DESIGN OF ADVANCED DIRECT MEMORY ACCESS 2 FOR SECURE DIGITAL HOST CONTROLLER

Ramya. M

PG Scholar/VLSI Design, ECE Dept, SNS College of Technology, Coimbatore-641 035, (India)

ABSTRACT

Portable storage devices are becoming popular and growing rapidly. These devices can store and acquire information wherever whenever you need. The aim of this project is to design Advanced Direct Memory Access2 (ADMA 2) that supports high capacity data transfer with greater Speed and reduced Power Consumption. The ADMA 2 Controller is a hardware feature that enables movement of blocks of data from peripheral to memory, memory to peripheral or memory to memory. This movement of data by a separate entity significantly reduces the load on the processor. ADMA 2 controller can be used to save power in a system by putting the Central Processing Unit in a low-power state and using the ADMA 2 controller (fewer gates/transactions) to move the data.

Keywords - SD Host Controller, DMA, ADMA 2, Verilog

I. INTRODUCTION

The need for portable digital storage in Embedded Systems is increasing rapidly. With the increasing consumer digital content, demand for high capacity digital storage is increasing rapidly. Today, portable storage media's are widely used in all mobile phones, digital cameras, camcorders, and in many multimedia devices.Secure Digital (SD) cards are designed for portable storage applications. They have many advantages over their predecessors. They have high storage capacity and built upon NAND flash technology. SD cards have security feature built in for protecting digital contents.

ADMA 2 transfers data directly between an I/O device and memory (memory to memory ADMA transfers are also possible). Whichever CPU is being used, it must have a ADMA 2 feature to determine when ADMA 2 is required, so that it can relinquish control of the address and data buses, as well as the control lines required to read and write to memory.

II. SECURE DIGITAL HOST CONTROLLER (SDHC)

Secure Digital (SD) is the most widely used portable memory standard. Its ultra-compact and rugged architecture, simple interface, high security, low power consumption, reliable operation and interoperability have made it the de-facto solution for portable storage.

2.1 SD Bus Interface

In systems-on-a-chip and embedded systems, typical system bus infrastructure is a complex on-chip bus such as AMBA High-performance Bus. AMBA defines two kinds of AHB components: master and slave.

A slave interface is similar to programmed I/O through which the software (running on embedded CPU, e.g. ARM) can write/read I/O registers or (less commonly) local memory blocks inside the device. A master interface can be used by the device to perform DMA transactions to/from system memory without heavily loading the CPU. Therefore high bandwidth devices such as network controllers that need to transfer huge amounts of data to/from system memory will have two interface adapters to the AHB: a master and a slave interface. This is because on-chip buses like AHB do not support tri-stating the bus or alternating the direction of any line on the bus. Like PCI, no central DMA controller is required since the DMA is bus-mastering, but an arbiter is required in case of multiple masters present on the system. Internally, a multichannel DMA engine is usually present in the device to perform multiple concurrent scatter-gather operations as programmed by the software.

The Data transfers to and from the SD Memory Card are done in blocks. Data blocks are always succeeded by CRC bits. Single and Multiple block operations are defined. Data can be transferred using single or multiple data lines. The SD Host Controller is fully compliant to SD Host Controller Specification version 3.0 and Physical Layer Specification version 3.01. The standard register set is implemented. The Host Processor accesses the various registers and FIFOs in the Host Controller to transfer data between Host and SD Card.

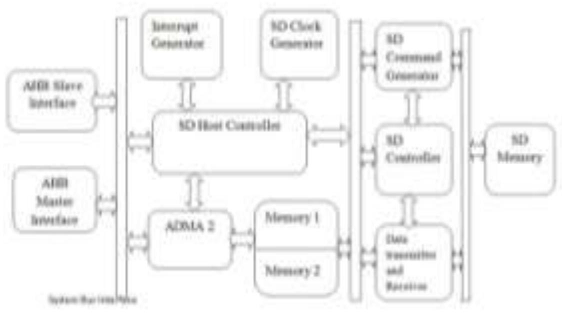


Fig 1.1: Basic block diagram of SDHC

Every data transfer is preceded by corresponding command and response. Command is send by the Host Controller to the Card and Response is send back by the Card to the Host Controller. The block diagram of Host Controller is given in Figure 1.1

III. DIRECT MEMORY ACCESS (DMA)

DMA is an essential feature of all modern computers, as it allows devices to transfer data without subjecting the CPU to a heavy overhead. Otherwise, the CPU would have to copy each piece of data from the source to the destination.

