Implementation of Physical Design in DDR SDRAM Memory Controller for Digital T.V. Decoder

MANIKAYA HRIDAYA, SUSHMA MERCILIN¹, G. RAMACHANDRA KUMAR²

¹PG Scholar, Dept of ECE, Aurora’s Scientific & Technological Institute, JNTUH, Hyderabad, India, E-mail: sushma.mercilin@yahoo.com.
²Assoc Prof, Dept of ECE, Aurora’s Scientific & Technological Institute, JNTUH, Hyderabad, India, E-mail: ramachandra2035@gmail.com.

Abstract: The DDR SDRAM controller provides the user with a simplified interface to an industry standard memory device. Using this controller makes access to DDR SDRAM devices as simple as possible. This application note describes a DDR SDRAM controller design implemented in a set up box device. It targets a DDR SDRAM device at a clock rate of 200 MHz with data transfer at 400 MB/s. In this design, the memory controller is used to perform the correct initialization sequence and to transmit data to the SDRAM device, through the appropriate sequence of control and data signals. The read and write command receives from user interface and transmits to DDR SDRAM through controller. This work shows that the implemented system running at 100 MHz can achieve the necessary bandwidth to decode and exhibit HD 1080p resolution videos at 30 frames per second. In this Project, we propose the design and implementation of a setup box for the DDR SDRAM Memory Controller. Along with the design of memory controller to provide proper commands For SDRAM initialization, read/write accesses and memory refresh. DDR SDRAM uses double data rate architecture to achieve high-speed data transfers. DDR SDRAM (referred to as DDR) transfers data on both the rising and falling edge of the clock. This DDR controller is typically implemented in a system between the DDR and the Processor units.

Keywords: DDR SDRAM; Set-top box; Memory Controller.

I. INTRODUCTION

Brazilian Digital Television System (BDTS) is a technical standard for digital television broadcast developed in Brazil and it is based on the Japanese ISDB-T standard. To receive a transmission, a Set-Top Box (STB) may be used. A STB is a device that connects to a television and an external source of signal, turning the signal into content which is then displayed on the television screen or other display device. The Brazilian Digital Television System uses H.264/AVC, the latest video coding standard of the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG), as video codec. This state-of-the-art video coding standard out performs previous standards by employing bi-predictive motion estimation, spatial prediction and adaptive entropy coding techniques. The new compression techniques used in H.264/AVC has as penalty an increase in the memory traffic. Memory access optimizations are necessary to handle data in the reference frames while video is decoded and exhibited at real time. Architectures of video processing systems require a single interface to off-chip DRAM memory in order to achieve the necessary storage capacity at low cost [1].

In this context, double data rate synchronous DRAM (DDR SDRAM) [2] memories have large use in embedded systems because of their low cost and high data storage capacity. This work is part of a national effort to develop a set-top box compliant with BDTS. The set-top box contains components like audio decoder, video decoder, general processor, graphics processor and memory controller. This paper presents a multichannel DDR SDRAM memory controller design implemented as an IP to be used in the set-top box. It is an extension of the work developed in [3] that implements the multichannel memory controller for a H.264/AVC video decoder. Now, the memory controller is extended in order to handle a H.264/AVC video decoder, a general CPU and a graphics processor.

II. MEMORY UNIT

In this we can discuss about the DDR SDRAM memory.

A. Introduction to DDR SDRAM Memory Controller

The DDR SDRAM uses double data rate architecture to achieve high-speed data transfers. DDR SDRAM transfers data on both the rising and falling edge of the clock. The DDR [4] controller is typically implemented in a system between the DDR and the bus master. Figure 1 shows the relationship of the controller between the bus master and the DDR. The bus master could be either a microprocessor like the Intel i960 or a user’s proprietary module interface.

B. Features of DDR SDRAM Memory Controller

The Double Data Rate (DDR) Synchronous Dynamic Random Access Memory (SDRAM) Controller provides a
simplified interface to industry standard DDR SDRAM [5] memory. The DDR SDRAM controller supports the following features, SDRAM burst lengths of 2, 4, or 8, CAS latency of 1.5, 2.0, 2.5, or 3.0.16 bit programmable refresh counter used for automatic refresh. It support for 2 banks of SDRAM devices. It supports NOP, READA, WRITEA,AUTO_REFRESH, PRECHARGE, ACTIVATE and LOAD_MR. Data Mask lines are supported for write operations. It supports SDRAM data path widths of 16, 32, and 64 bits. It utilizes two DLL’s to increase system performance. It provides a peak performance of 1.6GB/s with a system clock of 100MHz and a 64-bit. It simplifies SDRAM[6] command interface to standard system read/write interface. Internal state machine built for SDRAM power-on initialization. Read/write cycle access time optimized automatically according to the SDRAM timing spec and the mode it’s configured to. Dedicated auto-refresh request input and acknowledge output for SDRAM refresh. Easily configurable to support different CAS latency and burst length. Differential clock inputs (CK and CK#).Commands entered on each positive CK edge. DQS edge-aligned with data for READs; centre aligned with data for WRITE. DLL to align DQ &DQS transitions with CK 4 internal banks for concurrent operation.

