CSP315

Virtual FPGA

Sandeep Kr Bindal
Yogesh Kumar
Ankit Kr Jain
Tarundeep Singh
Anuj Chauhan

Motivation

- IIT-Delhi is planning a **virtual lab** that can benefit thousands of students across India learn experiments in engineering and technology.

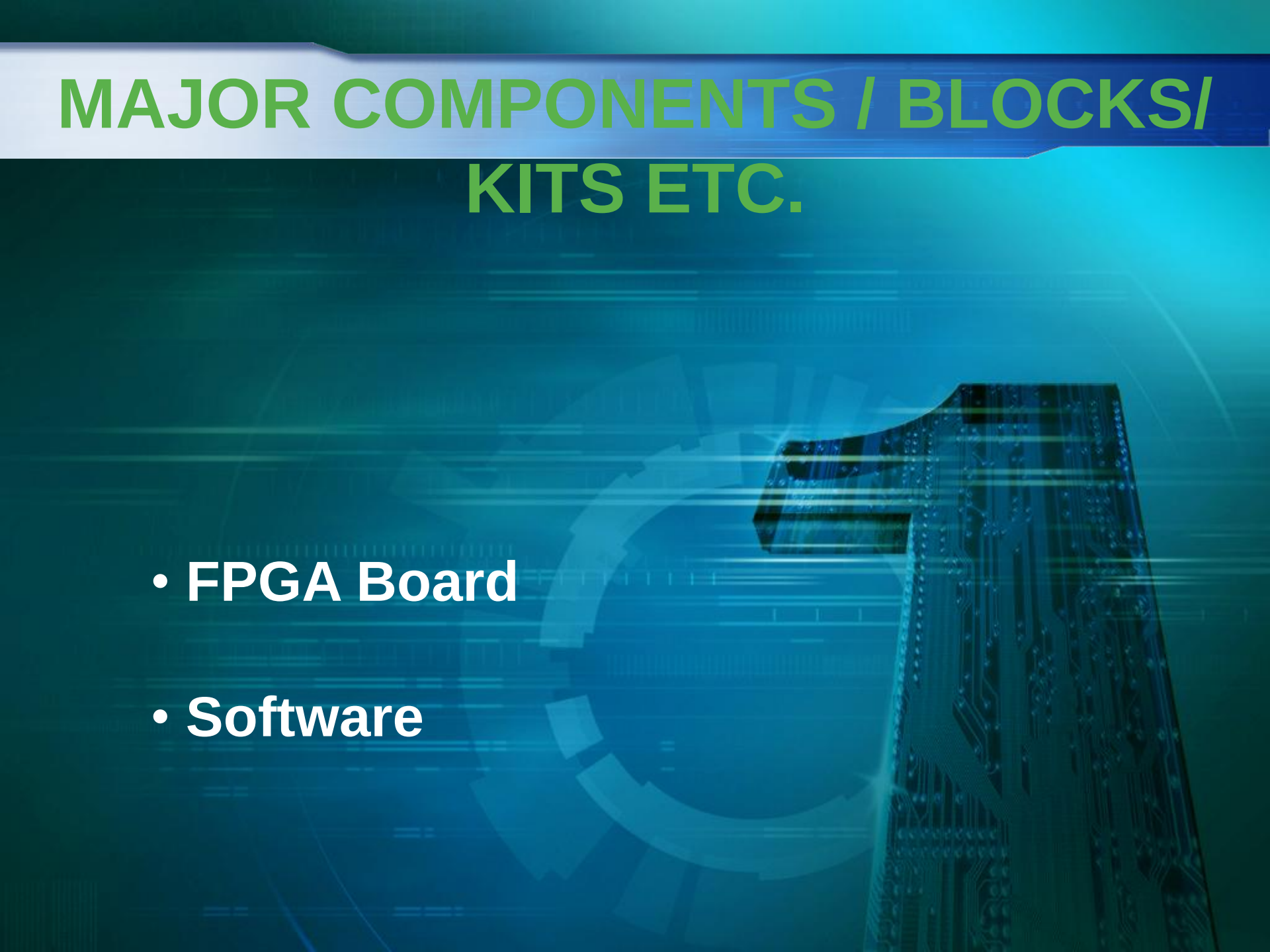
"The virtual laboratory will bridge physical distance and availability of resources in far off places. Today it is possible to design good experiments among students for better learning"

- Surendra Prasad
Director, IIT Delhi

Motivation

- As part of the FPGA and Digital Design Lab we aim to build :
  - experiments that will help students understand computer architecture related concepts such as caching, pipelining etc.

