FPGA IMPLEMENTATION OF DDR SDRAM MEMORY CONTROLLER FOR SOC APPLICATION

Poornima Aldi1, Sadhana C2, Basavalinga Swami3

1M. Tech Student, Department of Electronics and Communication, Lingaraj Appa College of Engineering, Bidar.
2Associate Professor, Department of Electronics and Communication, Lingaraj Appa College of Engineering, Bidar.
3Assistant Professor, Department of Electronics and Communication, Lingaraj Appa College of Engineering, Bidar.

ABSTRACT

BACKGROUND

In today’s technology, memory plays a very important role. Every system or component have memory device for storage purpose. In that, highly efficient, highly speed memory has popularity. Still today, RAM and ROM are very familiar, but in RAM and ROM, the data transfer rate is less. These memories will carry data on rising clock side. It consumes more time to transfer data. To transfer data faster compared to RAM and ROM, here using DDR SDRAM (Double Data Rate Synchronous Random Access Memory), which transfers data faster than any other memory. DDR will transfer data on both rising and falling side of the data, hence minimizes the read and write access time. In this project, data is transferring from source to destination using Encoder, Decoder, Router, DDR SDRAM. The system on chip has many more advantages. It consumes less power, less area, more efficient, high performance, cost effective and reliable data.

KEYWORDS

System on Chip, DDR SDRAM, Encoder, Decoder, EDAC.


BACKGROUND

Memory is a small component, which stores the data permanently or temporarily. These components can be found in almost all the systems. Commonly used SRAM and DRAM has some disadvantages. To overcome disadvantage, DDR SDRAM memory controller is used. This DDR SDRAM will carry the data on both rising and falling side of clock cycle; hence, data transfer rate is faster. This memory controller will optimise the access time of the write and read cycle. This controller will automatically refresh the memory. It has set of inbuilt command, so reduces the processing time and increase the data transfer rate.

Here, transferring data from source to destination, some components are used and all the components are fixed on single chip that can be called as the system on chip. System on chip has more advantages over system on board. All the components can be fixed or mounted on single chip so that can call as the system on chip. It will consume less power, less area, cost will be low, high performance, integration level is very high, miniaturisation, these all are the main advantages of the system on chip.

The first aim of the project is to minimise the access time of read cycle and write cycle. Read and write access time can be minimised by using DDR SDRAM memory controller (Double Data Rate SDRAM). In DDR SDRAM, memory controller has some inbuilt commands. It decreases the length of the program and saves the time. This memory controller is designed to send data from source to destination. Receiving data from source and store the data, then send to the destination. While Sending data, data may get corrupted due to the radiation in atmosphere.

Financial or Other, Competing Interest: None.
Submission 08-09-2016, Peer Review 08-12-2016, Acceptance 14-12-2016, Published 20-12-2016.
Corresponding Author:
Poornima Aldi,
Plot No. 104,
Om Nagar, Sedam Road,
Gulbarga,
E-mail: aldi poopirma@gmail.com
DOI: 10.14260/jtasr/2016/39

Figure 1. Block Diagram

The above figure is the DDR SDRAM memory controller block diagram for the SoC application. The input data is given to the data in and it will send the data to the encoder block. Encoder block will receive the 16 bit data as input data and it will generate the parity bit by using 16 bit input data. For 16 bit input data, 6 bits of parity bit is generated in the encoder. The 6 bits of parity will be placed with input data by using hamming code. While sending the data from source to destination due to the Single Event Upset (SEU) in atmosphere, the data can be corrupted. Radiation in nature will create the error in original data. These error can be filtered by using one best solution that is error detection and correction. This 16 bit data will be converted into 22 bit including parity bit. This 22 bit data will be sent to the DDR block and it will store the data and transfers the data very quickly, that is on rising and falling edges also. Again, 22 bit of data is sent to the decoder block and this block will remove the corrupted data and also removes the parity bits, so the data will be again the 16 bit data. This 16 bit data will be received by the router. The router will route the data to select the proper processor for further process.1
Encoder

The encoder is the first block, which comes within the EDAC unit. It receives the given data input bits and generates the parity bits and then generates the output, which is a combination of both input data bits and parity bits. The 16 bits of input data are given as an input to the encoder. Using these input data bits, the parity generator will generate the parity bits using formula Parity Generator. The parity generator tasks the data input bits and generate the parity bits as follows:[3]

<table>
<thead>
<tr>
<th>Check Bits</th>
<th>Total Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>39</td>
</tr>
</tbody>
</table>

The generated parity bits are placed in the powers of two position in hamming table and input data bits are placed in the remaining blocks in order from LSB to MSB. The output of the encoder is the encoded bits of length 22-bits are passed to the FIFO.

FIFO

The FIFO is of 22-bit input and output ports. The input port is controlled by a free running clock and write enable pin (W). Data is written into the synchronous FIFO on every rising clock edge when the write enable pin is asserted. The output port is controlled by read enable pin (R). This FIFO has two fixed flags, FIFO Empty (FE) and FIFO Full (FF). FIFO full goes high when the FIFO is filled completely and FIFO empty goes high when the FIFO has no data in it.

These FIFOs are fabricated using high-speed submicron CMOS technology. The common problem may occur in space applications are in the Low Earth Orbit (LEO) stored digital data suffers from SEUs. These upsets are naturally induced by radiation. Bit flips are caused by SEUs is a well-known problem in memory chips. The other common problem may occur in the memory is “stuck at fault”. So, the error detection and correction techniques are an effective solution to this problem.