C. Synchronous Dynamic Random Access Memory
SDRAM is high-speed Dynamic Random Access Memory (DRAM) [7] with a synchronous interface. The synchronous interface and fully pipelined internal architecture of SDRAM allows extremely fast data rates if used efficiently. SDRAM is organized in banks of memory addressed by row and column. The number of row and column address bits and number of banks depends on the size of the memory. SDRAM is controlled by bus commands that are formed using combinations of the RASN, CASN, and WEN signals as shown in Figure2. For instance, on a clock cycle where all three signals are high, the associated command is No Operation (NOP). A NOP is also indicated when the chip select is not asserted.

III. SET TOP BOX ARCHITECTURE
A. Set Top Box
This work focuses on a set-top box being developed to be compliant with BDTS. It is a prototype being developed as a national effort. Figure3 illustrates the STB architecture.

Figure1: DDR SDRAM Controller System.

Figure2: DDR SDRAM Controller System Level

C. Synchronous Dynamic Random Access Memory
SDRAM is high-speed Dynamic Random Access Memory (DRAM) [7] with a synchronous interface. The synchronous interface and fully pipelined internal architecture of SDRAM allows extremely fast data rates if used efficiently. SDRAM is organized in banks of memory addressed by row and column. The number of row and column address bits and number of banks depends on the size of the memory. SDRAM is controlled by bus commands that are formed using combinations of the RASN, CASN, and WEN signals as shown in Figure2. For instance, on a clock cycle where all three signals are high, the associated command is No Operation (NOP). A NOP is also indicated when the chip select is not asserted.

III. SET TOP BOX ARCHITECTURE
A. Set Top Box
This work focuses on a set-top box being developed to be compliant with BDTS. It is a prototype being developed as a national effort. Figure3 illustrates the STB architecture.

Figure3: Set Top Box Architecture.

The demultiplexer module (Demux) receives a Transport Stream (TS). The demultiplexer divides the TS in audio, video and data in format the modules: audio decoder, video decoder and CPU, respectively, through bus. Some data, e.g. legends, are sent directly to the graphic processor. Three modules access the DDR SDRAM memory: CPU, H.264/AVC video decoder and graphic processor. These three modules are connected to the multichannel DDR SDRAM memory controller. The CPU is a general processor and processes user inputs and software that comes in transport stream. It can also run a middleware. The video decoder decodes the H.264 frames and stores the decoded frames in the Reference Picture Buffer (RPB) memory, which is a region on DDR SDRAM memory reserved for frames storage. The Motion Compensator (MC) [8] module and filter are responsible for the video decoder accesses to memory. Motion Compensator contains a local tri-dimensional cache structure that is used to store the requested reference pixels while the reconstruction process is executed. As the video decoding process has an unpredictable behavior, the MC module can
access the main memory at different data rates. Also, the region of pixels used to reconstruct the image can be different in each decoded macro block.

The MC cache uses an addressing scheme based on: the x and y pixel block coordinates and the reference picture number. The memory controller uses this information to send a region of pixels to the MC with size 32x16 luminance pixels and two 16x8 chrominance pixels. The filter output generates decoded pixels in a sequence of macro blocks. The filter does not generate any addressing information and the decoded pixels are stored in the reference picture buffer indexed by the memory controller. Graphic processor takes decoded frames, combines them with an overlay and sends the new generated frame to a display device. It works by reading two lines: one of the actual frame being exhibited and the other from an overlay. It can also store a new overlay for future use. Images are stored in the form of macro blocks of pixels (MBs) in the Reference Picture Buffer (RPB) and the granularity of data transfers to the reference memory is a macro block organized in Luminance (Y) and Chrominance (Cb and Cr) information. The CPU has no defined granularity, since it works with other data than macro blocks.