MAJOR COMPONENTS / BLOCKS/ KITS ETC.



- FPGA Board
- Software

Board – Alpha Data ADM-XRC-II

- Based on the Xilinx® Virtex-2 FPGAs
- Up to 24MBytes of SSRAM
- Has interface to allow *specialised IO modules*
- *Complete processing system*, *memory*, *I/O and Host interface* resources.
- Contains a *PCI to local bus bridge*

Status

- Already installed on two machines in the FPGA Lab.
- Machines can be accessed using SSH from our laptops 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.

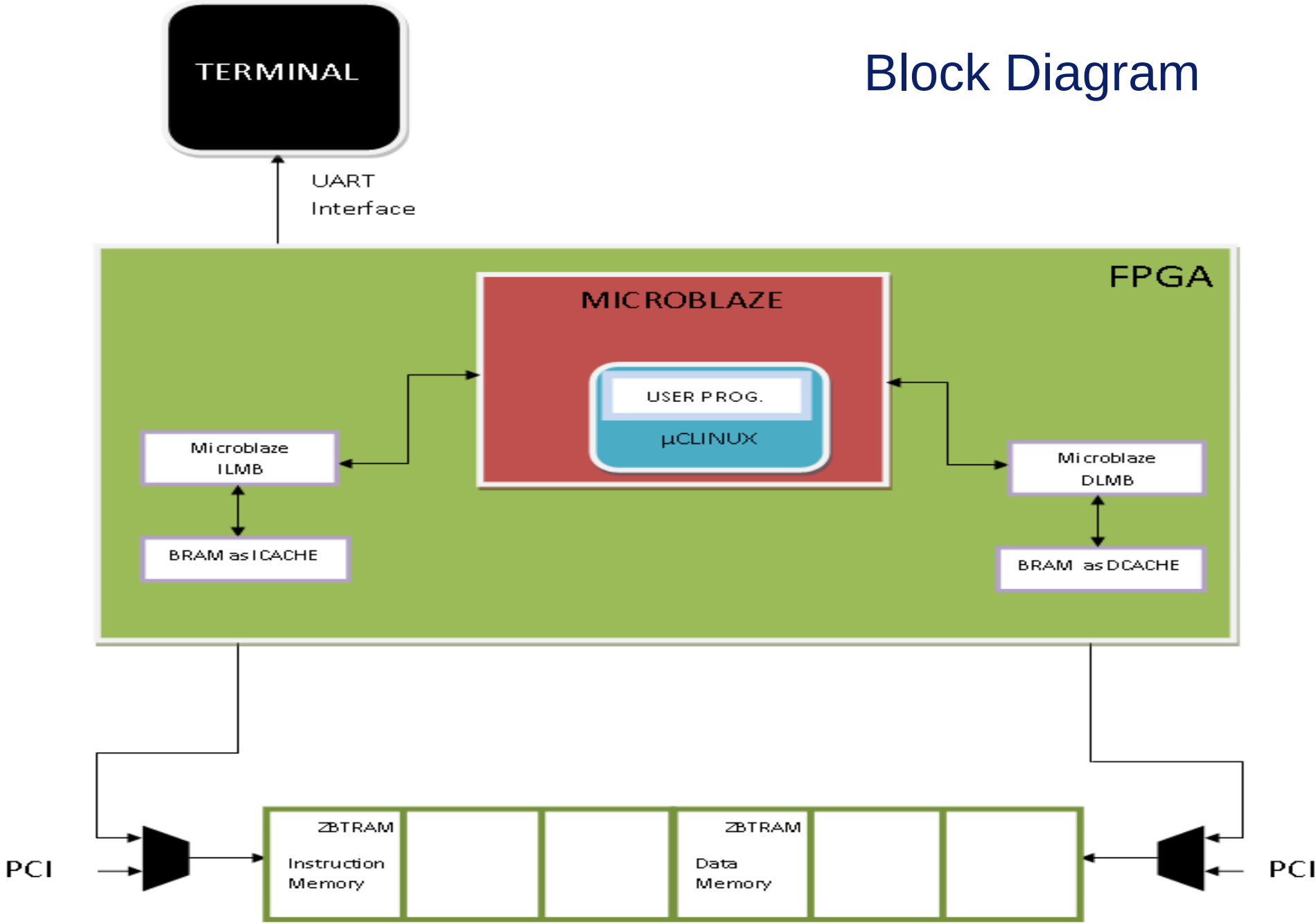
Work Done Previously

- We were able to access various memory modules like BRAM(on chip),ZBT-RAM(off chip)
- Ran Microblaze using BRAM as memory(dual ported) and tested it (executed instructions in machine code)
- Used 2 separate BRAMs for Instruction and Data memory.

