208A.