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. Proposed block diagram and methodology

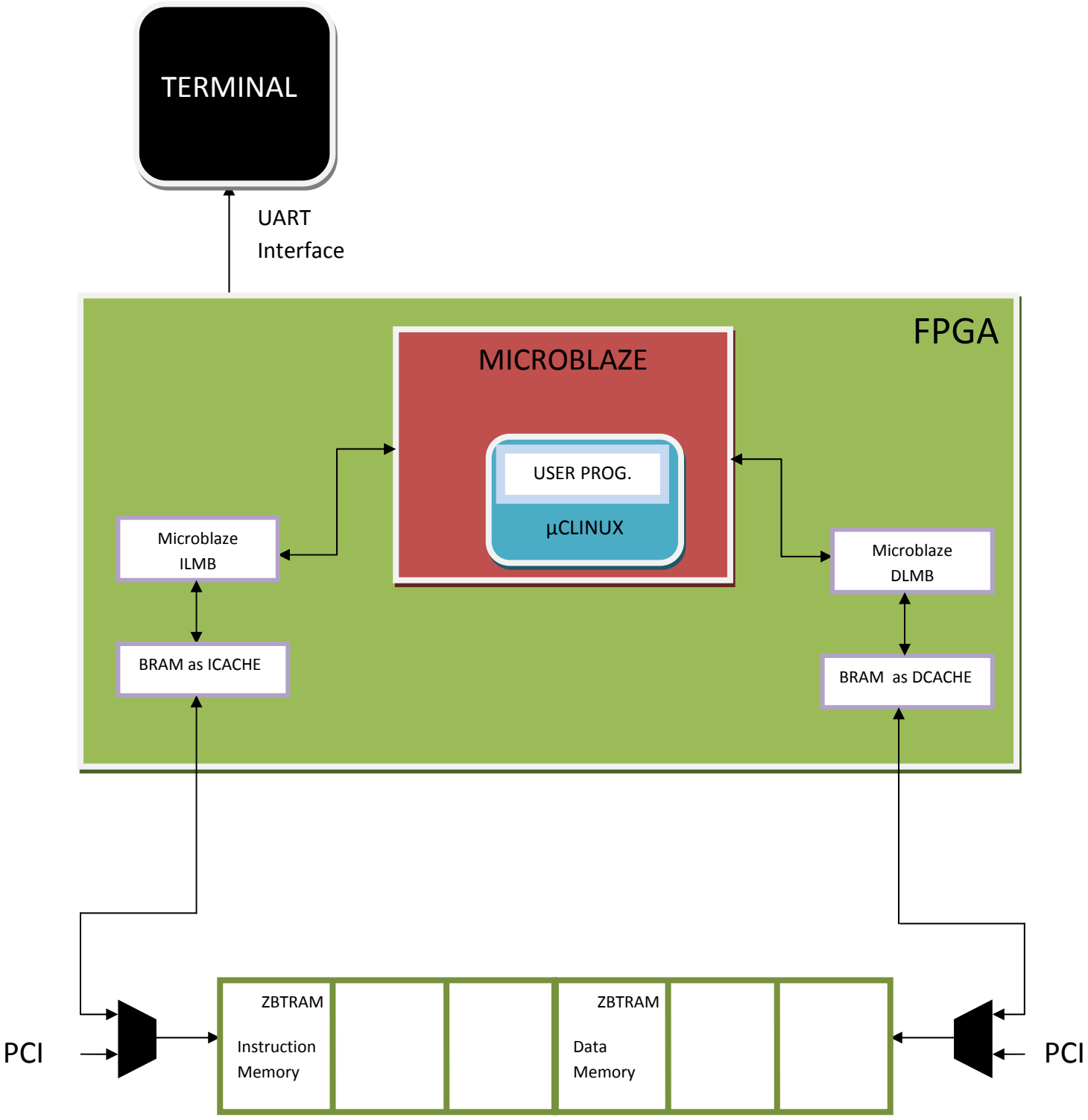


Figure 1: Block Diagram

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 will work on the running the OS on Microblaze and the UART design in parallel so that they can be integrated together to form an interface for the remote user.