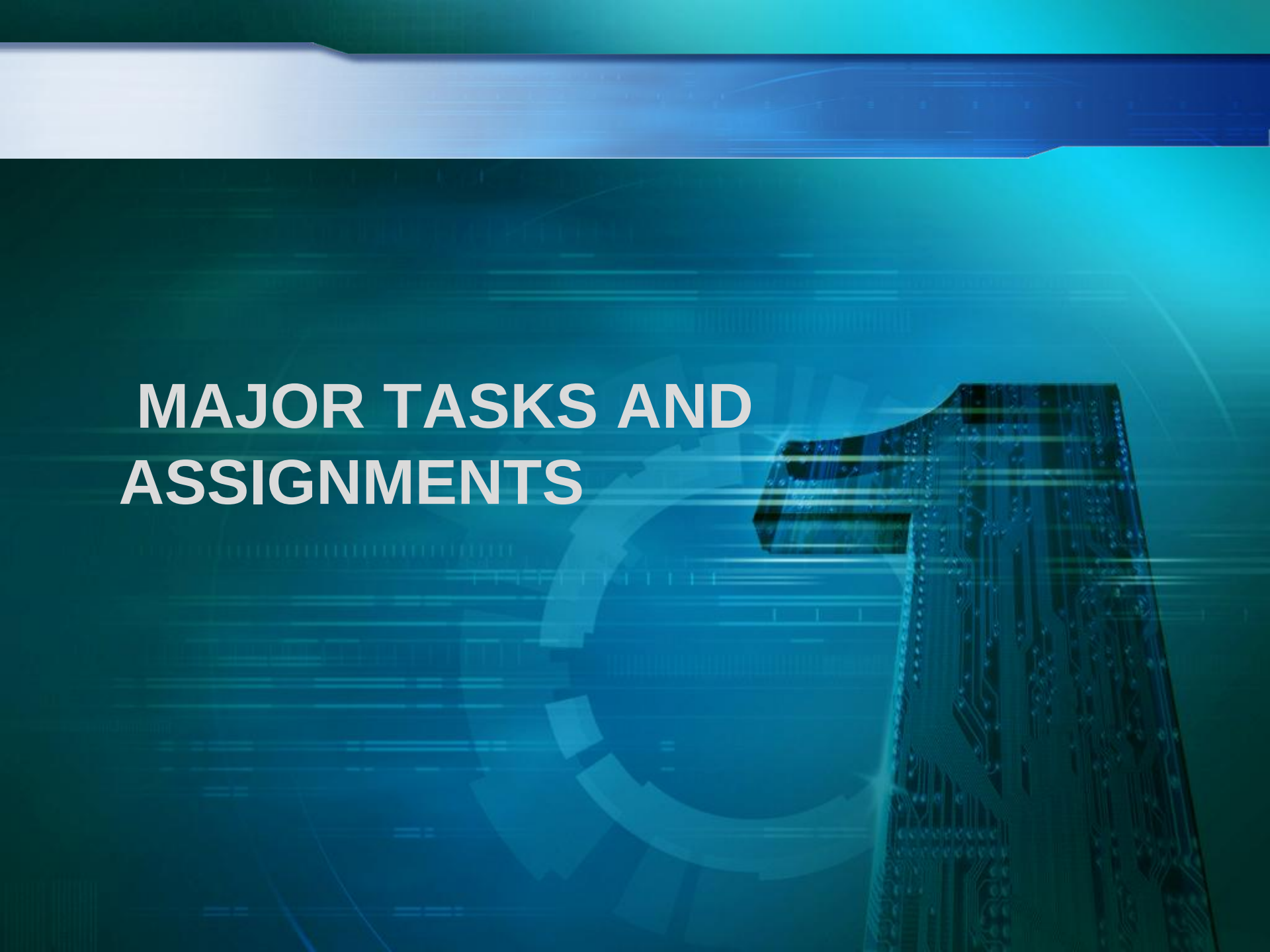
Work Done Previously

- Tested Interrupt (microblaze to PC) and polling.
- Automated execution of simple C programs with help of mb-gcc (cross compiler for Microblaze).

Block Diagram



MAJOR TASKS AND ASSIGNMENTS



Running Cache (Sandeep, Yogesh)

- Read cache implementation policies (write back/through)
- Adding cache to Microblaze and testing
