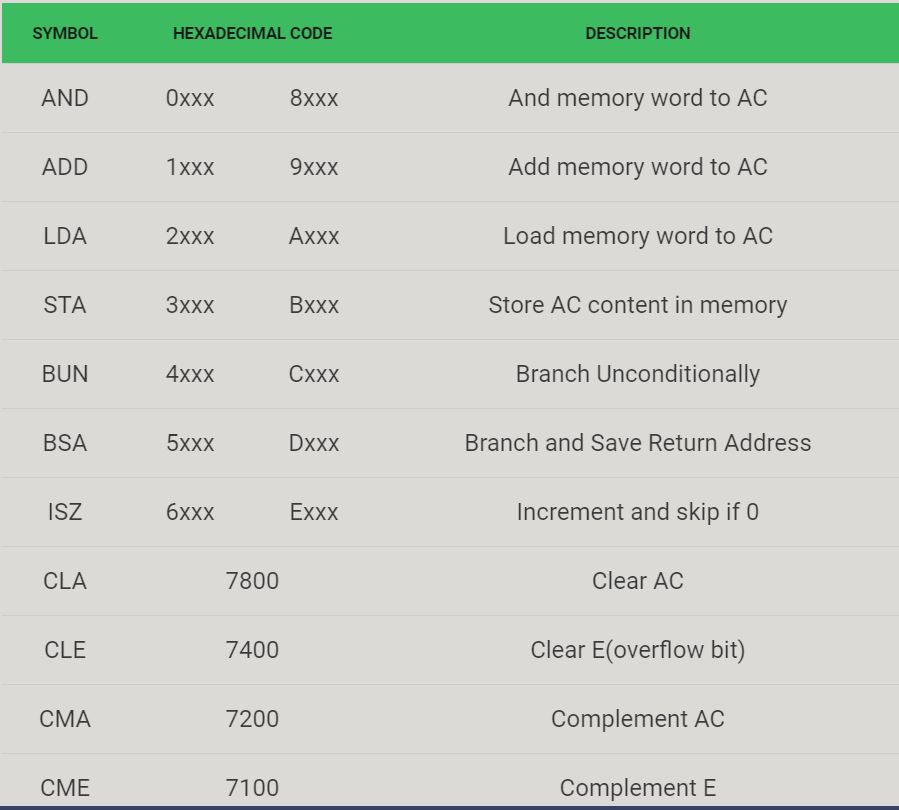
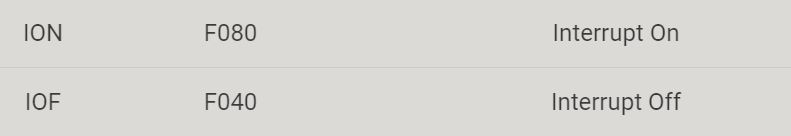
***Ripunjay Narula***

**19BCE0470**

**Instruction Set**

****

****

****

**Instructions (ISA) of chosen processor: Google TPU**

It chose the Complex Instruction Set Computer (CISC) style as the basis of the TPU instruction set instead. A CISC design focuses on implementing high-level instructions that run more complex tasks (such as calculating multiply-and-add many times) with each instruction. Let's take a look at the block diagram of the TPU.

The TPU includes the following computational resources:

