Ripunjay Narula
19BCE0470

Instruction Set

SYMBOL	HEXADECIMAL CODE		DESCRIPTION
AND	0xxx	8xxx	And memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store AC content in memory
BUN	4xxx	Cxxx	Branch Unconditionally
BSA	5xxx	Dxxx	Branch and Save Return Address
ISZ	6xxx	Exxx	Increment and skip if 0
CLA	7800		Clear AC
CLE	7400		Clear E(overflow bit)
CMA	7200		Complement AC
CME	7100		Complement E

CIR	7080	Circulate right AC and E
CIL	7040	Circulate left AC and E
INC	7020	Increment AC
SPA	7010	Skip next instruction if AC > 0
SNA	7008	Skip next instruction if AC < 0
SZA	7004	Skip next instruction if AC = 0
SZE	7002	Skip next instruction if E = 0
HLT	7001	Halt computer
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip on input flag
SKO	F100	Skip on output flag
ION	F080	Interrupt On
IOF	F040	Interrupt Off

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

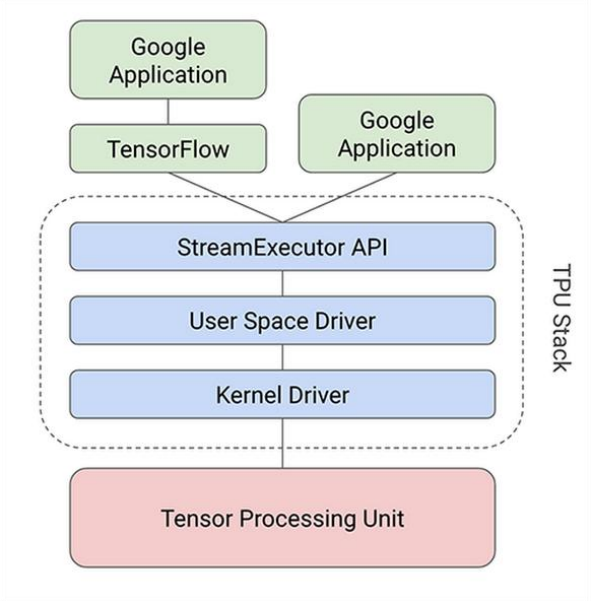
- Unified Buffer (UB): 24MB of SRAM that work as registers
- Activation Unit (AU): Hardwired activation functions

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.

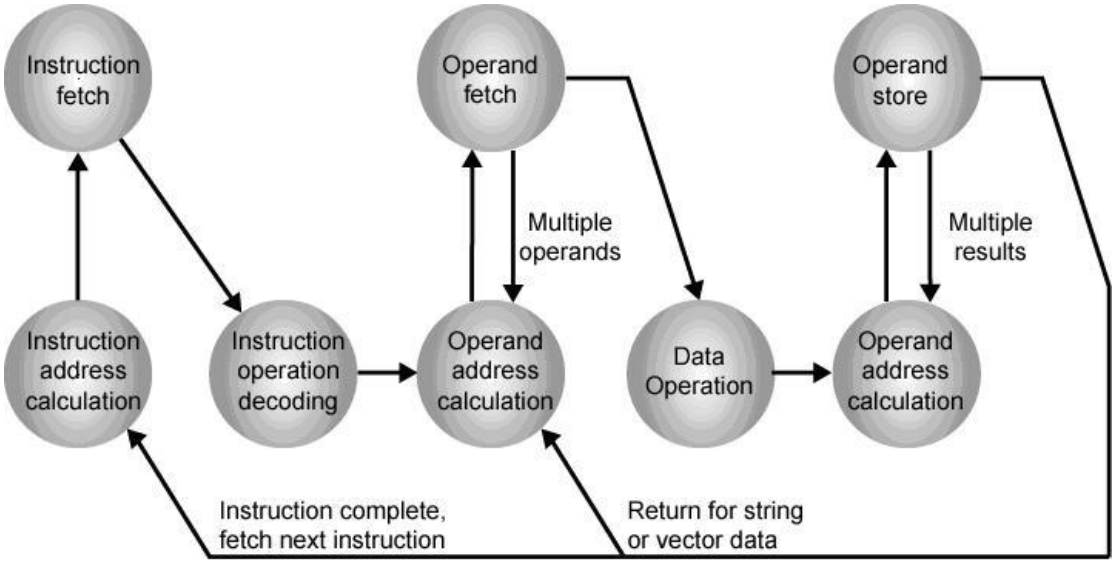
Norm says:

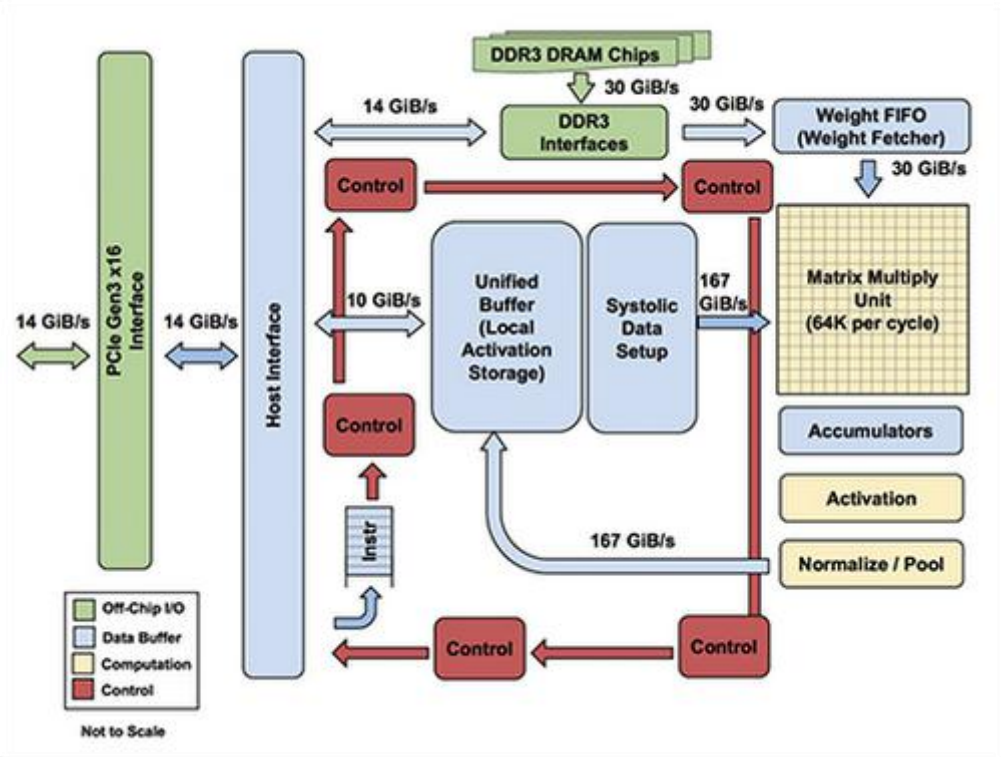
“Neural network models consist of matrix multiplies of various sizes — that’s what forms a fully connected layer, or in a CNN, it tends to be smaller matrix multiplies. This architecture is about doing those things — when you’ve accumulated all the partial sums and are outputting from the accumulators, 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