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. Major components/blocks/kits etc.

(Status - availability, to be procured etc)

Our project requires the following hardware/software:

Board – Alpha Data ADM-XRC-II based on the Xilinx® Virtex-2 FPGAs

The **ADM-XRC-II** is based on the Xilinx® Virtex-2 FPGAs, provides up to 24MBytes of SSRAM and has an interface to allow *specialised IO modules* to be connected to the board. It is a *complete processing system*, *memory*, *I/O and Host interface* resources. It contains a *PCI to local bus bridge* providing the target design with a simple interface to the Host system.

- **Board Format :** PMC
- **Host I/F :** PCI
- **Target Device(s) :** Xilinx Virtex-2
- **SRAM :** 6 independent ZBT SRAM banks 256K/512K/1024K x 36
- **DRAM :** 256Mbyte via XRM-DDR
- **FLASH :** 16MByte

Status – The Alpha Data boards are already installed on two machines in the FPGA Lab. These machines can be accessed using SSH from a remote location too. User accounts on the above machines have already been created for our use.

Software – Xilinx EDK 9.1
Xilinx **I**SE 9.1

Status – Licensed copies of these softwares are available from the DHD Lab and have already been procured by us.

4. Major tasks and assignments

to individual group members

Running Cache (Sandeep Kumar Bindal, Yogesh Kumar)

- Reading cache implementation policies (write back/through)
- Testing basic cache code
- Adding cache to microblaze and testing
- Adding functionalities such as set associativity, block size
- Testing improved cache with microblaze
- Tracing the hit/miss data for the cache
- Study the impact of varying cache parameters on performance

Running UART (Yogesh Kumar, Sandeep Kumar Bindal)

- Mapping UART pins to GPIO using level shifter to connect it to the CPU
- Testing UART for simple printf statements
- Integrating the UART with the OS (ucLinux)

Pipelining (Anuj Chauhan Ankit Kumar Jain)

- Study microblaze's pipeline implementation
- Tracing the stall cycles for pipelined Microblaze

Monitor/OS (Ankit Kr Jain, Tarundeep Singh)

- Finding the code for ucLinux (linux without MMU)
- Studying kernel dependencies and the code
- Compiling the ucLinux code for microblaze
- Running it on microblaze
- Combining it with UART to create a terminal like interface for remote viewing

Virtualization (Anuj Chauhan, Tarundeep Singh)

- Consolidating all the status signals from cache, pipelining etc.
- Creating an interface for the end user

Website Updating and Reports (Anuj Chauhan)

