**A SEMINAR REPORT**

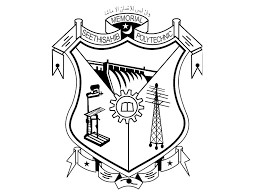
**TENSOR PROCESSING UNIT**

***Submitted by***

**MOHAMMED ASLAM EP**

***in partial fulfillment for the award of the diploma in***

## COMPUTER ENGINEERING



**DEPARTMENT OF COMPUTER ENGINEERING**

**SEETHI SAHIB MEMORIAL POLYTECHNIC COLLEGE, TIRUR**

## MARCH 2017-2018

**SEETHI SAHIB MEMORIAL POLYTECHNIC**

**COLLEGE, TIRUR**

## BONAFIDE CERTIFICATE

Certified that this report titled “**TENSOR PROCESSING UNIT (TPU)**” is the bona fide work of **Mr. MOHAMMED ASLAM E.P (REG NO: 15130399)** who carried out the seminar under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

### Mr. K N SALEEM Mr. RAFI P

Head of the Department, Lecturer in Computer Engineering,

Department of Computer Engineering, Department of Computer Engineering,

S S M Polytechnic College, S S M Polytechnic College, Tirur, Kerala 676105. Tirur, Kerala 676105.

### INTERNAL EXAMINER EXTERNAL EXAMINER

**Place: Tirur Date:**

# ABSTRACT

Although Google considered building an Application-Specific Integrated Circuit (ASIC) for neural networks as early as 2006, the situation became urgent in 2013.

That’s when google realized that the fast-growing computational demands of neural networks could require to double the number of data centers google operate.

Google announced its next generation of its custom Tensor Processing Units (TPUs) machine learning chips at Google I/O 2017. These chips, which are designed specifically to speed up machine learning tasks, are supposed to be more capable than CPUs or even GPUs at these tasks and are an upgrade from the first generation of chips the company released at 2016’s I/O.

And speed up they have. Google claims the each second-generation TPU can deliver up to 180 teraflops of performance. We will have to wait and see what the average benchmarks look like, but they are a step forward for more than speed. The first generation TPU was only able to handle inference. The new one can also be used for training machine learning models, a significant part of the machine learning workflow all within this single, powerful chip.

That means that you can build a machine learning model — for example, to correctly identify an object in a photo is a tree, a car or a cat. Inference in machine learning refers to the statistical likelihood that the machine’s conclusions are correct for example, based on the model, you may be 85 percent confident that this is actually a tree and not a artificial tree.

## ACKNOWLEDGEMENT

I hereby acknowledge all for helping me during my working .First I thank god, and dedicate this seminar for my dear mother, father, sister, and all my dear friends. Without whom it would never have been accomplished.

I would like to place on record my deep sense of gratitude to **Mr. SALEEM K.N**. Head of department, for his generous guidance, help and useful suggestions.

I express my sincere gratitude to **Mr. RAFI.P.** Lecturer of department, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

I also wish to extend my thanks to **Mrs. HASEENA.M,** and other colleagues for checking my seminar and for their insightful comments and constructive suggestions to improve this seminar.

I am also extremely grateful towards my teachers, staffs, and my colleagues of Department of Computer Engineering, SSM Polytechnic College, Tirur for providing me the facilities and also giving support for doing this seminar.

Finally I thank Department of Computer Engineering, SSM Polytechnic college Tirur for giving me this opportunity and providing an environment to do seminar.

**MOHAMMED ASLAM E.P**

## TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

[**ABSTRACT** **iii**](#_Toc33176)

[**LIST OF TABLES** **viii**](#_Toc33177)

[**LIST OF FIGURES** **ix**](#_Toc33178)

**LIST OF ABBREVIATIONS x**

1. **INTRODUCTION**  **01**
2. **HISTORY 02**
3. **WHAT IS ASIC? 03**
4. **INTRODUCTION TO MACHINE LEARNING 04**
5. **INTRODUCTION TO NEURAL NETWORK 05**
   1. What is NN? 05
   2. Prediction with Neural Networks 05
6. **ARCHITECTUR OF TPU 08**
   1. PCI-E Gen3x16 Interface 09
   2. Host Interface 09
   3. Instruction Buffer Register 09
   4. DDR3 DRAM Chip and Its Interface 09
   5. Weight FIFO 10
   6. Metrix Multiply Unit 10
   7. Accumulators 10
   8. Activation 10
   9. Unified Buffer Storage 11