- Adding functionalities such as set associativity, block size
- Tracing the hit/miss data for the cache
- Study the impact of varying cache parameters on performance

Running UART (Yogesh , Sandeep)

- 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)

Monitor/OS (Ankit , Tarundeep)

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

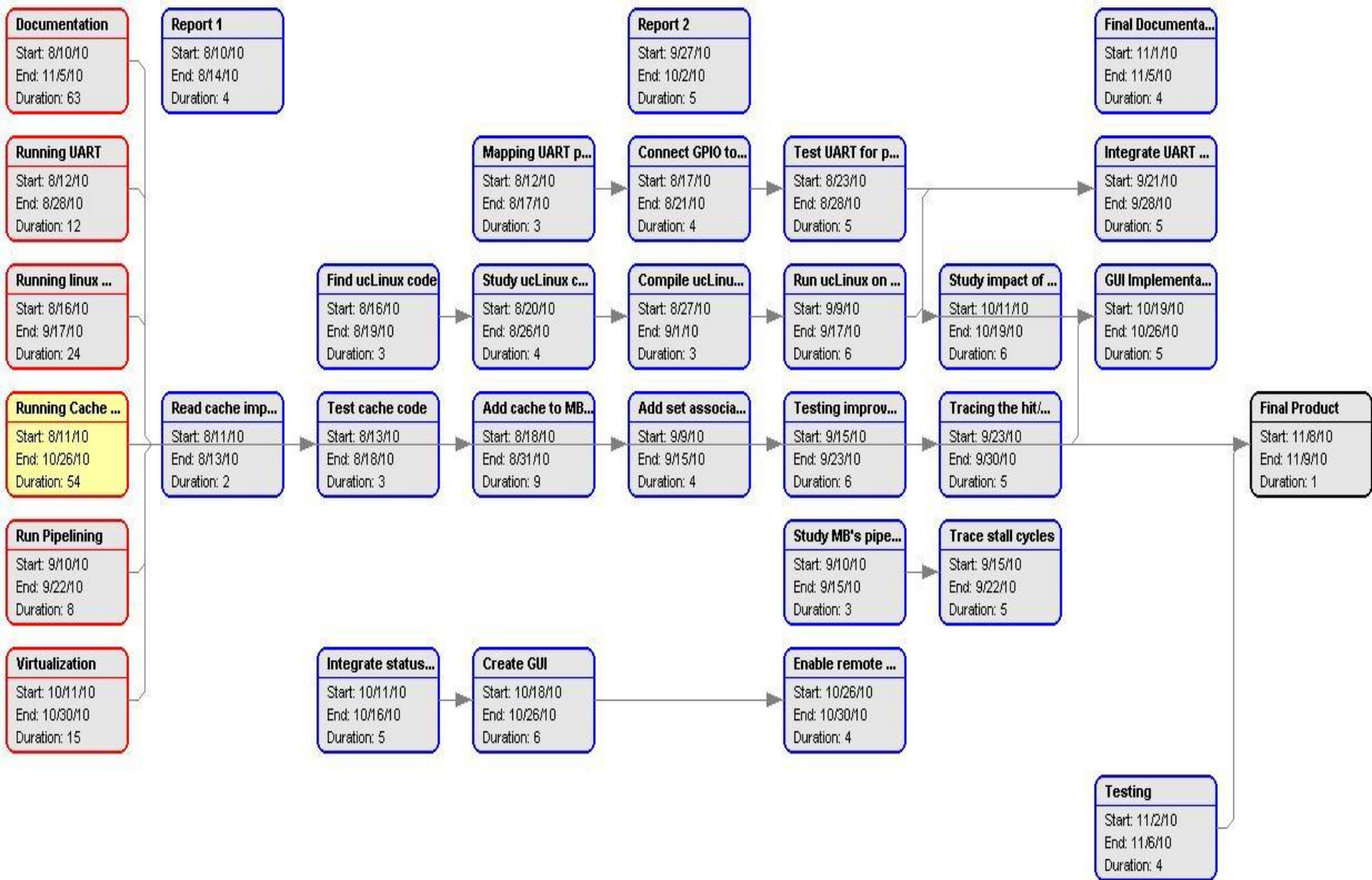
Pipelining (Anuj , Ankit)