5. Time line (PERT chart)

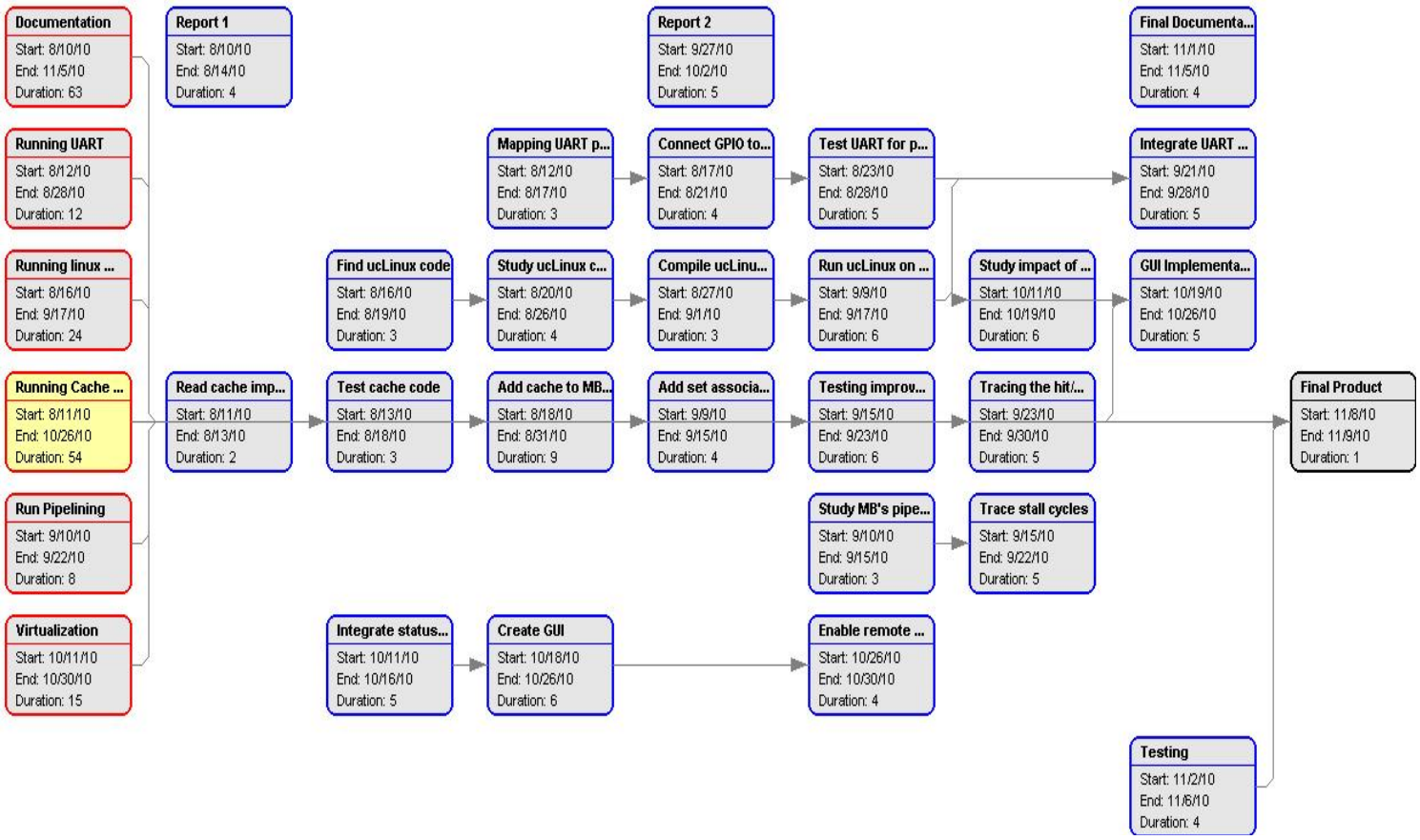


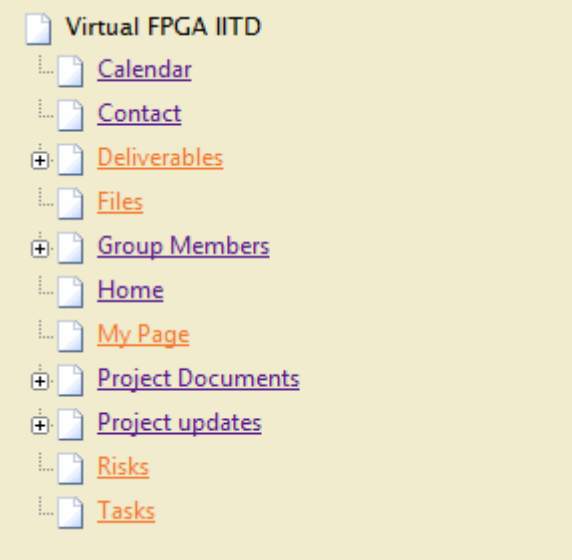
Figure 2: PERT Chart

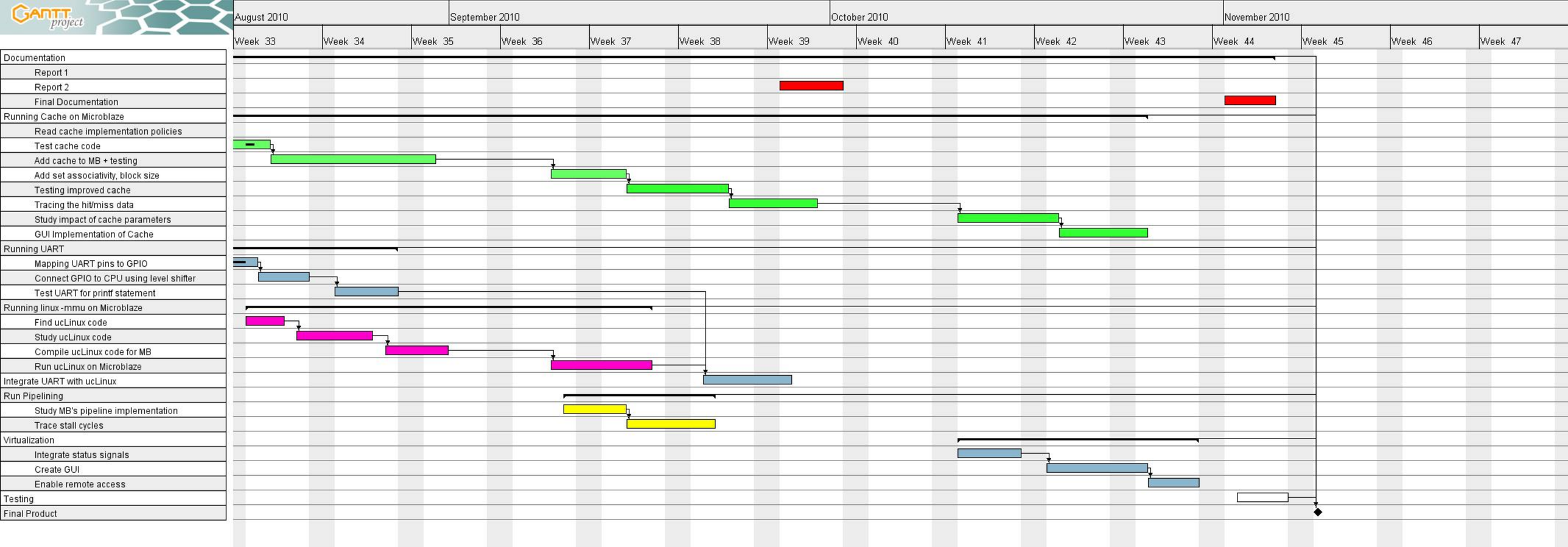
The detailed timeline is present in the attached document. It contains description of the work to be done along with their time schedules.