### 7 WORKING OF TPU 12 8 IMPLEMENTATION OF TPU 13 9 INSTRUCTIONS IN TPU 14 10 PARALLEL POEOCESSING ON MXU 15

10.1 How Systolic Execution Is Possible? 15

10.2 Data Flow of The MXU 18

### 11 TENSORFLOW TO TPU 19

11.1 What Is Tensorflow? 19

11.2 Tensorflow Graphs into TPU Instructions 19

### 12 GENERATIONS OF TPU 22

12.1 First Generation TPU 22

12.2 Second Generation TPU 22

### 13 APPLICATION OF TPU 23

13.1 Voice/Sound Recognition 23

13.2 Text Based Applications 23

13.3 Image Recognition 23

13.4 Time Series 24

13.5 Video Detection 24

### 14 ADVANTAGE OF TPU 25

14.1 Number of Operation Per Cycle 25

14.2 Minimal and Deterministic Design 26

14.3 Make Prediction 27

**CONCLUSION 29**

**REFERENCE 30**

viii

# LIST OF TABLES

**TABLE NO. TITLE PAGE NO.**

5.1 Six Application of NN 7

14.1 Number of Operation Per Cycle 25

ix

# LIST OF FIGURES

**FIGURE NO. TITLE PAGE NO.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5.1 |  |  | [Neural Network](https://en.wikipedia.org/wiki/Convolutional_neural_network)Matrix Multiplication |  | 06 |
| 6.1 |  |  | TPU Block Diagram |  | 08 |
| 6.2 |  |  | Floor Plan of TPU Die |  | 11 |
| 8.1 |  |  | TPU Printed Circuit Board |  | 13 |
| 10.1 |  |  | Systolic Array Chains |  | 16 |
| 10.2 |  |  | Data Flow of the Matrix Multiply Unit |  | 18 |
| 11.1 |  |  | The Software Stack of TPU |  | 20 |

14.1 Performance/Watt Relative to Contemporary

CPU’s and GPU’s 26

14.2 Throughput Under 7 ms Latency Limit 27

14.3 CPU, GPU and TPU Performance On Six

Reference Workloads 28

x

## LIST OF ABBREVATIONS

ASIC - [Application-Specific Integrated Circuit](https://en.wikipedia.org/wiki/Application-specific_integrated_circuit)

TPU - Tensor Processing Unit

GPU - [Graphics Processing Unit](https://en.wikipedia.org/wiki/Graphics_processing_unit)

CPU - Central Processing Unit

ML - Machine Learning

NN - Neural Network

RISC -  [Reduced Instruction Set Computer](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing)

CISC - [Complex Instruction Set Computer](https://en.wikipedia.org/wiki/Complex_instruction_set_computing)

MXU - Matrix Multiplier Unit

UB - Unified Buffer

AU - Activation Unit

PCIe - Peripheral Component Interconnect Express

MiB - Mebibyte

GiB - Gebibyte

FIFO - First In First Out

DMA - Direct Memory Access

FLOPS - Floating Point Operations Per Second

IoT - Internet of Things

CRM - Customer Relationship Management

1

**CHAPTER 1**

## INTRODUCTION TO TPU

A **Tensor Processing Unit** (TPU) is an application-specific integrated circuit (ASIC) developed by Google specifically for machin**e** learning. Compared to a [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU) , it is designed explicitly for a higher volume of reduced precision computation (e.g. as little as [8-bit](https://en.wikipedia.org/wiki/8-bit) precision) with higher [IOPS](https://en.wikipedia.org/wiki/IOPS) per [watt,](https://en.wikipedia.org/wiki/Watt) and lacks hardware for rasterisation/[texture mapping.](https://en.wikipedia.org/wiki/Texture_mapping) The chip has been specifically designed for Google's Tensor Flow framework.

Google has stated that its proprietary tensor processing units were used in the [Alpha-Go versus Lee Sedol](https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol) series of man-machine [Go](https://en.wikipedia.org/wiki/Go_(game)) games. Google has also used TPUs for [Google Street View](https://en.wikipedia.org/wiki/Google_Street_View) text processing, and was able to find all the text in the Street View database in less than five days. In [Google Photos,](https://en.wikipedia.org/wiki/Google_Photos) an individual TPU can process over 100 million photos a day. It is also used in [RankBrain](https://en.wikipedia.org/wiki/RankBrain) which Google uses to provide search results.

**CHAPTER 2**

## HISTORY

Although Google considered building an [Application-Specific Integrated Circuit](https://en.wikipedia.org/wiki/Application-specific_integrated_circuit) (ASIC) for neural networks as early as 2006, the situation became urgent in

2013. That’s when google realized that the fast-growing computational demands of neural networks could require doubling the numbers of datacenters google operate.