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

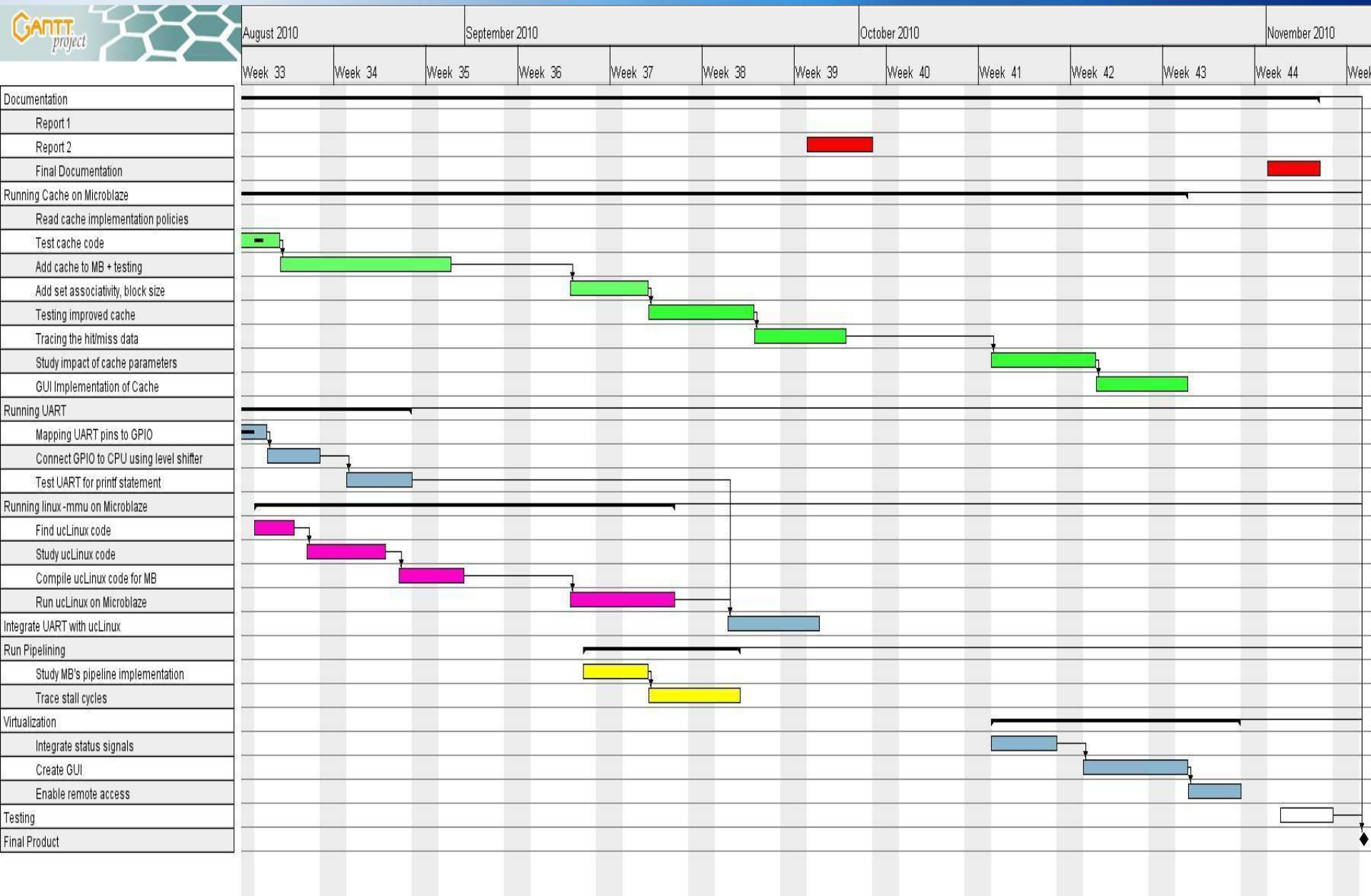
Virtualization (Anuj , Tarundeep)

