El Camino
Training - Engineering - Consultancy

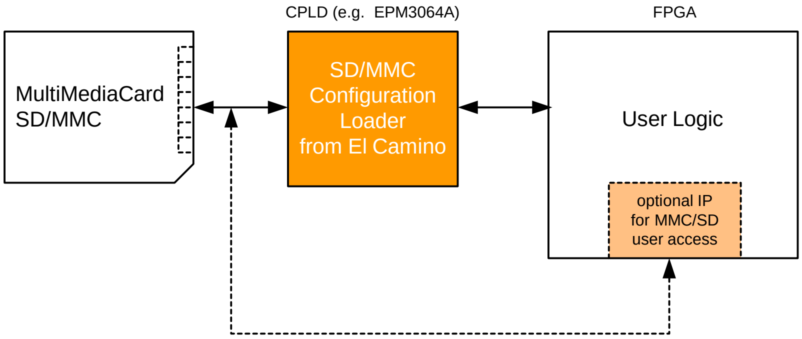
SD/MMC Loader

FPGA Configuration IP

General Description

As Altera® introduces higher-density programmable logic devices (PLDs) to the market, the size of the configuration bit streams also increases. Using a standard MultiMedia™ Card (MMC) memory device together with a standard PLD for configuring high density FPGAs provides a very flexible and versatile solution. The MultiMediaCard is a universal, low cost data storage and communication media, which is generally available and widely used in consumer products such as digital cameras or cellular phones.

Figure 1: Block Diagram



Preliminary Information

Features

- Requires only 55 Macrocells in a low cost MAX3000A device (e.g. fits easily into an EPM3064A)
- Accepts any input clock up to 170 MHz
  - Generation of up to 400 kHz clock for MMC initialization
  - Generation of up to 20 MHz clock for configuration data retrieval and loading into FPGA
- Automatically disconnects from MMC after configuration (design within FPGA can take control over MMC memory)
- Supports all current Altera SRAM based FPGA devices (Stratix™, Stratix II, Cyclone™, Mercury™, APEX™, ACEX® and FLEX®)

Applications

The SD/MMC Loader is ideal for applications where a mobile and exchangeable media for FPGA configuration data storage is required. Furthermore FPGA configuration data can be combined with application data or program storage on a removable, common and compact storage media.