Usually, ASIC development takes several years. In the case of the TPU, however, google designed, verified, built and deployed the processor to google datacenters in just 15 months. “Norm Jouppi”, the tech lead for the TPU project (also one of the principal architects of the [MIPS processor)](https://en.wikipedia.org/wiki/MIPS_instruction_set) described the sprint this way:

“We did a very fast chip design. It was really quite remarkable. We started shipping the first silicon with no bug fixes or mask changes. Considering we were hiring the team as we were building the chip, then hiring RTL (circuitry design) people and rushing to hire design verification people, it was hectic.”

First generation of TPU was announced in May 2016 at Google I/O 2016. The second generation TPU was announced in May 2017. Google has stated these second generation TPUs will be available on the [Google Compute Engine](https://en.wikipedia.org/wiki/Google_Compute_Engine) for use in Tensor Flow applications.

3

## CHAPTER 3

**WHAT IS ASIC?**

Application specific integrated circuits are highly specialized devices. Unlike other devices, ASICs are non- standard integrated circuits constructed for one specific purpose application only. The work related to ASIC began in the early 1980s itself. Today, we can see that Application Specific Integrated Circuit (ASIC) are changing the way electronic device systems are designed, manufactured, and marketed. So, it is important in this challenging design world to understand the nature, options, design methodologies, and costs of ASIC technology.

Examples of ASIC chip include chips for satellites, chips designed to run a cell phone, Bitcoin Miner, chip used in a voice recorder, Tensor Processing Unit etc.

**CHAPTER 4**

## INTRODUCTION TO MACHINE LEARNING

Machine learning is a method of data analysis that automates analytical model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

Machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It’s a science that’s not new – but one that’s gaining fresh momentum.

While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data – over and over, faster and faster – is a recent development. Here are a few widely publicized examples of machine learning applications you may be familiar with:

* The heavily hyped, self-driving Google car? The essence of machine learning.
* Online recommendation offers such as those from Amazon and Netflix? Machine learning applications for everyday life.
* Knowing what customers are saying about you on Twitter? Machine learning combined with linguistic rule creation.
* Fraud detection? One of the more obvious, important uses in our world today.

**CHAPTER 5**

## INTRODUCTION TO NEURAL NETWORK

**5.1 What is NN?**

In information technology, a neural network (nn) is a system of hardware and/or software patterned after the operation of neurons in the human brain. Neural networks also called artificial neural networks are a variety of [deep learning](http://searchbusinessanalytics.techtarget.com/definition/deep-learning) technologies. Commercial applications of these technologies generally focus on solving complex [signal processing](http://whatis.techtarget.com/definition/digital-signal-processing-DSP) or pattern recognition problems. Examples of significant commercial applications since 2000 include handwriting recognition for check processing, speech-to-text transcription, oil-exploration data analysis, weather prediction and facial recognition.

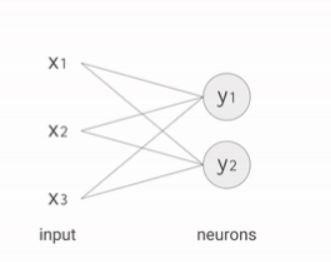
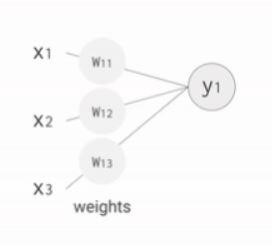
### 5.2 PREDICTION WITH NEURAL NETWORKS

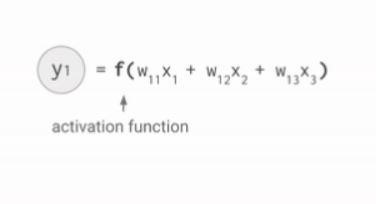
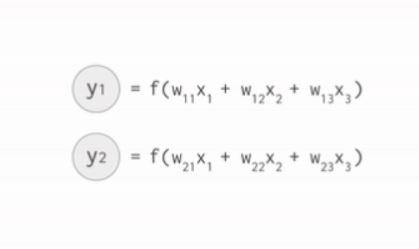
For inference, each neuron in a neural network does the following calculations:

* **Multiply** the input data (x) with weights (w) to represent the signal strength
* **Add** the results to aggregate the neuron’s state into a single value
* Apply an **activation** function (f) (such as [ReLU,](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)) [Sigmoid,](https://en.wikipedia.org/wiki/Sigmoid_function) [tanh](https://en.wikipedia.org/wiki/Hyperbolic_function) or others) to modulate the artificial neuron’s activity.

For example, if you have three inputs and two neurons with a fully connected single-layer neural network, you have to execute six multiplications between the weights and inputs and add up the multiplications in two groups of three. This sequence of multiplications and additions can be written as a **matrix multiplication**.

The outputs of this matrix multiplication are then processed further by an activation function. Even when working with much more complex neural network model architectures, multiplying matrices is often the most computationally intensive part of running a trained model.

