

Singing Very High Speed Integrated Circuit Hardware Description Language Board (S76D)

Kai Fabian
Hasso-Plattner-Institut Potsdam

Dominik Moritz
Hasso-Plattner-Institut Potsdam

May 12, 2013

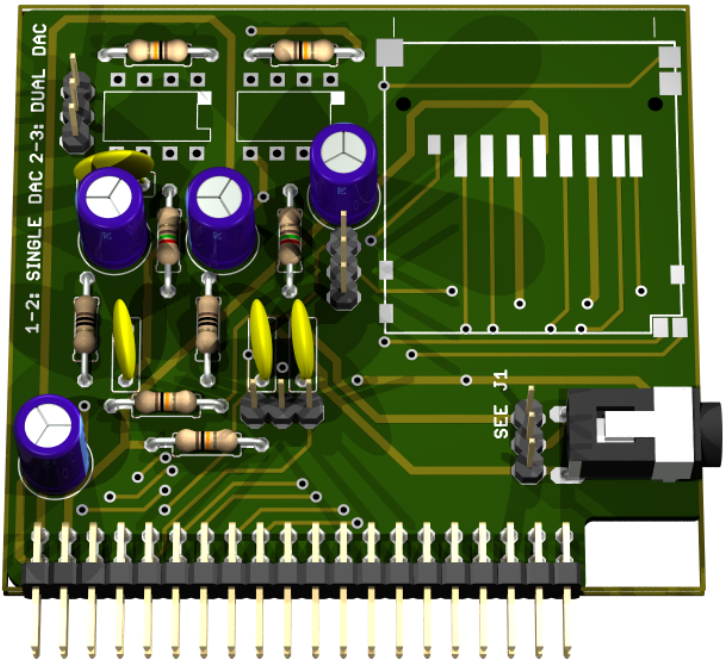


Figure 1: 3D rendered image of the custom extension board used for this music player

Abstract

A VHDL implementation of the MMC SPI interface ...

Contents

1	Introduction	2
2	Multimedia Card	2
2.1	Interface description	2
2.2	SPI Interface	2
2.2.1	Initializing the card	3
2.2.2	Reading data	3
2.3	Implementation of Communication	3
2.4	Most important MMC SPI commands	3

1 Introduction

2 Multimedia Card

2.1 Interface description

Figure 2 shows an MMC card and Table 1 describes the pins of the MMC card.

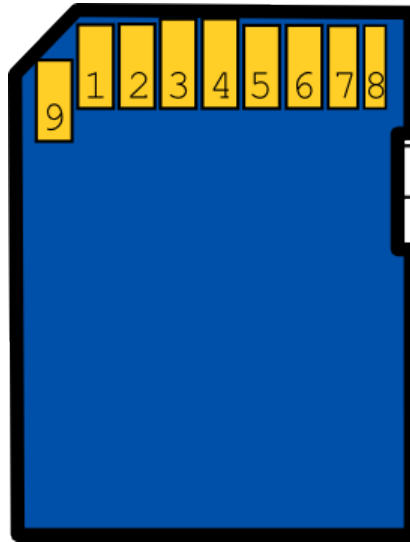


Figure 2: MMC Interface

Pin. No.	Name	Description	Note for our implementation
1	DAT3	MMC: Chip select	Chip IO Pin, mmc_cs, D7
2	CMD	Command	Chip IO Pin, mmc_mosi, D8, (MOSI – master out, slave in)
3	GND	Ground	Board ground
4	VDD	Voltage	Supply Voltage (+3.3 V)
5	CLK	Clock	Chip IO Pin, mmc_clock, D10
6	GND	Ground	Board ground
7	DAT0	Data	Chip IO, mmc_miso, B4, (MISO - master in, slave out)
8	DAT1	Data	Pull up (Voltage via resistor)
9	DAT2	Data	Pull up (Voltage via resistor)

Table 1: MMC card pin description

2.2 SPI Interface

The SPI Interface uses two connections to the MMC. First the *master out, slave in* to send commands to the MMC and the *master in, slave out* to receive the results from the MMC. You can see the pins on the MMC in figure 2.

Sending commands requires you to stick to a predefined protocol that we will describe. A basic sequence for one command can be found in Figure 3. In Figure 4 and 5 you can find the sequence of commands we have to send to the card in order to initialize it or read a block.