This is typically slower than copying normal blocks of memory since access to I/O devices over a peripheral bus is generally slower than normal system RAM. During this time the CPU would be unavailable for any other tasks involving CPU bus access, although it could continue doing any work which did not require bus access. A DMA transfer essentially copies a block of memory from one device to another. While the CPU initiates the transfer, it does not execute it. For so-called "third party" DMA, as is normally used with the ISA bus, the transfer is performed by a DMA controller which is typically part of the motherboard chipset. More advanced bus designs such as PCI typically use bus mastering DMA, where the device takes control of the bus and performs the transfer itself. A typical usage of DMA is copying a block of memory from system RAM to or from a buffer on the device. Such an operation does not stall the processor, which as a result can be scheduled to perform other tasks. DMA is essential to high performance embedded systems. It is also essential in providing so-called zero-copy implementations of peripheral device drivers as well as functionalities such as network packet routing, audio playback and streaming video.The alignment process subjects the CPU to a heavy overhead. There are some applications, like video processing that require an efficient support for unaligned accesses, degrading the performance significantly.

IV. ADVANCED DIRECT MEMORY ACCESS 2 (ADMA 2)

ADMA 2 is one of several methods for coordinating the timing of data transfers between an input/output (I/O) device and the core processing unit or memory in a computer. ADMA 2 is one of the faster types of synchronization mechanisms, generally providing significant improvement over interrupts, in terms of both latency and throughput. ADMA 2 controller can generate memory addresses and initiate memory read or write cycles. It contains several processor registers that can be written and read by the CPU. These include a memory address register, a byte count register, and one or more control registers The CPU then sends commands to a peripheral device to initiate transfer of data. The ADMA 2 controller then provides addresses and read/write control lines to the system memory. Each time a byte of data is ready to be transferred between the peripheral device and memory, the ADMA 2 controller increments its internal address register until the full block of data is transferred.

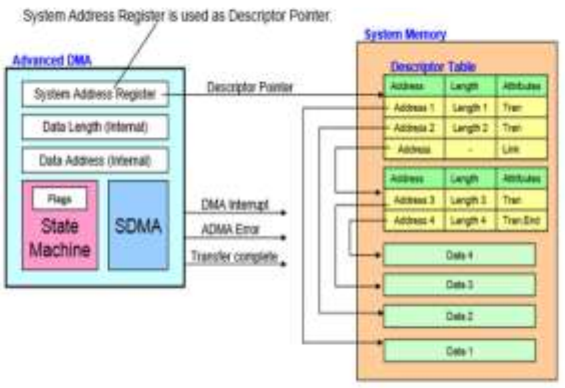


Fig 1.2: Block diagram of ADMA 2

The figure 1.2 shows the block diagram of advanced ADMA2.In ADMA2 operation a Descriptor Table is created in system memory by the Host Driver. 32-bit Address Descriptor Table is used for the system with 32-bit addressing and 64-bit Address Descriptor Table is used for the system with 64-bit addressing.

Each descriptor line (one executable unit) consists with address, length and attribute field. The attribute specifies operation of the descriptor line. ADMA2 includes SDMA, State Machine and Registers circuits. ADMA2 does not use 32-bit SDMA System Address Register but uses the 64-bit Advanced DMA System Address register for descriptor pointer. Writing Command register triggers off ADMA2 transfer. ADMA2 fetches one descriptor line and executes it. This procedure is repeated until end of descriptor is found.