Deliverables

- Encrypted gate level netlist optimized for Altera's CPLD architectures

alternatively

- VHDL source code

Architecture Specification

Figure 2: I/O Ports

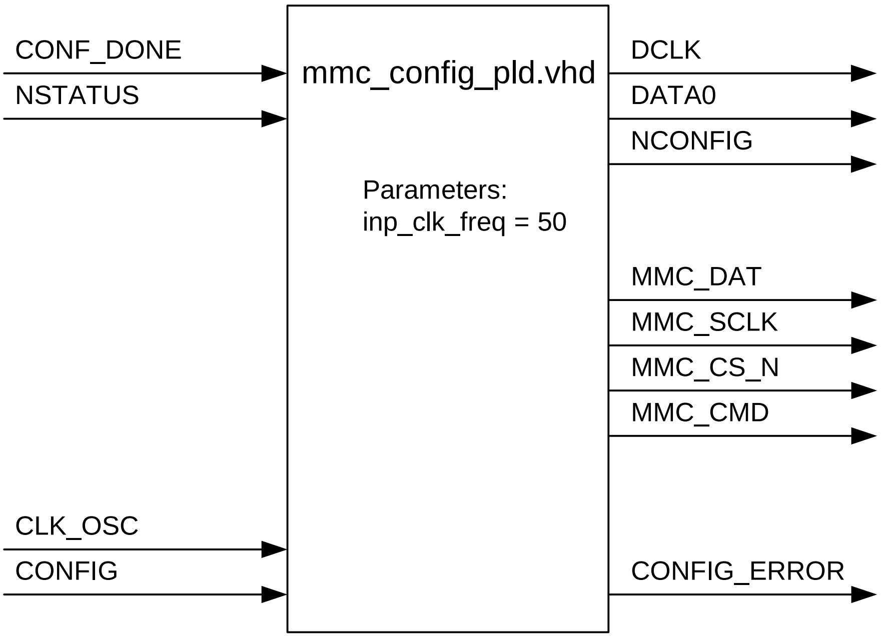


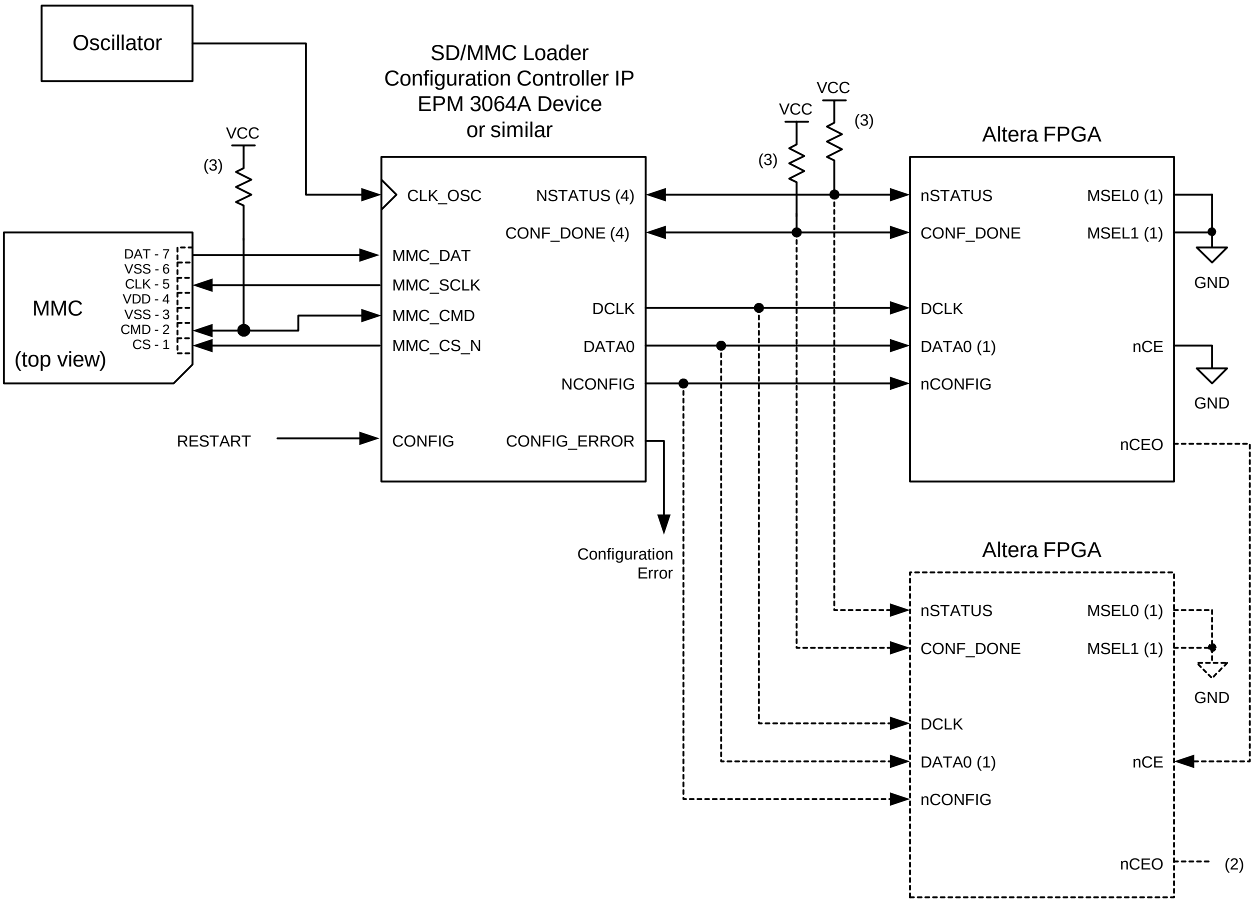
Table 1: Port Description

Port	Direction	Function
CLK_OSC	input	Main clock input. The internal clock dividers are parameterizable and support up to 170 MHz input clock. From this clock the up to 400 kHz MMC initialization clock and the up to 20 MHz configuration clock are derived. The input clock frequency may vary but must not exceed the frequency specified by the generic inp_clk_freq
CONFIG	input	Configuration control input. When low, resets the whole configuration controller. In the reset state the NCONFIG output is pulled low so the FPGA looses it's configuration. A low to high transition starts the configuration process.
CONF_DONE	open-drain	Status input. The target FPGAs drive the CONF_DONE pin low before and during configuration. Once all configuration data is transmitted without error and the initialization clock cycle starts, the target FPGAs release the CONF_DONE signal. There should be a 1 kOhm pull-up resistor connected to this signal. To be connected to the target FPGAs.
NSTATUS	open-drain	Configuration status signal. There should be a 1 kOhm pull-up resistor connected to this signal. To be connected to the target FPGAs.
DCLK	output	Configuration clock signal. During configuration the frequency will be up to 20 MHz depending on the CLK_OSC input clock and the inp_clk_freq generic setting. To be connected to the target FPGAs.
DATA0	output	Serial configuration data ouput. To be connected to the configuration data inputs of the target FPGAs.
NCONFIG	output	Configuration control output. A low transition resets the target device; a low-to-high transition begins configuration. To be connected to the target FPGAs.
CONFIG_ERROR	output	Configuration status output. This pin will go high if configuration was unsuccessful. To restart a new configuration the CONFIG input needs to be pulsed low.
MMC_DAT	input	MMC data signal. Only used as an input to the configuration controller. Carries the serial configuration data read from the MMC device.
MMC_SCLK	output	MMC clk signal. Up to 400 kHz clock during MMC initialization. Automatically switched to up to 20 MHz for configuration data transmission. Tri-stated after successful configuration, in case of configuration error and during reset (CONFIG input low).