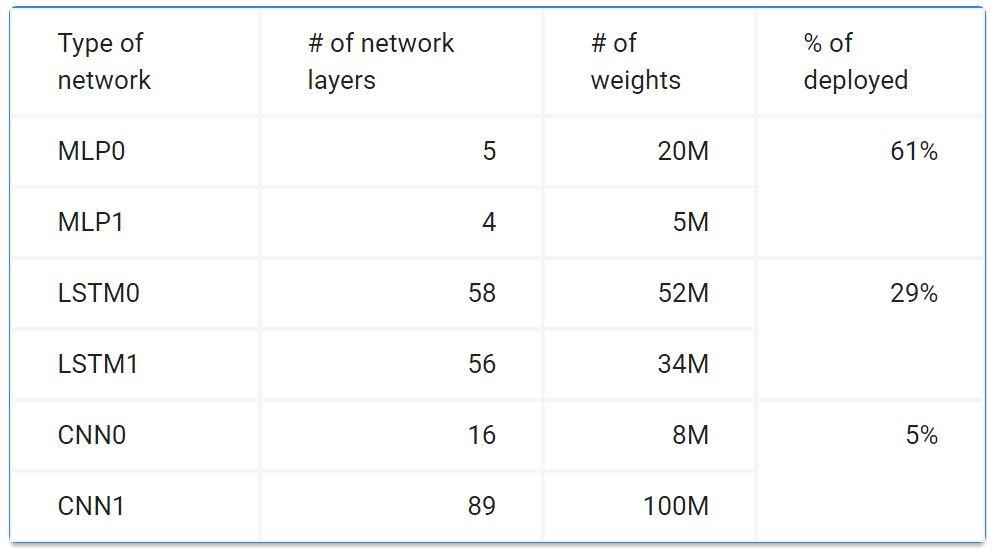
 

**Figure 5.1:** [**Neural Network**](https://en.wikipedia.org/wiki/Convolutional_neural_network) **Matrix Multiplication**

In July 2016 survey six representative neural network applications across

Google’s production services and summed up the total number of weights in each neural network architecture. You can see the results in the table below.

* MLP: [Multilayer Perceptron,](https://en.wikipedia.org/wiki/Multilayer_perceptron)
* LSTM: [Long Short-Term Memory,](https://en.wikipedia.org/wiki/Long_short-term_memory)
* CNN: [Convolutional Neural Network](https://en.wikipedia.org/wiki/Convolutional_neural_network)



**Table 5.1: Six Application of NN**

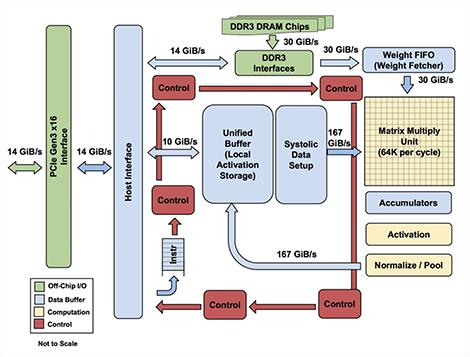
As you can see in the table, the number of weights in each neural network varies from 5 million to 100 million. Every single prediction requires many steps of multiplying processed input data by a weight matrix and applying an activation function.

In total, this is a massive amount of computation. As a first optimization, rather than executing all of these mathematical operations with ordinary 32-bit or 16-bit floating point operations on CPUs or GPUs, we apply a technique called quantization that allows us to work with integer operations instead. This enables us to reduce the total amount of memory and computing resources required to make useful predictions with Google’s neural network models.

**CHAPTER 6**

## ARCHITECTURE OF TPU

Most modern CPUs are heavily influenced by the [Reduced Instruction Set Computer (RISC)](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing) design style. With RISC, the focus is to define simple instructions (e.g., load, store, add and multiply) that are commonly used by the majority of applications and then to execute those instructions as fast as possible. We chose the [Complex Instruction Set Computer (CISC)](https://en.wikipedia.org/wiki/Complex_instruction_set_computing) 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.



**Figure 6.1: TPU Block Diagram**

### 6.1 PCI-E Gen3x16 Interface

**PCI Express** (**Peripheral Component Interconnect Express**), officially abbreviated as **PCIe** or **PCI-e**, is a high-speed [serial](https://en.wikipedia.org/wiki/Serial_communication) [computer](https://en.wikipedia.org/wiki/Computer) [expansion bus](https://en.wikipedia.org/wiki/Expansion_bus) standard, designed to replace the older [PCI,](https://en.wikipedia.org/wiki/Conventional_PCI) [PCI-X,](https://en.wikipedia.org/wiki/PCI-X) and [AGP](https://en.wikipedia.org/wiki/Accelerated_Graphics_Port) bus standards. PCIe has numerous improvements over the older standards, including higher maximum system bus throughput, lower I/O pin count and smaller physical footprint, better performance scaling for bus devices, a more detailed error detection and reporting mechanism (Advanced Error Reporting), and native [hot-plug](https://en.wikipedia.org/wiki/Hot-plug) functionality. More recent revisions of the PCIe standard provide hardware support for [I/O virtualization.](https://en.wikipedia.org/wiki/I/O_virtualization)

### 6.2 Host Interface

A **host** controller **interface** (HCI) is a register-level interface that enables a host controller for USB or IEEE 13 hardware to communicate with a host controller driver in software.