**IV. MULTICHANNEL SDRAM CONTROLLER**

DDR SDRAM memories provide only one bus for data transfers, thus it is necessary a multichannel memory controller, whose purpose is to guarantee Quality of Service (QoS) between the processing units accessing external memory. The controller needs to provide fast access permissions to read or write shared data during the decoding process. It also needs to split the available data channel bandwidth in order to guarantee real time decoding by providing the necessary bandwidth that each PU need for their data requirements. The implemented multichannel DDR SDRAM memory controller is divided in an arbiter module, a data-path module and a DDR SDRAM memory controller, implemented as an IP in [6]. It is not a new design, but an extension of the multichannel memory controller implemented in [3]. The original design works with the video decoder and a simple video output module. In this work, it is extended to support the video decoder, the graphics processor and CPU.

The video output module used in [3] is replaced by the graphics processor. Figure 2 shows a block diagram of the proposed memory controller. If a PU A needs to access memory, it sends an access request. Arbiter then decides, according to a policy, if A will have the channel assigned to it or not. If arbiter decides to assign the channel to A, it verifies if there is another PUB accessing memory and asks it to leave. When B leaves, arbiter sends a permission signal back to A, indicating that A can access memory. All accesses and permissions signals goes through data path, that provides an interface communication for PUs and arbiter to communicate. This interface makes possible to replace an arbiter with policy X by another with policy Y without changing other parts of the memory controller, including the connections with the processing units.

**A. Buffers**

DDR SDRAM memories can be seen as a group of banks, each bank behaving like a matrix with rows and columns.DDR SDRAM memories provide linear space by an address.

![Figure 4. Multichannel memory controller architecture, PUs and buffers](image)

Scheme that combines the bank, row and column where data is stored. Before one can read or write, it is necessary to activate the bank and row that contain the data using a series of commands. This setup needs some clock cycles to take place and, during the setup phase, no data can be transferred. Once setup is done, it is possible to transfer data in bursts, sending or receiving 2, 4 or 8 data words in each memory access as long as data is stored in the same row. As it is impossible to transfer data in the memory setup phase, it is necessary to minimize the time spend in setup in order to maximize the amount of data that can be transferred. Buffers can help increase maximum data transfer if PU has some spatial or temporal locality by reducing unnecessary repetitive accesses to memory. The processing units involved in the video decoding process and exhibition present a high degree of both spatial and temporal locality. The spatial and temporal locality of CPU depends from the program being executed. Each processing unit has a buffer attached to it. Figure 4.1 shows the multichannel memory controller, processing units and buffers. MC and deblocking filter have buffers with capacity to store nine macroblocks. Graphics processor has 4 buffers. Two of them are used to store the lines of a frame and overlay to be exhibited. The other two are used to store the generated line of a new overlay and the next line of frame to be exhibited. Each of these buffers can hold an entire line of macroblocks of a 1920x1080 frame. CPU has a simple buffer with capacity for 8KB.

**B. Memory Layout**

To take advantage of buffer utilization and DDR SDRAM’s bursts transfers, it necessary to store data in a way that related data are together. Some memory layouts were defined to store decoded frames and the actual frame being exhibited. Motion compensation process takes...
macroblocks on a reference frame to decode the actual macroblock. The coordinates of macroblocks in reference frame are similar to the one being decoded. The next macroblock to be decoded may have in common some search area search with the previous decoded macroblock. To take advantage of this, decoded macroblocks are stored in a tiled way.

Table. Required Bandwidth

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>Filter</th>
<th>Graphics Processor</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>254MB/s</td>
<td>89MB/s</td>
<td>overlay + frame</td>
<td>overlay</td>
<td>not known</td>
</tr>
</tbody>
</table>

When anew search area needs to be fetched only the necessary macroblocks are updated. Graphics processor, on the other hand, needs data to be stored in a raster scan order, this leads to duplication of frames, with the same frame being stored in both tiled and raster scan way. While MC reads data in a tiled manner, graphics processor reads the same frames in a raster manner. Grouping related data together can reduce 86% of redundant accesses to memory as presented in [7], for those propose three access schemes in order to minimize DRAM access latency.

C. Data Requirements

Table I shows the necessary bandwidth; each PU needs to achieve in order to decode and exhibit at real time. CPU data requirements are unknown because it depends from the program that is going to execute.

V. RESULTS AND ANALYSIS

A. Simulation results

The functionality is checked, whether the design is working correctly or not, to the input the parallel output is correct or not is shown in fig 5.

B. Synthesis results

In this stage after applying the constraints to the design a technology mapped gate level netlist is obtained as output, and also the area, timing, qor reports are generated at this step of the procedure. The netlist schematic is shown in fig6. Netlist schematic is the representation of the design in the form of gates and flip-flops.