MMC_CS_N	output	Chip select signal to the MMC card. Always high during MMC card access (only relevant in MMC SPI mode which is not used by the configuration controller). Tri-stated after successful configuration, in case of configuration error and during reset (CONFIG input low).
MMC_CMD	open-drain	MMC command signal. Used to initialize and control MMC. There should be 1 kOhm pull-up resistor connected to this signal. Tri-stated after successful configuration, in case of configuration error and during reset (CONFIG input low).

Table 2: Parameter Description

Parameter	Legal Values	Default	Description
inp_clk_freq	1-170	50	Input clock frequency in MHz. Allows generation of up to 400 kHz during MMC initialization and up to 20 MHz during data transfer by internal clock generators.

Figure 3: Connecting the configuration controller



(1) FLEX 6000 devices have a single MSEL pin, which is tied to ground. Additionally, its DATA0 pin is renamed DATA

(2) The nCEO pin is left unconnected for the last device in a chain

(3) Pull-up resistors are 1 kOhm except for APEX 20KE devices. For APEX 20KE devices, pull-up resistors are 10 kOhm

(4) The nSTATUS and CONF_DONE pins are open-drain on the Altera FPGA devices. The corresponding pins on the configuration controller are also open-drain.

Configuration Process

Upon power up or after the CONFIG input going high the configuration controller will automatically start the configuration process. While the MultiMediaCard standard allows a bus of MultiMedia Cards, the configuration controller only supports a single MMC card connected to the configuration controller. Upon configuration start the controller will control both the FPGA configuration pins as well as the MMC card in parallel. Once both the FPGA and the and MMC card are ready the transmission of the configuration data starts.

The FPGA configuration scheme used is Passive Serial (PS). Configuration starts by pulsing NCONFIG low for at least 40 us. After the releasing the NCONFIG signal, the configuration controller waits for NSTATUS to go high. Once NSTATUS is high it will wait at least 1 us before transmission of configuration data starts. Whenever NSTATUS goes low, signaling a checksum error in one of the configuration data frames, the configuration controller enters an error state and drives the CONFIG_ERROR output high. A new configuration can be started by pulsing CONFIG low or cycling power. A successful configuration can be determined by examining the FPGAs optional INIT_DONE output.