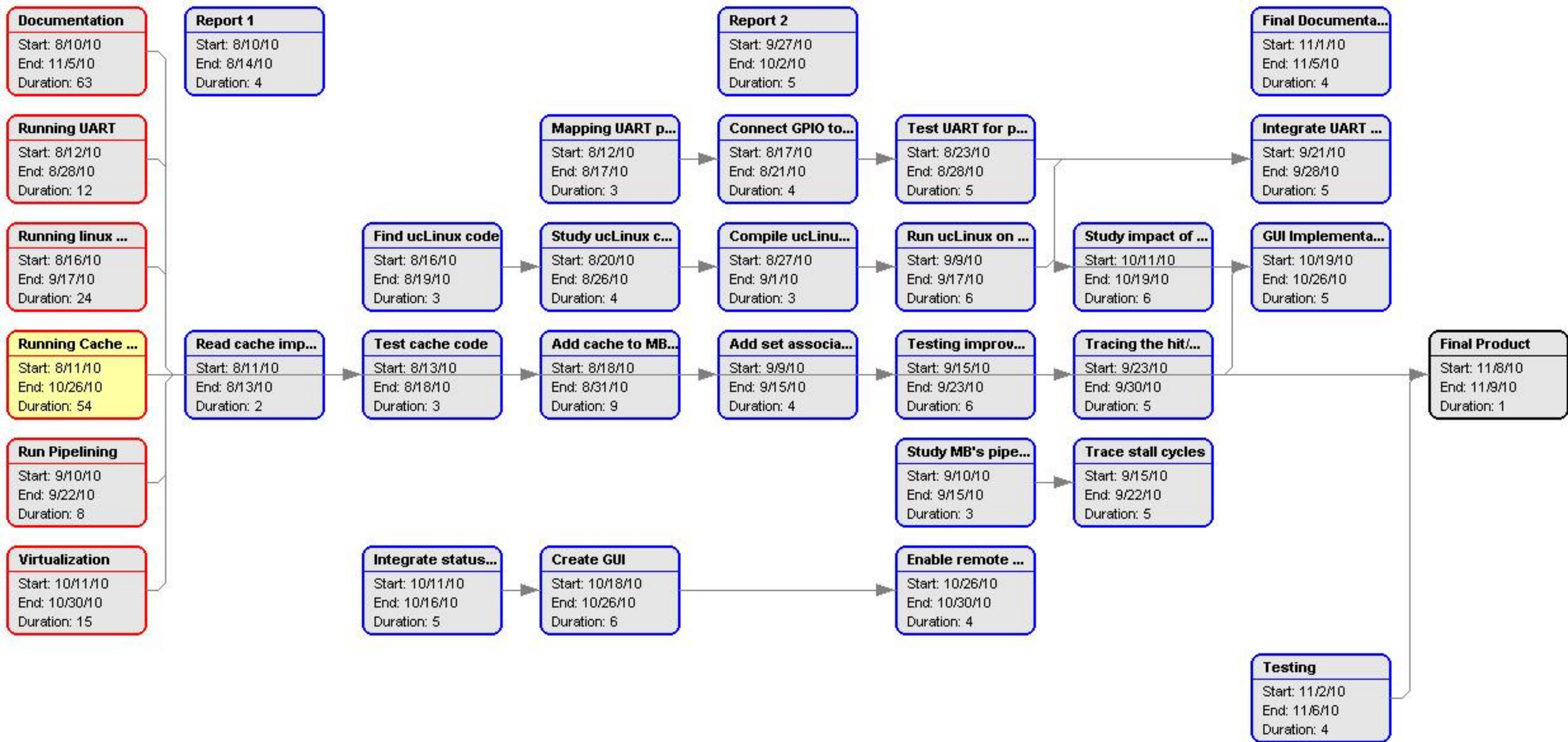
6. Weblink of our project

http://sites.google.com/site/iitdvirtualfpga/

The site contains all the required information related to the project.




GANTT project
August 2010
September 2010
October 2010
November 2010
Week 33
Week 34
Week 35
Week 36
Week 37
Week 38
Week 39
Week 40
Week 41
Week 42
Week 43
Week 44
Week 45
Week 46
Week 47
Documentation
Report 1
Report 2
Final Documentation
Running Cache on Microblaze
Read cache implementation policies
Test cache code
Add cache to MB + testing
Add set associativity, block size
Testing improved cache
Tracing the hit/miss data
Study impact of cache parameters
GUI Implementation of Cache
Running UART
Mapping UART pins to GPIO
Connect GPIO to CPU using level shifter
Test UART for printf statement
Running linux-mmu on Microblaze
Find ucLinux code
Study ucLinux code
Compile ucLinux code for MB
Run ucLinux on Microblaze
Integrate UART with ucLinux
Run Pipelining
Study MB's pipeline implementation
Trace stall cycles
Virtualization
Integrate status signals
Create GUI
Enable remote access
Testing
Final Product