### 6.3 Instruction Buffer Register (IBR)

**IBR (Instruction Buffer Register)** is a temporary register where the opcode of the currently fetched instruction is stored.

### 6.4 DDR3 DRAM Chip and Its Interface

**Double data rate type three DRAM** (**DDR3 DRAM**) is a type of [dynamic random-access memory](https://en.wikipedia.org/wiki/Synchronous_dynamic_random-access_memory) (DRAM) with a high [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)) ("[double data rate"](https://en.wikipedia.org/wiki/Double_data_rate)) interface, and has been in use since 2007. It is the higher-speed successor to [DDR](https://en.wikipedia.org/wiki/DDR_SDRAM) and [DDR2](https://en.wikipedia.org/wiki/DDR2_SDRAM) and predecessor to [DDR4](https://en.wikipedia.org/wiki/DDR4_SDRAM) dynamic rando[m-access memory](https://en.wikipedia.org/wiki/Synchronous_dynamic_random-access_memory) (DRAM) chips. TPU use 32 GB of its own memory, it was able to deliver 34 GB/sec of memory bandwidth on the card.

### 6.5 Weight FIFO

The weights for the matrix unit are staged through an on-chip **Weight FIFO**that reads from an off-chip 8GiB DRAM called Weight Memory(for inference, weights are read-only; 8GiB supports many simultaneously active models). The weight FIFO is four tiles deep.

### 6.6 Metrix Multiply Unit (MXU)

The **Matrix Multiply Unit**is the heart of the TPU. It contains 256x256 MACs that can perform 8-bit multiply-and-adds on signed or unsigned integers.

### 6.7 Accumulators

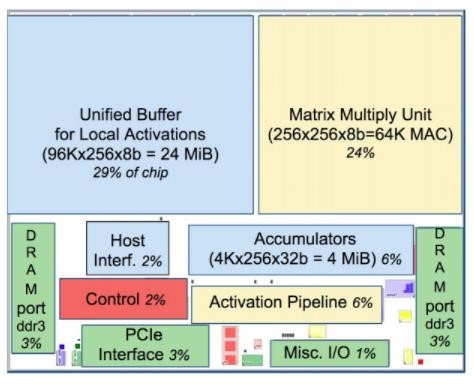
In a central processing unit ([CPU)](https://en.wikipedia.org/wiki/Central_processing_unit), an **accumulator** is a [register](https://en.wikipedia.org/wiki/Processor_register) in which [intermediate](https://en.wiktionary.org/wiki/intermediate) [arithmetic and logic](https://en.wikipedia.org/wiki/Arithmetic_logic_unit) results are stored. In TPU the 16-bit products from MXU are collected in the 4MiB of 32-bit **Accumulators**below the matrix unit.

### 6.8 Activation

It is the unit that activate or apply the function for Matrix Multiplication. By the input given by activator to Metric Multiply unit we can control the computational speed of the Metric Multiplying Unit.

### 6.9 Unified Buffer Storage

**Unified Buffer** can serve as inputs to the Matrix Unit. A programmable DMA controller transfers data to or from TPU Host memory and the Unified Buffer. Figure 2 shows the floor plan of the TPU die. The 24MiB Unified Buffer is almost a third of the die and the Matrix Multiply Unit is a quarter, so the datapath is nearly two-thirds of the die. The 24MiB size was picked in part to match the pitch of the Matrix Unit on the die and, given the short development schedule, in part to simplify the compiler Control is just 2%.



**Figure 6.2: Floor Plan of TPU die**

**CHAPTER 7**

## WORKING OF TPU

The TPU instructions are sent from the host over the PCIe Gen3 x16 bus into an instruction buffer. The internal blocks are typically connected together by 256- *byte*wide paths.

Starting in the upper-right corner, the *Matrix Multiply Unit* is the **heart** of the TPU. It contains 256x256 MACs that can perform 8-bit multiply-and-adds on signed or unsigned integers. The 16-bit products are collected in the 4 MiB of 32-bit *Accumulators* below the matrix unit. The 4 MiB represents 4096, 256-element, 32-bit accumulators. The matrix unit produces one 256-element partial sum per clock cycle. We picked 4096 by first noting that the operations per byte need to reach peak performance is 1350, so we rounded that up to 2048 and then duplicated it so that the compiler could use double buffering while running at peak performance.