An Example of ADMA2 Programming

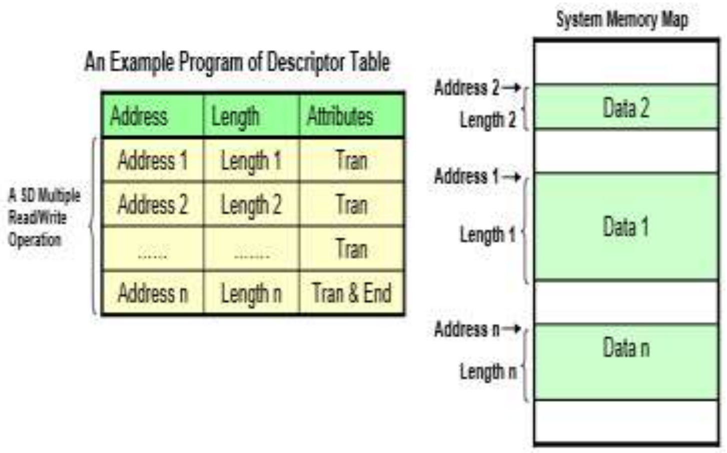


Fig 1.3: An Example of ADMA2 Data Transfer

Figure 1.3 shows a typical ADMA2 descriptor program. The data area is sliced in various lengths and each slice is placed somewhere in system memory. The Host Driver describes the Descriptor Table with set of address, length and attributes. Each sliced data is transferred in turns as programmed in descriptor.

4.1 ADMA2 States and its Operation

The figure shows various states of ADMA2 and the operation of all states. It also describes the condition when ADMA 2 fetches the descriptor line and transfers the data and the increment in the address. ADMA 2 adopts scatter gather DMA algorithm so that higher data transfer speed.

The Host Driver can program a list of data transfers between system memory and SD card to the Descriptor Table before executing ADMA. It enables ADMA to operate without interrupting the Host Driver. Furthermore, ADMA can support not only 32-bit system memory addressing but also 64-bit system memory addressing. The 32-bit system memory addressing uses lower 32-bit field of 64-bit address registers.

ST_FDS (Fetch Descriptor): ADMA 2 fetches a descriptor line and set parameters in internal registers. Next go to ST_CADR state.

