**Implementation of 4 bit ALU using NASIC based system**

This project will explore design of 4 bit ALU design. The ALU will support the following operations –

Arithematic functions -

ADD, Subtract

Logical functions -

AND, OR, NAND, NOR

Extension (if time permits) -

XOR?

Branch instruction?

A simple flipflop/memory element?

A simple instruction set will be developed for these operations

**Performance evaluation –**

Performance comparison will be made with CMOS technology for these operations. The metrics will be power/speed analysis

HSPICE based simulations will be used to do the performance/power evaluation

**Preliminary work –**

Till now we have done the following –

Studied and derived CMOS circuits for the following - NOT/AND/NAND/OR/NOR/passthrough/Multiplexer/Adder/Subtractor/ALU

We have attached our preliminary circuit diagrams as part of this report

We have not yet generated SPICE netlists for these models and run simulations yet but by end of week 1 we plan on finishing this part

Studied NASIC based models/read papers to get started

Are in process of putting together simulation models for –

NOT gate (already provided as prototype)

NAND gate (as provided in the book chapter)

FLIPFLOP (as outlined in the book chapter)

Once we get basic simulation of these gates, we will focus on getting other circuits needed for ALU design

*Long shot -* A critical component we want to add is a support for GOTO instruction. If we can get that to work, then we can do some basic benchmarks like loops. We will keep this as a long shot for the project.

**Power/performance analysis –**

A large part of this will be done by evaluating the CMOS ALU design with the NASIC based ALU design. Functional evaluation will be based upon HSPICE simulations

Power/performance comparisons will also be performed

*Long shot* – if we can have a small memory going, then ideally we will like to study Compiler impact on NASIC based CPU – One of the things we will like to study is how we can rearrange a benchmark and impact it would have on program execution. Assuming that we have a sequence of independent instructions, we want to investigate if order of operations running on the ALU have any change in the power metrics.