Documentation
Start: 8/10/10
End: 11/5/10
Duration: 63
Report 1
Start: 8/10/10
End: 8/14/10
Duration: 4
Report 2
Start: 9/27/10
End: 10/2/10
Duration: 5
Final Documenta...
Start: 11/1/10
End: 11/5/10
Duration: 4
Running UART
Start: 8/12/10
End: 8/28/10
Duration: 12
Mapping UART p...
Start: 8/12/10
End: 8/17/10
Duration: 3
Connect GPIO to...
Start: 8/17/10
End: 8/21/10
Duration: 4
Test UART for p...
Start: 8/23/10
End: 8/28/10
Duration: 5
Integrate UART ...
Start: 9/21/10
End: 9/28/10
Duration: 5
Running linux ...
Start: 8/16/10
End: 9/17/10
Duration: 24
Find ucLinux code
Start: 8/16/10
End: 8/19/10
Duration: 3
Study ucLinux c...
Start: 8/20/10
End: 8/26/10
Duration: 4
Compile ucLinu...
Start: 8/27/10
End: 9/1/10
Duration: 3
Run ucLinux on ...
Start: 9/9/10
End: 9/17/10
Duration: 6
Study impact of ...
Start: 10/11/10
End: 10/19/10
Duration: 6
GUI Implementa...
Start: 10/19/10
End: 10/26/10
Duration: 5
Running Cache ...
Start: 8/11/10
End: 10/26/10
Duration: 54
Read cache imp...
Start: 8/11/10
End: 8/13/10
Duration: 2
Test cache code
Start: 8/13/10
End: 8/18/10
Duration: 3
Add cache to MB...
Start: 8/18/10
End: 8/31/10
Duration: 9
Add set associa...
Start: 9/9/10
End: 9/15/10
Duration: 4
Testing improv...
Start: 9/15/10
End: 9/23/10
Duration: 6
Tracing the hit/...
Start: 9/23/10
End: 9/30/10
Duration: 5
Final Product
Start: 11/8/10
End: 11/9/10
Duration: 1
Run Pipelining
Start: 9/10/10
End: 9/22/10
Duration: 8
Study MB's pipe...
Start: 9/10/10
End: 9/15/10
Duration: 3
Trace stall cycles
Start: 9/15/10
End: 9/22/10
Duration: 5
Virtualization
Start: 10/11/10
End: 10/30/10
Duration: 15
Integrate status...
Start: 10/11/10
End: 10/16/10
Duration: 5
Create GUI
Start: 10/18/10
End: 10/26/10
Duration: 6
Enable remote ...
Start: 10/26/10
End: 10/30/10
Duration: 4
Testing
Start: 11/2/10
End: 11/6/10
Duration: 4

GanttProject Report

Project : Virtual FPGA

Start : 8/10/10

End : 11/9/10

Organization : IIT Delhi

Web Link : http://sites.google.com/site/iitdvirtualfpga/

Description :

This Project is done as part of the CSP315 Course.

