# INTRODUCTION

In this project we have suppose to design hardware accelerator for natural language processing for that we implemented a hardware for BERT model. Particularly we implemented non-linear vector unit part of the complete architecture through Verilog code. Mainly we can divide the architecture in three parts namely control unit, memory unit and compute unit.

# Design steps

Control unit

Control unit generates all the control flags and addresses need to fetch and operate on the values stored at the respective locations in memory unit. It contains instruction buffer(IB) which takes micro instructions from the user. Micro instruction is a 13 bit out of which 10 bits are read and write addresses and 3 are opcode. This three bit opcode later will get decoded into 8 bits further. We designed an instruction register of 18 bits. These 18 bits comprises of 6 components. 6 Components are as below:

SV[1bit]: It differentiates scalar and vector operations.

R\_am[2bit]:It determines read addressing mode.

W\_am[2bit]: It determines write addressing mode.

RA[5bit]: It determines Read Address.

WA[5bit]: It determines write address.

Op[3bit]: It determines opcode.

Entire flow works in two cycles as below:

1. Fetch cycle: In this cycle we fetch the instruction into instruction register.
2. Decode cycle: from the instruction we fetched previously, we decode the instruction into flags.

Architecture of control unit works as follows:

1)In Fetch cycle, User can provide up to 9 instructions into instruction buffer which is a queue.

2)From the instruction buffer, instruction will then flow into micro control unit where it gets divided into 10+3 segments.

3) 10 bits will be fed to shift register.

4) 3 bits will be sent to address register

5) These 3 bits will be fed to micro instruction memory as address, and we fetch data (8bits) located at the address into data register.

6) From the data register, we send the 8 bit data to Instruction register where it combines with 10 bits which we separated in step 2 and form 18 bit complete instruction.

In decode cycle,

We segment these 18 bits as per their significance explained above and decoded into flags.

Opcode decides the operation which compute unit performs. SV flag defines whether it should perform scalar or vector function. Read addressing mode and write addressing mode defines whether it is I/O addressing mode, direct and indirect addressing modes.

Memory unit

Memory unit further divided in three units first part is main memory which has 32 locations of 64 bits each (to store complete vector of size 4 each element of 16 bit).5 bit address is used.

Second part is load store unit which loads the data from the memory to the compute unit and also store data back to the memory after computation.

Third part is variable register file which has 32 registers of 64 bits each (to store complete vector of size 4 each element of 16 bit).

Compute unit will get inputs from memory unit and store back its result in memory.

Compute unit

It contains two major parts one part contains all the vector operations including addition, subtraction, multiplication, dot product, comparator, magnitude calculator.

Other part is piece-wise linearization module which can convert any non-linear function into its piece-wise linear model. Initially we have given boundaries for functions like square root, SoftMax, Gelu but later any other non linear function can also be implemented in this just by giving proper boundaries.