When using a mix of 8-bit weights and 16-bit activations (or vice versa), the Matrix Unit computes at half-speed, and it computes at a quarter-speed when both are 16 bits. It reads and writes 256 values per clock cycle and can perform either a matrix multiply or a convolution. The matrix unit holds one 64KiB tile of weights plus one for double-buffering (to hide the 256 cycles it takes to shift a tile in). This unit is designed for dense matrices. Sparse architectural support was omitted for time-todeploy reasons. Sparsity will have high priority in future designs.

The weights for the matrix unit are staged through an on-chip *Weight FIFO* that reads from an off-chip 8GiB DRAM called *Weight Memory* (for inference, weights are read-only; 8GiB supports many simultaneously active models). The weight FIFO is four tiles deep. The intermediate results are held in the 24 MiB on-chip *Unified Buffer*, which can serve as inputs to the Matrix Unit. A programmable DMA controller transfers data to or from CPU Host memory and the Unified Buffer.

**CHAPTER 8**

## IMPLEMETAION OF TPU

Figure 8.1 shows the TPU on its printed circuit card, which inserts into existing servers like an SATA (Serial Advanced Technology Attachment) disk. But the card uses PCIe Gen3 x16 bus the server must have this bus for implementation.



**Figure 8.1: TPU Printed Circuit Board**

**CHAPTER 9**

## INSTRUCTIONS IN TPU

As instructions are sent over the relatively slow PCIe bus, TPU instructions follow the CISC tradition, including a repeat field. The average clock cycles per instruction (CPI) of these CISC instruction is typically 10 to 20. It has about a dozen instructions overall, but these five are the key ones:

* **Read\_Host\_Memory** : Reads data from the TPU host memory into the Unified Buffer (UB).
* **Read\_Weights** : Reads weights from Weight Memory into the Weight FIFO as input to the Matrix Unit.
* **MatrixMultiply/Convolve** : Causes the Matrix Unit to perform a matrix multiply or a convolution from the Unified Buffer into the Accumulators. A matrix operation takes a variable-sized B\*256 input, multiplies it by a 256x256 constant weight input, and produces a B\*256 output, taking B cycles to complete.
* **Activate** : Performs the nonlinear function of the artificial neuron, with options for ReLU, Sigmoid, and so on. Its inputs are the Accumulators, and its output is the Unified Buffer. It can also perform the pooling operations needed for convolutions using the dedicated hardware on the die, as it is connected to nonlinear function logic.
* **Write\_Host\_Memory** : Writes data from the Unified Buffer into the CPU host memory.

The other instructions are alternate host memory read/write, set configuration, two versions of synchronization, interrupt host, debug-tag, nop, and halt. The CISC Matrix Multiply instruction is 12 bytes, of which 3 are Unified Buffer address; 2 are accumulator address; 4 are length (sometimes 2 dimensions for convolutions); and the rest are opcode and flags.

**CHAPTER 10**

## PARALLEL POEOCESSING ON MXU

The philosophy of the TPU micro architecture is to keep the matrix unit busy. It uses a 4-stage pipeline for these CISC instructions, where each instruction executes in a separate stage. The plan was to hide the execution of the other instructions by overlapping their execution with the Matrix Multiply instruction.

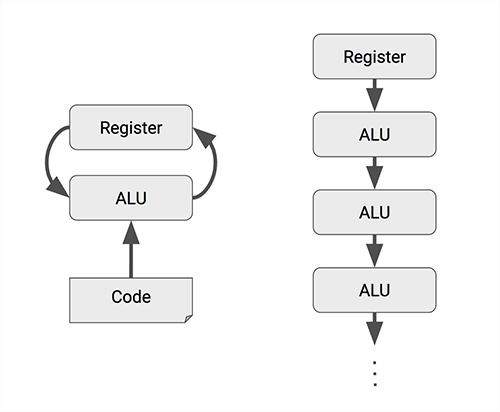
Toward that end, the Read Weights instruction follows the decoupledaccess/execute philosophy, in that it can complete after sending its address but before the weight is fetched from Weight Memory. The matrix unit will stall if the input activation or weight data is not ready. We don’t have clean pipeline overlap diagrams, because our CISC instructions can occupy a station for thousands of clock cycles, unlike the traditional RISC pipeline with one clock cycle per stage.

Interesting cases occur when the activations for one network layer must complete before the matrix multiplications of the next layer can begin; we see a “delay slot,” where the matrix unit waits for explicit synchronization before safely reading from the Unified Buffer. As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the Unified Buffer.