C. Floor planning Results

The reports that are created in the floor planning does not vary as compared to that of the synthesis because no extra
Implementation of Physical Design in DDR SDRAM Memory Controller for Digital T.V. Decoder

cells will be added in between these steps. In this stage only core utilization, aspect ratio, blockages, interconnect length, virtual placement can be done in this stage. But the cells and the net length area of the design remain 0 so the report does not change from synthesis to floor plan. The screen shot of the floor plan stage is shown in the figure8. The block which is surrender by the blue line is the core all the cells need to be placed in the core. The blue colours i.e. present are the ports of the design and the components that are present outside are the cells.

D. Placement Results
Placement is a step in the Physical Implementation process where the standard cells location is defined to a particular position in a row. Space is set aside for interconnect to each logic/standard cell. After placement, we can see the accurate estimates of the capacitive loads of each standard cell must drive. The tool places these cells based on the algorithms which it uses internally. It is a process of placing the design cells in the floorplan in the most optimal way. The main of the placement algorithm is making the chip as dense as possible (Area Constraint), minimize the total wire length (Reduce the length for critical nets). The number of horizontal/vertical wire segments crossing a line. Constraints for doing the above is that placement should be routable (no cell overlaps; no density overflow) and timing constraints are met. In the placement the total number of cells increases due to additional buffers and inverters increases due to changes in the length and width of the soft macro cells, and also the net length of the cells adds to the area of the design, the screen shot of the placement stage is shown in the fig9. The lines that are highlighted are the straps and the line present on the outer layer of core is the power rings, in this design blue represents VDD and pink represents the VSS.

E. Clock Tree Synthesis Results
After placing all blocks after the placement an actual clock is assigned to the design and also a clock tree will be formed accordingly replacing the virtual clock which has been in the design till now, and the clock tree bifurcates to individual blocks. The screen shot of the CTS stage is shown in the figure10.

Figure10: Clock Tree Synthesis.

In the timing report the clock changes its state from ideal to propagated so the real time delays will be added to the design, there will be a variation in the slack. In the figure 5.6 the yellow line represents the level 0 logic and to the right side represents the number of cells that are connected to the level 0 of the clock. Pink represents the level 1 of the clock, blue represents the level 2 of clock, orange represents the level 3 of the clock, violet represents the level 4 of the clock. This design contains 5 logical levels of the clock, as the logical level of the clock increases the clock buffer strength decreases.

Figure11: Routing.

F. Routing results
Here in this step the total area is maximum when compared to the remaining all steps because all the interconnects, net lengths, wire loads, vias and also extra buffers and inverters will be added to the design. Due to this the slack again decreases, area and the number of cells in the design increases. All the different wires in the library
are used so that all the cells in the design are interconnected to each other. A different color in the figure represents the different metal layers that are used to interconnect the cells. The below figure shows the routing. After the routing process is completed the major factor which has to checked is congestion which is shown in the above report, and we can see that the GRC (Global Routing Congestion) is 0(zero). Wire routing congestion is a leading cause of late and underperforming chips designs.

VI. CONCLUSION

The decoding process requires a high volume of data and an external memory with an efficient memory hierarchy is necessary to allow real time decoding. Buffers and memory layouts help to achieve the necessary bandwidth by avoiding redundant access to external memory. The modules share the data bus and this leads to the necessity of multiplexing memory channel. In this work it was presented a multichannel memory controller that provides the necessary bandwidth for each module. Experiments showed that a priority based arbitration scheme with preemption enables real time decoding, but provides low bandwidth for CPU. The implemented system can run at 100 MHz, the necessary frequency to achieve real time decoding. Future work include the study of different memory addressing organizations, use of the others memory’s banks (this work store all data in only one bank) and different memory layouts. Designing a sophisticated VLSI system of any kind is a complicated task. Completing a design can take over one year from system conception to fabricated chips. In this project I have verified the functionality of this design and performed the basic synthesis to get the technology mapped netlist with optimized timing. The main phase of the project starts from this point, with the number of transistors that can be involved and the challenges posed and each process node, no one person or team could manually design and complete a system within the given time frames required to be competitive. Here comes the importance of EDA tools. For this particular design I have used IC Compiler which is a Physical Design tool. The physical design is the process of transforming a circuit description into the physical layout, which describes the position of cells and routes for the interconnections between them. Giving the technology mapped netlist as input and performing the step by step process of floor planning, placement and routing, I have tried to optimize the timing, area and power of this particular design by comparing the reports generated at each step of the design flow.

VII. REFERENCES