**DESIGN OF RISC PROCESSOR USING MULTI-CYCLE DATAPATH**

 **SAI PAVAN YARAGUTI (****szy0014@auburn.edu****)**

 **SWATHI BATTHULA (****szb0047@auburn.edu****)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

 We designed a RISC instruction set processor able to execute 16 bit instructions using “Multi-Cycle Datapath”. At first a VHDL model is designed which is later simulated and verified in Mentor Graphics’ Model Sim. This is then implemented on Alteras DE2 FPGA board using Quartus II software. From this project, we learnt the advanced programming methodologies of the VHDL language and also could do indepth analysis of the functioning of a digital system (by implementing it on FPGA) and also some basic digital entities like the ALU, FSM, memory etc. We implemented a memory swap program (using only two registers and storing them in memory) by which we could clearly analyze the datapath for basic instructions on the hardware.

Our instruction set consists of all the basic instructions required to execute a simple program. During the initial execution, we had problems writing back into the register file as the data arrival and the Register Write enable signals are going high at the same time, leaving room for the trouble of data not being written into the Register file. We corrected this problem by making the Register Write enable signal active for the negative clock edge. As future work, we would like to reduce the delays between each module by using better VHDL models so as to increase the frequency of operation of the processor. Or we would instead implement the processor using Pipeline Datapath which reduces the average time of execution of an instruction and thus speeding up the processor.

We can also analyze the processor performance and try to design a more efficient processor by reducing the delay (thus increasing the speed), reducing the overall hardware etc.

For anyone who is taking up this project in future we would recommend to start on the project earlier as there might by lot of problems during the final hardware implementation part. Once the processor is working, we would recommend them to analyze the processor performance (instead of leaving it there or going to a new datapath) and try to make the processor more efficient (by decreasing the hardware, decreasing the delay etc.). It is always recommended to gradually increase the complexity of the test program and then to analyze the average execution time for each instruction of each program. This would give more accurate performance ratings for a processor. We would recommend them to concentrate more on the jump and branch type instructions as we found them to be most difficult to implement on a FPGA board.