**10.1 How Systolic Execution Is Possible?**

To implement such a large-scale matrix processor, the MXU features a drastically different architecture than typical CPUs and GPUs, called a [systolic array.](https://en.wikipedia.org/wiki/Systolic_array) CPUs are designed to run almost any calculation; they're general-purpose computers. To implement this generality, CPUs store values in registers, and a program tells the Arithmetic Logic Units (ALUs) which registers to read, the operation to perform (such as an addition, multiplication or logical AND) and the register into which to put the result. A program consists of a sequence of these read / operate / write operations.

All of these features that support generality (registers, ALUs and programmed control) have costs in terms of power and chip area.



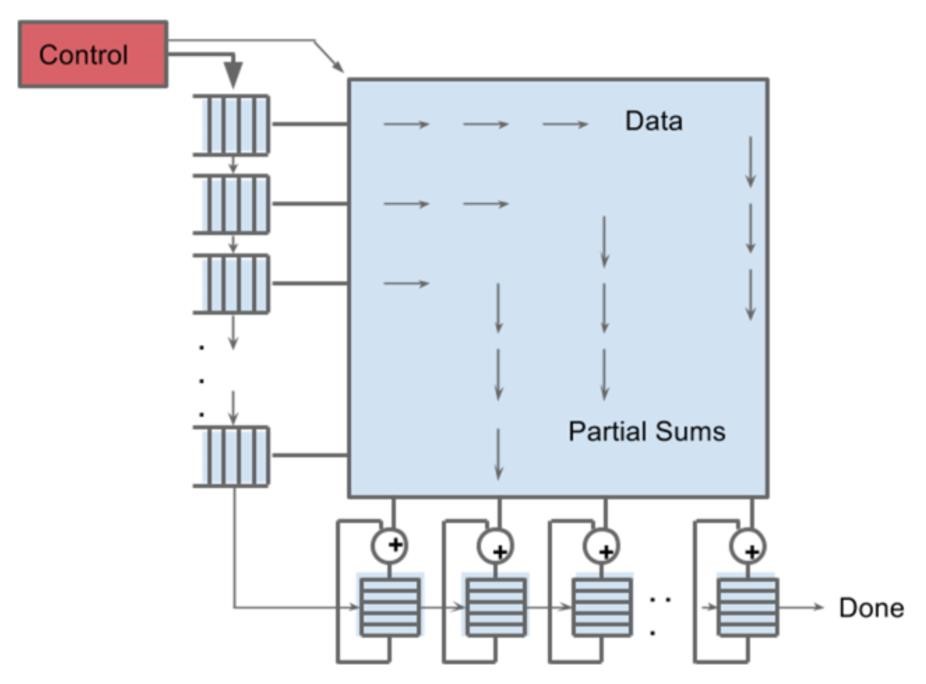
**Figure 10.1 : Systolic array chains**

For an MXU, however, matrix multiplication reuses both inputs many times as part of producing the output. We can read each input value once, but use it for many different operations without storing it back to a register. Wires only connect spatially adjacent ALUs, which makes them short and energy-efficient. The ALUs perform only multiplications and additions in fixed patterns, which simplifies their design.

The design is called systolic because the data flows through the chip in waves, reminiscent of the way that the heart pumps blood. The particular kind of systolic array in the MXU is optimized for power and area efficiency in performing matrix multiplications, and is not well suited for general-purpose computation. It makes an engineering trade off: limiting registers, control and operational flexibility in exchange for efficiency and much higher operation density.

The TPU Matrix Multiplication Unit has a systolic array mechanism that contains 256 × 256 = total 65,536 ALUs. That means a TPU can process 65,536 multiply-and-adds for 8-bit integers every cycle. Because a TPU runs at 700MHz, a TPU can compute 65,536 × 700,000,000 = 46 × 1012 multiply-and-add operations or 92 Traps per second (92 × 1012) in the matrix unit.

### 10.2 Data Flow Of The MXU



**Figure 10.2 Data flow of the Matrix Multiply Unit**

Figure 5 shows that data flows in from the left, and the weights are loaded from the top. A given 256-element multiply-accumulate operation moves through the matrix as a diagonal wave front. The weights are preloaded, and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give the illusion that the 256 inputs are read at once, and that they instantly update one location of each of 256 accumulators. From a correctness perspective, software is unaware of the systolic nature of the matrix unit, but for performance, it does worry about the latency of the unit.