During initialization the MMC card will be assigned the Relative Card Address (RCA) 0x1. After successful configuration the MMC card is left in the „Sending-data-State (data)“. It can easily be sent back to the „Idle State“ by issuing a CMD0 from a user application within the FPGA. By sending CMD0 the MMC card can also be switched into SPI mode or be re-initialized in MultiMediaCard mode with a different RCA and different options. Contact El Camino for further details on the commands and sequences used to initialize the MMC card. Refer to the MulitMediaCard System Specification, which is available from the MMCA Technical Committee for further information on accessing the MultiMediaCard.

Resource Utilization and Performance

The following results are based on synthesis and place & route in Quartus II Version 4.0. The Fmax frequency is the maximum synchronous input clock frequency supported. The maximum frequency of the data transfer is limited by both the maximum DCLK frequency of the FPGA and the maximum SCLK frequency of the MMC.

Table 3: Resources and Performance

Device Family	Ressource	Fmax
MAX II	102 Logic Elements	170 MHz
MAX 3000A	55 Macrocells	153 MHz
MAX 7000S	55 Macrocells	111 MHz

Configuration Data Storage

Basic Concept

When developing the configuration controller, one goal was to be able to fit the design into a small, non-volatile MAX device. Since the resources in such a device are very limited it was not possible to handle a complex file system on the MMC card and access a configuration file based on its name. As a consequence the configuration controller expects the raw configuration data at a fixed address, at 0x400. It is still possible to have a hard-disk like file system with logical drives on the MMC card. The configuration data is expected at 0x400 which is after a possible partition table that normally ends at 0x1ff.

Accessing the MMC after configuration

It is possible to connect the configuration controller in parallel with a FPGA to the MMC card. After successful configuration the configuration controller will tri-state all signals connected to the MMC card and allow a design in the FPGA to take control over the MMC card. Since the MultiMediaCard mode is used for configuration by the configuration controller the application can still select between MultiMediaCard mode and SPI mode. If the application switches to SPI mode the power to the MMC card must be cycled (turned off and on) before a new configuration can take place. This is because MMCs do not allow to leave the SPI mode without cycling the power. For maximum flexibility the MultiMediaCard mode was chosen for configuration. Contact El Camino for availability information on a NIOS ready MMC/SD card interface with according C/C++ software drivers to be used within a custom FPGA application.

Storing configuration data on MMC card

The file format used to put configuration data onto the MMC card is the Altera Raw Binary File (.rbf) format. In Quartus II you can use the **Programming Files** tab of the **Device & Pin Options** dialog box, which is available from the **Device** page of the **Settings** dialog box (Assignments menu), to direct the Quartus II compiler to generate a RBF. It will contain configuration data for the Passive Serial (PS) configuration scheme. You can also generate RBFs with the makeprogfile utility, or from previously generated SRAM Object Files (.sof) with the **Convert Programming Files** command in Quartus II (File menu). The latter method can be used to chain together multiple .sof configuration files into a single .rbf file. The resulting configuration file can be used to configure a chain of FPGAs as shown in figure 2.

Since the FPGA expects the configuration data on its DATA0 input LSB first, the MMC however supplies the bits of each byte MSB first, some conversion needs to take place. In order to keep the size of the configuration controller as small as possible it was decided to do the conversion before storing the data on the MMC card. Otherwise it would have been necessary to convert each byte of serial data within the configuration controller into parallel data and then shift out the data in reverse order. If however the data is read in the correct bit order from the MMC card it can be directly passed on to the FPGA.

There is a Windows utility that comes with the configuration controller IP. This utility will read a specified .rbf file, reverse the bit order in each byte and store the data at physical address 0x400 onwards in raw binary format on a specified removable storage device.