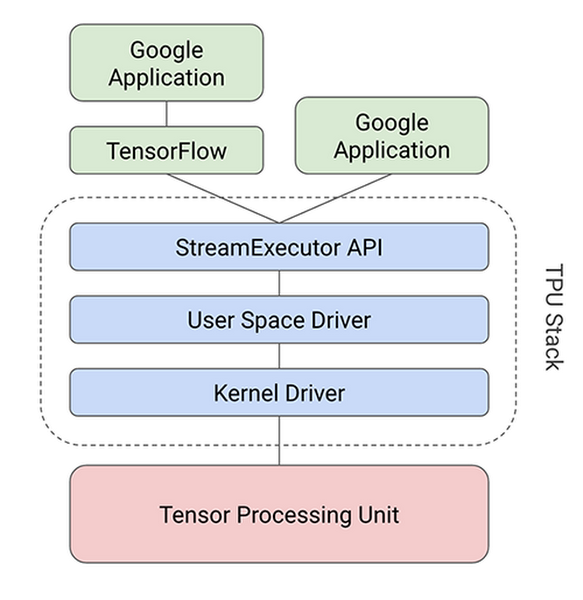
* Matrix Multiplier Unit (MXU): 65,536 8-bit multiply-and-add units for matrix operations
* Unified Buffer (UB): 24MB of SRAM that work as registers
* Activation Unit (AU): Hardwired activation functions

This instruction set focuses on the major mathematical operations required for neural network inference that we mentioned earlier: execute a matrix multiply between input data and weights and apply an activation function.

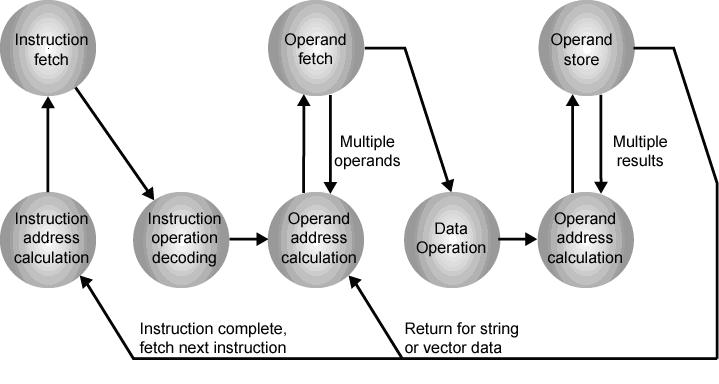
Norm says:

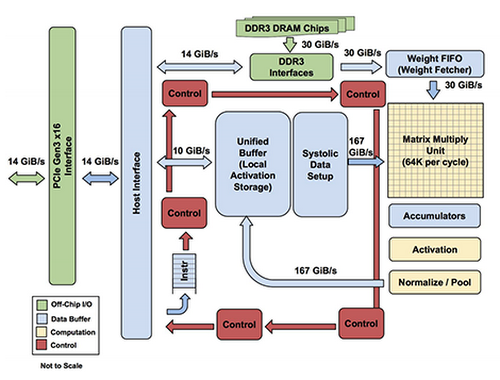
“Neural network models consist of matrix multiplies of various sizes — that’s what forms a fully connected layer, or in a CNN, it tends to be smaller matrix multiplies. This architecture is about doing those things — when you’ve accumulated all the partial sums and are outputting from the accumulators, everything goes through this activation pipeline. The non-linearity is what makes it a neural network even if it’s mostly linear algebra.”(from First in-depth look at Google's TPU architecture, the Next Platform)”

In short, the TPU design encapsulates the essence of neural network calculation, and can be programmed for a wide variety of neural network models. To program it, we created a compiler and software stack that translates API calls from TensorFlow graphs into TPU instructions.



**Instruction Cycle State Diagram**

****



* Matrix Multiplier Unit (MXU): 65,536 8-bit multiply-and-add units for matrix operations
* Unified Buffer (UB): 24MB of SRAM that work as registers
* Activation Unit (AU): Hardwired activation functions

Some high Level Instructions

To control how the MXU, UB and AU proceed with operations, we defined a dozen high-level instructions specifically designed for neural network inference. Five of these operations are highlighted below.

|  |  |
| --- | --- |
| TPU Instruction | Function |
| Read\_Host\_Memory | Read data from memory |
| Read\_Weights | Read weights from memory |
| MatrixMultiply/Convolve | Multiply or convolve with the data and weights,accumulate the results |
| Activate | Apply activation functions |
| Write\_Host\_Memory | Write result to memory |