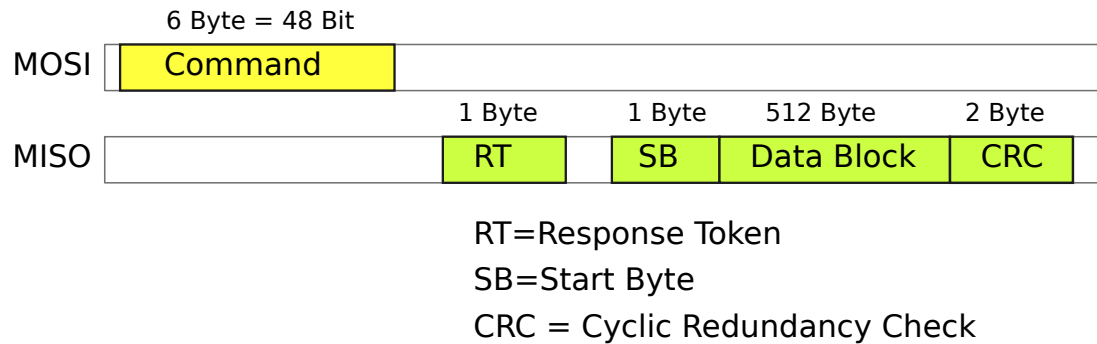


Figure 3: Sending a command to the MMC

Each command is a binary coded sequence in a format that is defined in the official reference manual. Figure 3 should give you a basic idea of how the communication works. The start byte (SB) consists of 8 times HI (1). Before that, the card only sends 0s. The response token (RT) consists of 8 bits. Each bit indicates a certain error or a successful command. In case the command finished successfully, the card returns 0000 0001.

2.2.1 Initializing the card

The MMC cards needs to be initialized before it can be used to read data. According to the specs, one need to send a **CMD0** and **CMD1** after at least 74 clock cycles to synchronize the card clock. The initialization process is complete when the card responds with 0000 0000 (busy) instead of 0000 0001 (ready).

The whole initialization procedure is shown in Figure 4.

2.2.2 Reading data

Reading from the MMC card is shown in Figure 5. The command **CMD17** to reads 1 block of data from the MMC card starting at the address specified in the argument field. The size of the block that is read has to be defined with a command **CMD16** (which is not shown here).

2.3 Implementation of Communication

Figure 6 shows the finite state machine of our implementation in VHDL. {*n*} means that the number preceding the brackets is repeated *n* times.

2.4 Most important MMC SPI commands

Table 2 shows the most important command that we used in our implementation.

More details about the SPI commands, responses and timings can be found in the MMC specification.

List of Figures

1	3D rendered image of the custom extension board used for this music player	1
---	--	---

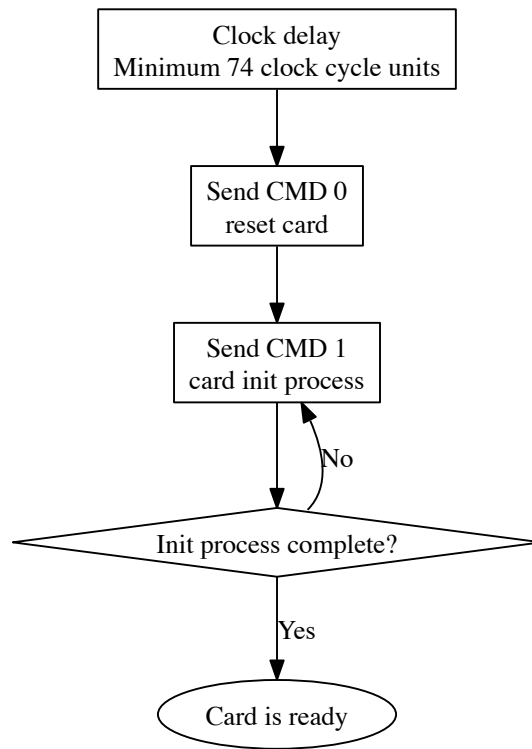


Figure 4: Flow for initializing

2	MMC Interface	2
3	Sending a command to the MMC	3
4	Flow for initializing	4
5	Flow for reading	5
6	Controller VHDL FSM	6

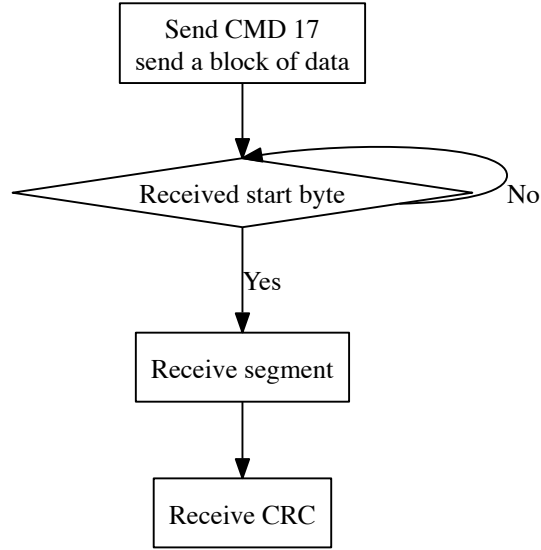


Figure 5: Flow for reading

Command	Name	Description
CMD0	GO_IDLE_STATE	Reset the card. If CS == 0, the card goes into SPI mode.
CMD1	SEND_OP_COND	initialize the card. The command has to be sent repeatedly until the response changes from 0000 0000 (busy) to 0000 0001 (ready).
CMD16	SET_BLOCKLEN	Set the block size. Send the block size as the argument.
CMD17	READ_SINGLE_BLOCK	Read one block as specified in CMD16. Send the start address as the argument.

