FPGA DESIGN IMPLEMENTATION ON DDR
AND B-RAM FOR STRING COMPARISON

Mohammed Zaheer¹, A.M Khan²

¹Research Student Department of Electronics Mangalore University, Karnataka (India)
²Professor Department of Electronics Mangalore University, Karnataka, (India)

ABSTRACT
In this paper comparison of strings between the DDR and B-RAM memory is discussed. Embedded system is hard
ware and software mixed design to perform a specific application. The design is implemented on Spartan 6 Atlys
board consisting of memories like B-RAM, DDR, Cache, UART etc. B-RAM is a static memory which stores the
data temporarily. MPMC is customized memory controller that supports DDR memory. Using XPS tool the
embedded processor Microblaze is configured inside the Spartan 6 FPGA. The data is transferred to the
memory through UART IP which is connected to RS 232 serial port.

Keyword- DDR, BRAM, FPGA, MPMC.

I. INTRODUCTION
The words string comparison is mainly associated with the programming concept of microprocessor and
microcontroller. Primarily codes are developed in the form of assembly level language or using C language. If
the same codes are carried over extended C language to develop a hardware-software co-design then, the result
will be valid though the hardware implemented and not only by simulation. Embedded system is a technique
were HW/SW component work together to perform a specific application. Embedded system has variety
applications and play an important role in our daily life. They are found in automobiles, medical field, industrial
control system and different areas of entertainment electronics. The design of embedded system is done
depending on the features of microcontroller and its constraints. Presently HW/SW Co-design methodology is
preferred to design soft-core processor. Here designing each and every hardware component is an impractical
preposition which leads to the hardware complexity and costlier. Therefore a better way of designing the
hardware is through the designed and tested form of component called intellectual property (IP). These IP can be
synthesized on FPGA which are predesigned and pretested IP codes, which also helps the designer with lot of
flexibility. FPGA along with softcores processor provides designers with increased flexibility and debugging the
advancement in FPGA. Which has resulted in development of softcore processor for variety of application.
MicroBlaze is a 32-bit RISC Harvard architecture soft processor IP core with a rich instruction set
optimized for embedded applications [4]. The implementation is achieved using the Embedded Development
Kit provided by Xilinx, which helps to design the complete embedded processor more quickly and easily [7].
II. SOFT CORE PROCESSOR

Soft core processor is a hardware description language (HDL) Model of specific processor it can be customized for a given application and synthesized on ASIC FPGA target [1]. The softcore processors are easy to customize and they have the advantages of flexibility. These FPGA are more immune than logic circuit based technology. The architecture of required system and its behaviour are described at higher abstract level such as HDL. This makes the design more understandable and easy to design softcores processor. There are different supporting and development tools, they are AlteraNiosII, micro 32 and XilinxMicro blaze and Picoblaze. These tools offer logic elements and intellectual property for development of system on microcontroller chip. UART, memory controller, interrupt controllers, Ethernet controllers, timers these cores are the components in the form of IP used to create larger or complex system.

III. HARDWARE PLATFORM

FPGA is a very versatile unit which can be used for initializing the cores on the chip memory. The concept of placing all these essential components on a single chip refers to system on programmable chip as shown in fig1 [3] they consist of GPIO push button, PIP switch, led, HDMI video, Gigabit Ethernet, 128mb DDR memory, an USB and audio ports. These make the platform an ideal system to develop a SOPC.

IV. DDR CONTROLLER BLOCK

DDR memory provides double data rates by transfer of data. These DDR memories are high speed and low power. Multiport memory controllers know as native port interface. This type of interface are used when H.W needs bandwidth and low latency transfer to off chip memory. The DDR memory can manage a large amount of data. DDR is a high speed synchronous dynamic random access memory with eight banks [1].
V. MULTI PORT MEMORY CONTROLLER
FPGA consist of DDR3/DDR2/DDR SDRAM which is a high speed synchronous dynamic random access memory with eight blank [8]. It uses the double data rate architecture to achieve high speed. MPMC provides an access of 1GB memory available on the board. An external DDR memory is used where each port can be chosen from personality interface module. That connects to the micro blaze processor using PLB (processor Local Bus).

VI. BLOCK RAM
B-RAM is a configurable memory module [9]. Block Ram performs FIFO process. They are useful as buffer to capture and retrieve the data as per the designer convince. This Block RAM are having better timing approach which gives out the best result compared with any other type of FIFO memory. Block RAM structural is configured through BRAM interface controller IP.

VII. XPS UART LITE
The UART (Xilinx IP) core handles I/O to and from the system. It is connected to have a feature of one transmit and one receive channel in full duplex mode. It is connected to the off chip RS 232 serial port.

VIII. MICRO BLAZE PROCESSOR
In present trend electronic design automation industry particularly FPGA vendors provide pre-tested and pre-designed soft processor cores [2]. Towards customization & re-configurability and emulation. Manufactures develop these IP cores for specific FPGA solutions. One of the popular software in EDK industry is 32 bit RISC core given by the Xilinx known as micro blaze processor. In figure 2 the design of Hardware block is performed using Xilinx platform studio. The hardware block is developed using the predesigned and pretested IP.

![Diagram of MPMC Module Interface](image_url)

Fig-2: Design of Hardware Block

IX. IMPLEMENTATION
In the figure 1, the hardware designed module is implemented using Xilinx EDK platform. The software design, which includes Software development kit (SDK), is used to write the application for data transfer. The test data are placed in BRAM module and DDR memory port address. The first string data is written in BRAM memory location through UART. The second string is written into the DDR memory through UART. The data transferred in two different memories are seen on the terminals of SDK. With the help of micro blaze processor, the two strings in the BRAM and DDR memory location are compared to check whether the data strings to be tested are similar or dissimilar.

**X. FLOW CHART**

The figure 3 describes the flow chart which explains about the comparison of two strings stored in the BRAM and DRAM memory location. These memories are communicated through the UART. The process is continued to enter the strings in two different memory location. The string is viewed on the terminals of SDK. If the string present in the DDR matches with BRAM, this results out in the statements like string similar or dissimilar string.

**XI. CONCLUSION**

This comparison of string in two different memory locations of BRAM and DDR is processed through H.W/S.W co-design to develop an Embedded System as the string comparison. It can be done through simulation,
which helps in determining the output through software. But design of H.W/ S.W results out in real time application with more accuracy and practical applications. The figure 4 describes the table about the device utilization; the total of fully used LUT-FF pairs with help of IP is 46% only which shows the effective use LUT and Flip Flop present.

<table>
<thead>
<tr>
<th>Slice Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of LUT Flip Flop pairs used</td>
<td>4,796</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number with an unused Flip Flop</td>
<td>1,542</td>
<td>4,796</td>
<td>32%</td>
</tr>
<tr>
<td>Number with an unused LUT</td>
<td>1,037</td>
<td>4,796</td>
<td>21%</td>
</tr>
<tr>
<td>Number of fully used LUT-FF pairs</td>
<td>2,217</td>
<td>4,796</td>
<td>46%</td>
</tr>
<tr>
<td>Number of RAMB16BWERs</td>
<td>13</td>
<td>116</td>
<td>11%</td>
</tr>
</tbody>
</table>

**Fig.4— Device utilization**

**REFERENCES**