Date : Aug 14, 2010 4:36:17 PM

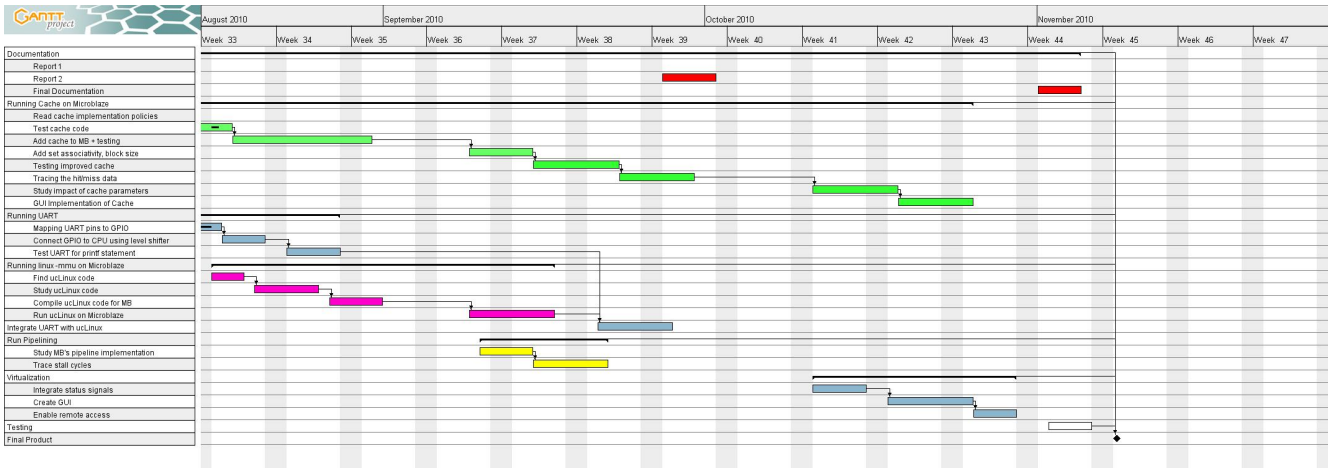
Tasks List

Name	Begin date	End date	Resources
Documentation	8/10/10	11/5/10	Anuj Chauhan
Report 1	8/10/10	8/14/10	
Report 2	9/27/10	10/2/10	
Final Documentation	11/1/10	11/5/10	
Running Cache on Microblaze	8/11/10	10/26/10	Sandeep Kumar Bindal
Read cache implementation policies	8/11/10	8/13/10	
Test cache code	8/13/10	8/18/10	
Add cache to MB + testing	8/18/10	8/31/10	
Add set associativity, block size	9/9/10	9/15/10	
Testing improved cache	9/15/10	9/23/10	
Tracing the hit/miss data	9/23/10	9/30/10	
Study impact of cache parameters	10/11/10	10/19/10	
GUI Implementation of Cache	10/19/10	10/26/10	Anuj Chauhan Sandeep Kumar Bindal
Running UART	8/12/10	8/28/10	Ankit Kumar Jain Anuj Chauhan Yogesh Kumar
Mapping UART pins to GPIO	8/12/10	8/17/10	
Connect GPIO to CPU using level shifter	8/17/10	8/21/10	
Test UART for printf statement	8/23/10	8/28/10	
Running linux -mmu on Microblaze	8/16/10	9/17/10	Ankit Kumar Jain Tarundeep Singh
Find ucLinux code	8/16/10	8/19/10	
Study ucLinux code	8/20/10	8/26/10	
Compile ucLinux code for MB	8/27/10	9/1/10	
Run ucLinux on Microblaze	9/9/10	9/17/10	
Integrate UART with ucLinux	9/21/10	9/28/10	
Run Pipelining	9/10/10	9/22/10	
Study MB's pipeline implementation	9/10/10	9/15/10	
Trace stall cycles	9/15/10	9/22/10	
Virtualization	10/11/10	10/30/10	
Integrate status signals	10/11/10	10/16/10	
Create GUI	10/18/10	10/26/10	
Enable remote access	10/26/10	10/30/10	
Testing	11/2/10	11/6/10	
Final Product	11/8/10	11/9/10	

Resources List

Name	Default role
Tarundeep Singh	developer
Yogesh Kumar	developer
Anuj Chauhan	developer
Ankit Kumar Jain	developer
Sandeep Kumar Bindal	developer

Gantt Chart


August 2010
September 2010
October 2010
November 2010
Week 33
Week 34
Week 35
Week 36
Week 37
Week 38
Week 39
Week 40
Week 41
Week 42
Week 43
Week 44
Week 45
Week 46
Week 47
Documentation
Report 1
Report 2
Final Documentation
Running Cache on Microblaze
Read cache implementation policies
Test cache code
Add cache to MB + testing
Add set associativity, block size
Testing improved cache
Tracing the hit/miss data
Study impact of cache parameters
GUI implementation of Cache
Running UART
Mapping UART pins to GPIO
Connect GPIO to CPU using level shifter
Test UART for printf statement
Running linux -mmu on Microblaze
Find ucLinux code
Study ucLinux code
Compile ucLinux code for MB
Run ucLinux on Microblaze
Integrate UART with ucLinux
Run Pipelining
Study MB's pipeline implementation
Trace stall cycles
Virtualization
Integrate status signals
Create GUI
Enable remote access
Testing
Final Product

Resources Chart