Figure 4: MMC card memory space

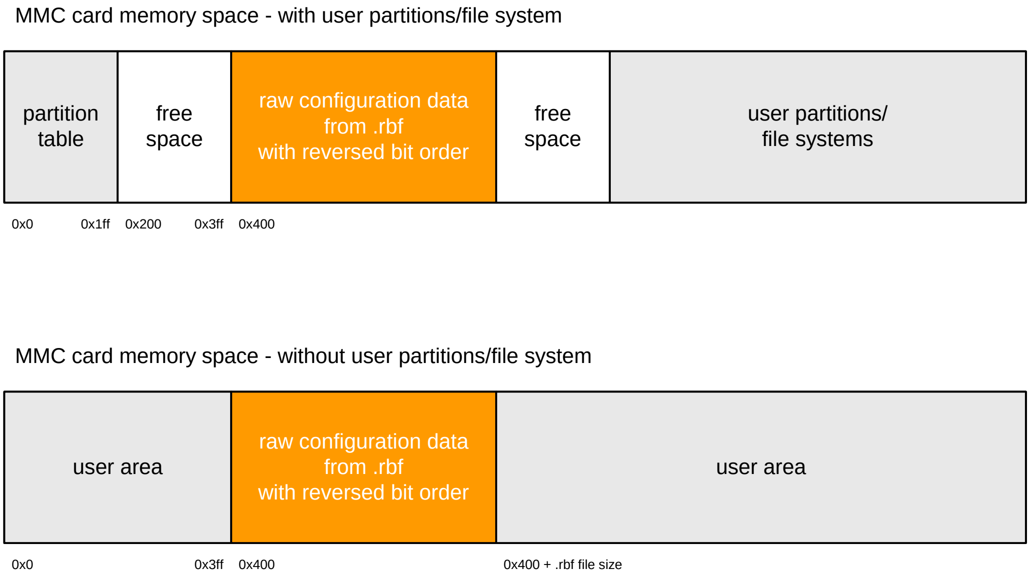

MMC card memory space - with user partitions/file system
partition table
free space
raw configuration data from .rbf with reversed bit order
free space
user partitions/ file systems
0x0
0x1ff
0x200
0x3ff
0x400
MMC card memory space - without user partitions/file system
user area
raw configuration data from .rbf with reversed bit order
user area
0x0
0x3ff
0x400
0x400 + .rbf file size

Figure 4 shows possible organizations of the MMC card memory space. In case a file system is to be used it is the responsibility of the user to partition the MMC card in an appropriate way. One needs to make sure that there’s enough space left at the very beginning to store the configuration data before the first partition starts. Use the Windows disk manager to partition and format the MMC card so it offers the required logical drives but yet leaves enough space at physical address 0x400 onwards for .rbf file storage. The utility provided will check for enough space outside of any partitions before writing data to the MMC card.

In case no file system is used it is still possible to access all areas of the

MMC card using physical addresses. It is up to the user to make sure that the configuration data does not get corrupted during access to the remaining memory areas of the MMC card. Again, contact El Camino for availability information on IP cores and software drivers to access the MMC card from within the FPGA.

Custom Solutions

Please contact El Camino if you require custom configuration solutions based on this IP. The following options, among others are feasible:

- Select from multiple configurations based on the setting of configuration controller device inputs
- Store configuration data at a different memory location
- Use SPI mode instead of MultiMediaCard mode for MMC access
- Automatically restart configuration on error
- and more...

Notes:

El Camino
Training - Engineering - Consultancy

El Camino GmbH
Landshuter Str. 1
D-84048 Mainburg
Germany
Telefone +49-8751-8787-0
Telefax +49-8751-842876
E-mail: info@elca.de
http://www.elcamino.de

El Camino GmbH Training - Engineering - Consultancy, DIGILAB 10K10, DIGILAB picoMAX, DIGILAB 10Kx240, DIGILAB 20Kx240, DIGILAB megAPEX and other names of El Camino products, product features and services are trademarks and/or service marks of El Camino GmbH in Germany and other countries. Altera, APEX, Stratix, Quartus, NIOS and other names of Altera products, product features and services are trademarks and/or service marks of Altera Corporation in the United States and other countries. Other product and company names mentioned in this document may be the trademarks of their respective owners.

No warranties: This documentation is "as is" without any express or implied warranty of any kind including warranties of merchantability, no infringement of intellectual property or of fitness for any particular purpose. In no event shall El Camino or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption or loss of information) arising out of the use of or inability to use this documentation, even if El Camino has been advised of the possibility of such damages. Because some jurisdictions prohibit the exclusion or limitations of liability for consequential or incidental damages, some of the above limitations may not apply to you.

El Camino further does not warrant the accuracy or completeness of the information, text, graphics or other items contained in this document. El Camino may make changes to these materials, or to the products described therein, at any time without notice. El Camino makes no commitment to update this document.