Table 2: MMC SPI commands used in this implementation.

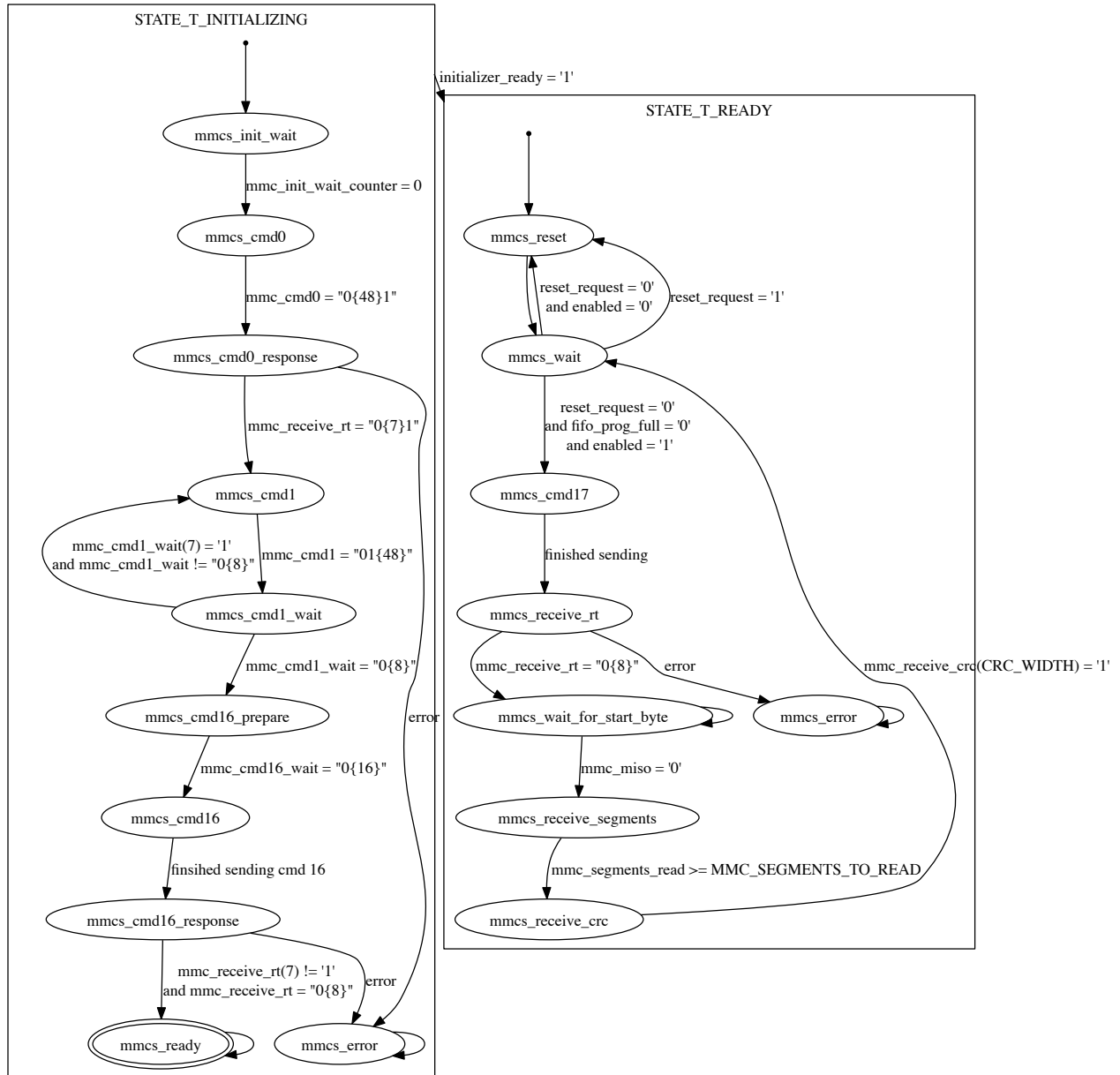


Figure 6: Controller VHDL FSM