MMC Device IP Core

Introduction

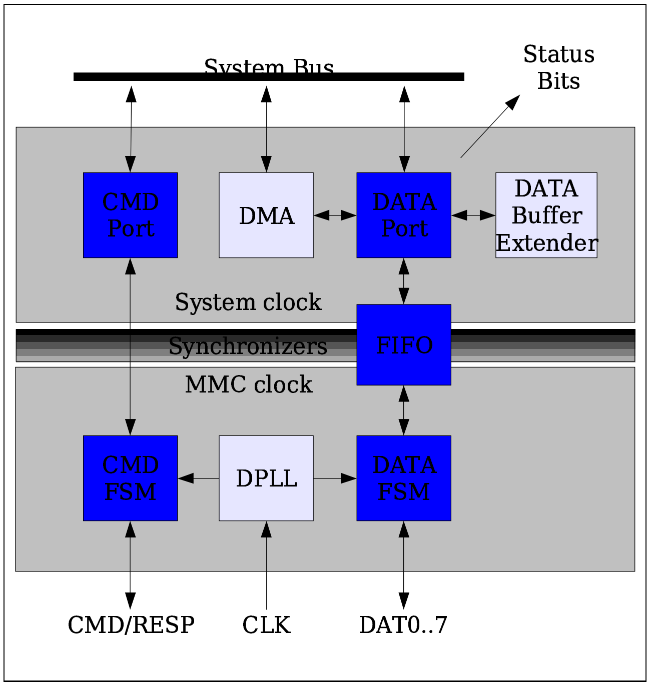
A compact low power and scalable IP core which provides a simple, firmware-friendly cost-effective Physical Link interface for MultiMediaCard-based memory and i/o devices, such as MMC memory cards, MMC-RS, MMC-Mobile, CE-ATA devices etc.

Features

The main features of the MMC Device IP Core are:

- Compliant with MultiMediaCard Spec ver4.X/3.X
- Supports MMC 1bit, 4bit, 8bit modes, as well as SPI mode
- Maximum data rate up to 416Mbits/sec
- All command and response types are supported
- Supports Interrupt-mode (Wait-IRQ)
- Supports CEATA specifications (ver1.0), including Completion Signal
- Generic 8/16/32 bit system bus interface
- Optional extended data buffering 0-4K bytes.
- Optional read/write data DMA
- Card-Busy signal asserted by hardware, negated by firmware
- CRC7 and CRC16 checksum logic
- Supports data block size of 1 byte to 4K bytes
- Set of "Read-Clear" status bits with software interrupt mask
- Multi-block read and write
- Stream read and write
- Built-in Bus Test Procedure
- Supports fast and slow cards. MMC clock frequency: 0-52+MHz
- Supports MMC clock suspension
- Supports hot card insertion and removal
- Compact and trivial firmware interface

The MMC Device IP core is designed to integrate easily with any card controller and any system bus, with minimum firmware development overhead.



An optional data buffer extension module is provided in order to improve CPU utilization, e.g. for performance-sensitive systems. An internal DMA engine gives further CPU off load capability.

The power and area are key factors throughout the design, ensuring a real cost-effective solution for MMC based products.

Architecture

The front end of the core implements a MMC Physical Link as defined by MMC specifications for 1/4/8 data lines and for SPI interface. At its back end the core interfaces to the system controller through a generic 8 bit bus based on Wishbone specifications.

Command Channel

The Command Channel consists of a Command Port and a Command FSM. The Command Port provides a programing interface for accepting commands and sending back responses. The control and status signals are synchronized and connected to the Command FSM, which resides in the MMC clock domain. The following activities are done within the context of the Command FSM: serializing and de-serializing the commands over the MMC command line, CRC7 check for incoming commands, CRC7 calculation for outgoing responses, command/response related errors and status report, controlling busy signal after responses, SPI mode recognition.

Data Channel

The Data Port together with the Data FSM build up the Data Channel portion. The Data Port provides a programing interface for receiving write data sent by the host following write commands, and for sending data to the host in response to read commands. The MMC physical data interface supports 1, 4 and 8 lines of data. Following a read command, the system controller starts writing data bytes to the Data Port. The data are manipulated and sent over the physical lines by the data FSM. CRC16 checksum is added at the end of each block. In the reverse way, following a write command, the data FSM collects bits from the physical interface, packs them into bytes and pushes the bytes into the FIFO. The system controller in turn reads the new data through the Data Port. System resources can be better utilized with the buffer extension module which stores the data blocks during the data FSM read/write operations, thereby providing an additional system controller and bus offload.

DMA (Optional)

The DMA engine can be programmed to transfer data between a specified locations on the system bus address space and the Data Port FIFO, in both directions (receive and send).

Synchronization

The front end of the core consists of command and data state machines driven by MMC clock. The back end consists the programing interface entities (Command Port and Data Port) and is driven by the same clock domain of the system bus. The control and status signals that cross the two clock domains are synchronized using 2-flop based paradigm, while the data are synchronized using a Dual-port FIFO.

Size And Speed

Sample synthesis results. Actual gate count depends on core configuration.

Technology	***Gate Count***	***Operating Frequency***
Xilinx Virtex 4 -11	772 Slices, 2 BRAMs	Card Bus >50 MHz; SoC bus > 100 MHz
UMC 0.18u	10K Gates	Card Bus >50 MHz; SoC bus > 160 MHz

These synthesis results are provided for reference only. Please contact us for estimates for your application.