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

PERT CHART


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

GANTT CHART

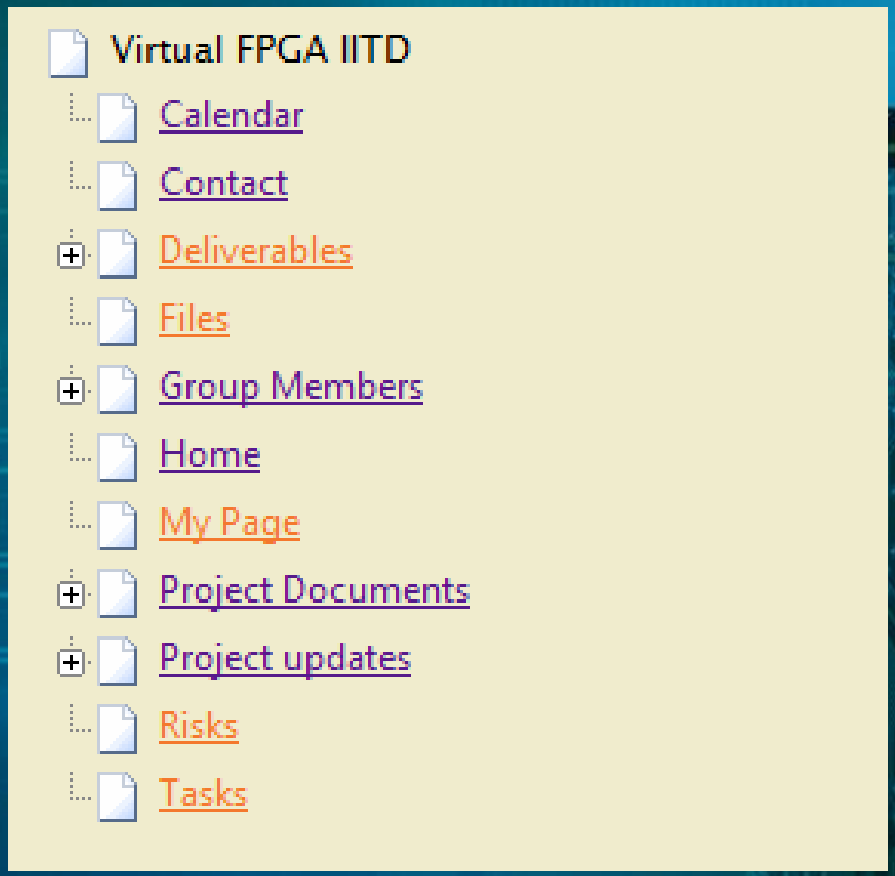

August 2010
September 2010
October 2010
November 2010
Week 33
Week 34
Week 35
Week 36
Week 37
Week 38
Week 39
Week 40
Week 41
Week 42
Week 43
Week 44
Documentation
Report 1
Report 2
Final Documentation
Running Cache on Microblaze
Read cache implementation policies
Test cache code
Add cache to MB + testing
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

POSSSIBLE EXTENSIONS

- Multiple Microblazes
- FPGA Scheduling so that multiple users can access the FPGA chip at once.

Project Weblink

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



Thank You !