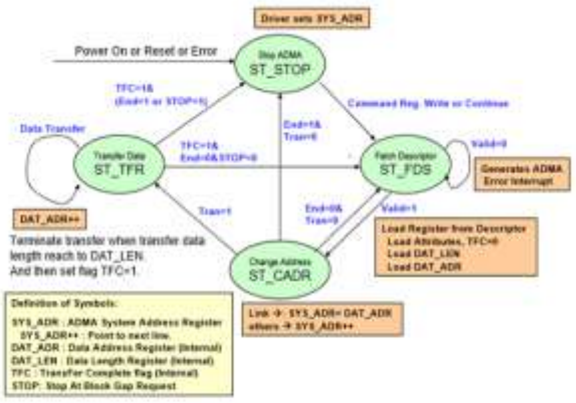


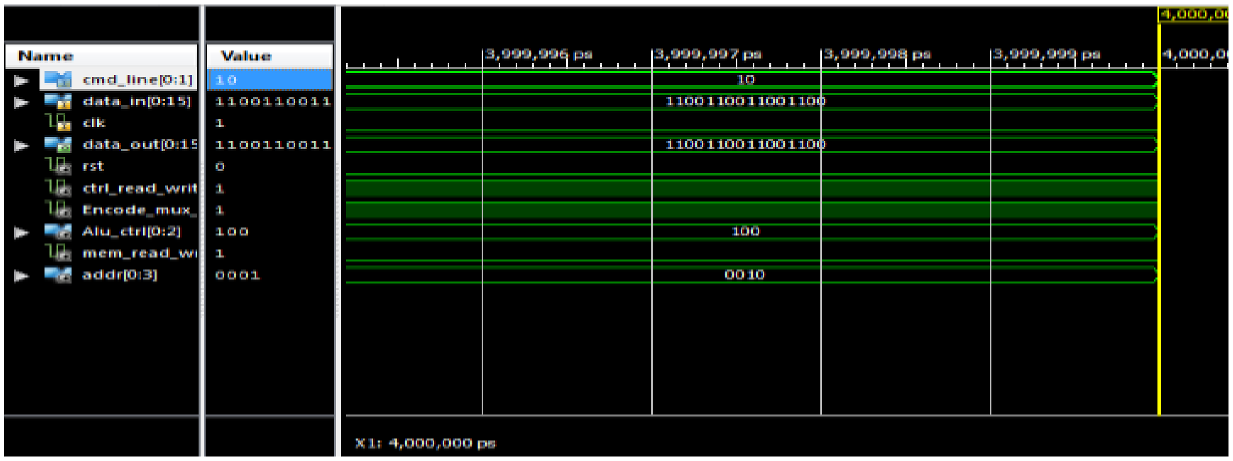
Fig 1.5: States of ADMA 2

ST_CADR (Change Address): Link operation loads another Descriptor address to ADMA 2 System Address register. In other operations, ADMA System Address register is incremented to point next descriptor line. If End=0, go to ST_TFR state. ADMA 2 shall not be stopped at this state even if some errors occur.

ST_TFR(Transfer Data): Data transfer of one descriptor line is executed between system memory and SD card. If data transfer continues (End=0) go to ST_FDS state.

ST_STOP (Stop DMA): After Power on reset or software reset. All descriptor data transfers are completed .Error occurrence during ADMA2 transfer may stops ADMA2 operation and generates ADMA 2 Error Interrupt. Error Status registers holds state of ADMA 2 stopped. The host driver can identify error descriptor location by following method. If ADMA 2 stopped at ST_FDS state, the ADMA 2 System Address Register points the error descriptor line. If ADMA 2 stopped at ST_TFR or ST_STOP state, the ADMA 2 System Address Register points the next location of error descriptor line. By this reason, ADMA2 shall not stop at ST_CADR state. ADMA 2 allows an I/O device to communicate directly to the main memory bypassing CPU in order to speed up memory operations..

V. SIMULATION RESULTS



Name	Value
cmd_line[0:1]	10
data_in[0:15]	1100110011
clk	1
data_out[0:15	1100110011
rst	0
ctrl_read_writ	1
Encode_mux_	1
Alu_ctrl[0:2]	100
mem_read_wi	1
addr[0:3]	0001

Fig 1.6 Write function for command line 10.

In figure 1.6 read operation is completed. It is said to perform write operation as the command line is changed from 01 to 10.

VI. CONCLUSION

The battery operated device which is one of the fastest growing electronic industries, in which speed of data transfer is becoming a major issue. With the proposed design of secure digital host controller intellectual property, we can achieve a high speed data transfer without compromising power. Simulated results for data transfer in ADMA 2 which provides high capacity data transfer with the reduction in load on Host CPU is obtained.

REFERENCES

[1].Technical Committee SD Association. (2011): SD Specifications Part A2 SD Host Controller Simplified.

[2].Specification Version 3.00. Retrieved February 25, 2011 from SD.

[3].Technical SD association: SD Specifications Physical Layer Simplified Specification Version 3.01. Retrieved February 18, 2011 from SD card.

[4].AMBA Advanced High Performance Bus Protocol specification http://www.arm.com.

[5].Chuan-Sheng Lin and Lan-Rong Dung, “A NAND Flash Memory Controller for SD/MMC Flash Memory Card”. IEEE Transactions on Magnetics, vol.43, No.2, February2007.

[6].P. Rashinkar, P. Paterson, Leena singh, “System-on-a-chip verification methodology and techniques,” Kluwer Academic Publishers, 2001.

[7]. SD Association: SD Memory Card Choices.

[8]. SD Association: Greater Performance Choices.