Decoder

The decoder reads the data from the FIFO unit. It detects and corrects the one bit error, which maybe in the FIFO unit. It also detects the two bit error maybe created in FIFO. The decoder performs the separation data bits and parity bits from the read data. Then, it generates parity bits and also generates the decoding bits in order to detect and correct error. The read input data is into the data bits and parity bits and they are stored in different registers. Now, it generates the parity bits using the data bits, which are stored in the register.

Parity Generator

The parity generator generates the parity bits from the data. A bit using the similar formula was described in the section. After the parity generation, need to the syndrome, will described in next subsection following.

Extended Hamming Code

For many applications, a single error correcting code be considered unsatisfactory, because it accepts all received. A SEC-DED code is safer and it is the level of correction and detection most often used in computer memories. The Hamming code can be converted to a SEC-DED code by adding one check bit, which is a parity bit (let us assume even parity) on all in the SEC code word. This code is called an extended Hamming code. The bits representation of extended hamming code given in the table 1.1.

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>Check Bits</th>
<th>Total Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Hamming Code Bits Representation

Design of EDAC Unit

Error Correction Codes (ECC) and Error Detection and Correction (EDAC) have been implemented in memory designs to tolerate faults and enhance reliability. Extra check bits (parity bits) have to be stored along with the information bits, so the hardware overhead includes the encoding/decoding circuit and the memory space for check bits. ECCA can protect the memory from attacks hard soft errors. The modified Hamming Code and Hsiao Code are the most used Single Error Correctable and Double Error Detectable (SEC-DED) codes. The word format is shown in Figure.[3]

Design of Router

The router receives 16-bit of input from the decoder. Output transmits the 14-bit of output data to the appropriate processor. Input of 16-bit data is separated as bits and data bits. From the MSB side, 2-bits are address and 14-bits are data bits. The address bits are used for the selection of the correct processor to transmit the data for the selected processor. Based on the value of the 14th and 15th bit of the address bits, the below processor is selected.

<table>
<thead>
<tr>
<th>14th and 15th</th>
<th>PROCESSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Reset Condition</td>
</tr>
<tr>
<td>01</td>
<td>Processor 1</td>
</tr>
<tr>
<td>10</td>
<td>Processor 2</td>
</tr>
<tr>
<td>11</td>
<td>Processor 3</td>
</tr>
</tbody>
</table>

Table 2. Processor Selector

Figure 2. ECC Code Word Format
Memory devices are almost found in all systems and nowadays high speed and high performance memories are in great demand. For better throughput and speed, the controllers are to be designed with clock frequency in the range of megahertz. As the clock speed of the controller is increasing, the design challenges are also becoming complex. Therefore the next generation memory devices require very high speed controllers like double data rate and quad data rate memory controllers.[4]

DDR SDRAM controller module receives addresses and control signals from the BUS Master. The controller generates command signals and based on these signals, the data is either read or written to a particular memory location.

**The DDR SDRAM Controller Architecture is shown in Figure. It consists of Three Modules;**

1. Main Control Module.
2. Signal Generation Module.
3. Data Path Module.

**Main Control Module**

The DDR SDRAM Controller has to undergo an initialisation process by a sequence of command signals before the normal memory access. Whenever reset signal is high, the initialisation FSM will switch to idle state. Once the reset signal goes low, the controller has to wait for 200us clock stabilisation delay. This is constantly checked by sys_dly_200us signal and a high on the sys_dly_200us, the clock delay is complete.

The DDR initialisation sequence will begin immediately after the clock power stabilisation is completed. These refresh commands are generated to refresh the DRAM memory.

**Data Path Module**

The main control module has two state machines and a refresh counter. The two state machines are for initialisation of the SDRAM and for generating the commands to the SDRAM. They generate iState (initialisation state) and cState (Commands State) outputs according to the system interface control signals. The signal generation module now generates the address and command signals depending upon the iState and cState. The path module performs the read and operations between the bus master and DDR.

**Following are some of the important features of DDR SDRAM Controller**

- The DDR SDRAM Read and Write operations are simplified by the controller.
- For initialising, the DDR SDRAM controller, separates state machines internally.
- The access time for read and the write cycle is optimised based on the latency and burst length of the DDR SDRAM.
- The auto refresh for the SDRAM is done by the controller.

**The main Control Module Consists of Three Submodules**

- Initialisation AFSM module (INIT_FSM).
- Command AFSM module (CMD_FSM)
- Counter A module.

**Signal Generator**

This signal module generates signal for the DDR SDRAM those signals are command signal. This command signal
includes the addresses for particular data it can write as DDR add, to read and write operation, also contains the adders of the column and rows. It can also be written as DDR_CASn and DDR_RASn, respectively.

CONCLUSION
In the project, the design of the encoder with the EDAC (Error Detection and Correction) is done successfully. Input data is encoded by this DDR SDRAM controller. It is also designed for transferring of data is on both cycles of clock that is rising clock side and falling clock side clock. Designing of decoder and router is also done. In most of the SoC, design implementation of the components is done separately here and the implementation of all components are done on single chip. All the blocks are designed by using VHDL and function of whole block and function of each block will be verified by Model Sim.

ACKNOWLEDGEMENTS
I am extremely pleased to present myself this paper after getting innumerable movements and hard time with a never-ending coordination. I sincerely acknowledge offering my words of thanks to our beloved and respected Sadhana C, associate professor, who helped me during the paper presentation.

REFERENCES