CSP PROJECT

VIRTUAL FPGA

Working with Microblaze on Alpha Data board

Tarundeep Singh **Yogesh Kumar**

2008CS10195 **2008CS10197**

Sandeep Kr Bindal **Ankit Kr Jain** **Anuj Chauhan**

2008CS50536 **2008CS10157** **2008CS10162**

Contents

1. Specifications

Steps are being taken by the IITs to virtualise the available resources make them available to other colleges. As part of these efforts we aim at developing some experiments involving FPGAs to demonstrate computer architecture concepts such as caching, pipelining etc.

With the help of FPGAs installed in the lab machines using the PCI Interface students from remote locations will be able to perform experiments and see the results on their own systems.

We will first design a setup using Microblaze running on the AlphaData board containing Virtex II FPGAs. The setup will consist of

- Microblaze running on the FPGA chip
- Instruction and Data memory on ZBTRAM
- Instruction and Data cache on BRAM
- Pipelined processor design
- Linux operating system (ucLinux) running on Microblaze

Thus the experiments that we will design on the above setup will test the following architecture related concepts:

- Impact of varying cache parameters such as set associativity, block size in perfomance
- Study stall cycles in a pipelined processor design

The final deliverable will be an interface that allows students to experiment with computer architecture concepts using the infrastructure present remotely.

2. Hardware / Software

The complete block diagram for our project is provided in the next section. The Hardware and Software we will be using is as follows:

Hardware

- Alpha Data ADM XRC-II board containing Xilinx Virtex 2 FPGA

Software

- Xilinx ISE 9.1
- Xilinx EDK 9.1
- mb-gcc toolchain
- GTK Term / Hyper Terminal

3. Proposed block diagram and methodology

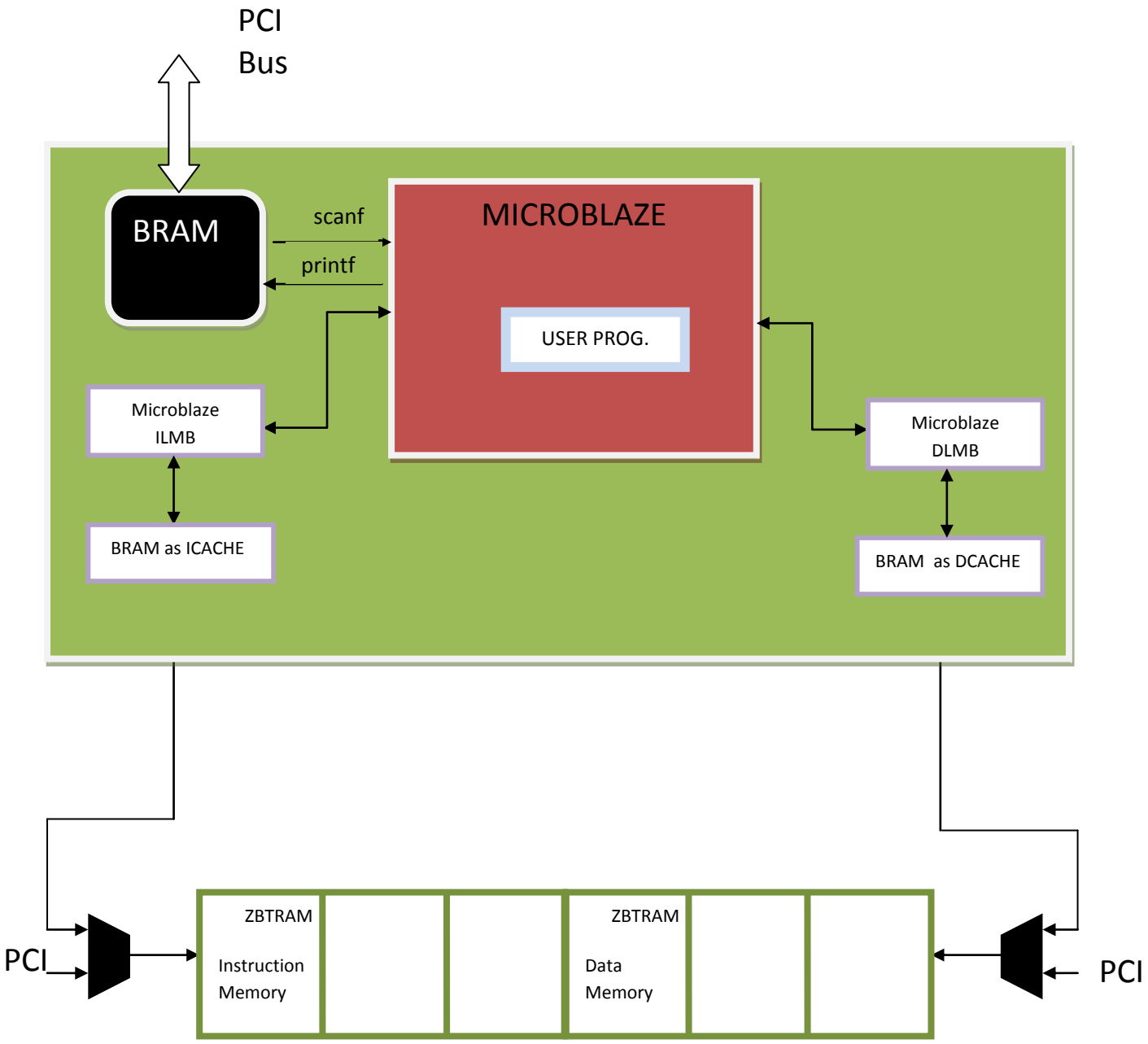

Figure 1: Block Diagram

Methodology

As we are continuing our project from the summers we already have a simple interface of memory modules and interrupt, polling that runs on Microblaze. Thus we plan to build our project layer by layer adding functionalities for cache, pipelining, OS, UART etc. to the basic design sequentially.

We have decided to work on incorporating OS on Microblaze later and the UART design in parallel after we have finished working on the cache and fully implemented it

The work on pipelining will begin after work on the cache has been finished, while virtualization will take place after the successful implementation of caching and pipelining.

3. Progress Report

Running UART (Yogesh Kumar, Sandeep Kumar Bindal)

- Successfully mapped UART pins to General Purpose Input Output pins
- Found TTL to RS-232 convertor to use as a level shifter but it failed
- Owing to the failure of the convertor and the unsafe nature of using external hardware on the board we are planning to shift our focus to another approach

We have been able to take input and print integers by changing the assembly code loaded in the BRAM and are now trying to automate the process by allowing C code to take input and print integers and strings as a replacement for printf and scanf statements.

Running Cache (Sandeep Kumar Bindal, Ankit Kumar Jain)

- Read cache implementation policies (write back/through)
- As suggested by Prof. Balakrishnan in the previous presentation we have decided to use Microblaze's own cache instead of developing our own cache.
- Currently we have been able to obtain the code for the ZBT-Controller with an interface for cache implementation for the Virtex 4 FPGA and are planning to use it for our own board by following the same approach.
- If the above approach does not work then we will use opb-ipif module. The opb-ipif is used for interfacing the
- Otherwise we can also use the opb-emc module to interface the ZBT RAM

Pipelining (Anuj Chauhan, Ankit Kumar Jain)

- Study microblaze's pipeline implementation

- Tracing the stall cycles for pipelined Microblaze

As stated in the earlier document and the timeline we will start work on pipelining when the work on cache has been finalized.

Monitor/OS (Ankit Kr Jain, Tarundeep Singh)

- We found the code for ucLinux (linux without MMU)
- After fulfilling the kernel dependencies in the linux code we were able to compile the code with the required modifications. Thus we generate the linux image for our board.
- Since our approach of the UART interface has changed and as suggested by Bala Sir we have decided to put our use of the OS temporarily on hold.

Virtualization (Anuj Chauhan, Tarundeep Singh)

- Presently we have the following configurations of our system :
    - Only the BRAM as Instruction and Data memory
    - Split BRAM – 1 for Instruction and 1 for Data memory
    - BRAM as Data memory and ZBTRAM as Instruction memory
    - ZBTRAM as Instruction memory and BRAM as Data memory
    - BRAM and ZBTRAM having combined address space.

- For the above configurations we are planning to allow the user to choose between the differing configurations and show the difference in performance by different parameters.
- When the cache is implemented we will consolidate all the status signals from cache along with the other performance indicators.
- We will create an interface for the end user that shows the above signals.

Website Updating and Reports (Anuj Chauhan)

- The website has been regularly updated with any relevant updates about the project.