**CHAPTER 11**

## TENSORFLOW TO TPU

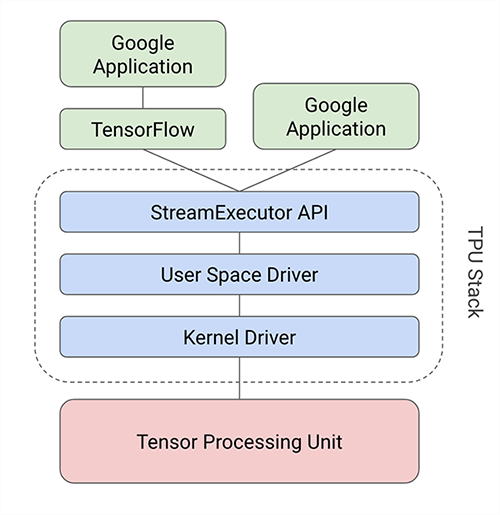
**11.1 What Is Tensorflow?**

**TensorFlow** is an open source software library released in 2015 by Google to make it easier for developers to design, build, and train deep learning models. TensorFlow originated as an internal library that Google developers used to build models in-house, and we expect additional functionality to be added to the open source version as they are tested and vetted in the internal flavor. Although TensorFlow is only one of several options available to developers, we choose to use it here because of its thoughtful design and ease of use.

At a high level, TensorFlow is a Python library that allows users to express arbitrary computation as a graph of data flows. Nodes in this graph represent mathematical operations, whereas edges represent data that is communicated from one node to another. Data in TensorFlow are represented as tensors, which are multidimensional arrays. Although this framework for thinking about computation is valuable in many different fields, TensorFlow is primarily used for deep learning in practice and research.

### 11.2 Tensorflow Graphs Into TPU Instructions

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.



**Figure 11.1 The software stack of TPU**

The TPU software stack had to be compatible with those developed for CPUs and GPUs so that applications could be ported quickly to the TPU. The portion of the application run on the TPU is typically written in TensorFlow and is compiled into an API that can run on GPUs or TPUs Like GPUs, the TPU stack is split into a User Space Driver and a Kernel Driver. The Kernel Driver is lightweight and handles only memory management and interrupts. It is designed for long-term stability. The User Space driver changes frequently. It sets up and controls TPU execution, reformats data into TPU order, translates API calls into TPU instructions, and turns them into an application binary. The User Space driver compiles a model the first time it is evaluated, caching the program image and writing the weight image into the TPU’s weight memory; the second and following evaluations run at full speed. The TPU runs most models completely from inputs to outputs, maximizing the ratio of TPU compute time to I/O time. Computation is often done one layer at a time, with overlapped execution allowing the matrix multiply unit to hide most non-critical-path operations.

**CHAPTER 12**

## GENERATIONS OF TPU

### 12.1 First Generation TPU

The first generation TPU is an [8-bit](https://en.wikipedia.org/wiki/8-bit) matrix multiply engine, driven with [CISC instructions](https://en.wikipedia.org/wiki/CISC_instruction) by the [host processor](https://en.wikipedia.org/w/index.php?title=Host_processor&action=edit&redlink=1) across a [PCIe 3.0](https://en.wikipedia.org/wiki/PCI_Express#PCI_Express_3.0) bus. It is manufactured on a [28 nm](https://en.wikipedia.org/wiki/28_nm) process with a die size ≤ 331 [mm2](https://en.wikipedia.org/wiki/Millimetre). The [clock speed](https://en.wikipedia.org/wiki/Clock_speed) is 700 [MHz](https://en.wikipedia.org/wiki/MHz) and has a [thermal design power](https://en.wikipedia.org/wiki/Thermal_design_power) of 28-40 [W.](https://en.wikipedia.org/wiki/Watt) It has 28[MiB](https://en.wikipedia.org/wiki/MiB) of on chip memory, and 4[MiB](https://en.wikipedia.org/wiki/MiB) of [32-](https://en.wikipedia.org/wiki/32-bit)

[bit](https://en.wikipedia.org/wiki/32-bit) [accumulators](https://en.wikipedia.org/wiki/Accumulator_(computing)) taking the results of a 256x256 array of 8-bit [multipliers.](https://en.wikipedia.org/wiki/Binary_multiplier) Instructions transfer data to or from the host, perform [matrix multiplies](https://en.wikipedia.org/wiki/Matrix_multiplies) or [convolutions,](https://en.wikipedia.org/wiki/Convolution) and apply [activation functions.](https://en.wikipedia.org/wiki/Activation_function)

### 12.2 Second Generation TPU

The second generation TPU was announced in May 2017. The individual TPU ASICs are rated at 45 [TFLOPS](https://en.wikipedia.org/wiki/FLOPS) and arranged into 4-chip 180 TFLOPS modules. These modules are then assembled into 256 chip pods with 11.5 PFLOPS of performance. Notably, while the first generation TPUs were limited to integers, the second generation TPUs can also calculate in floating point. This makes the second generation TPUs useful for both training and inference of machine learning models. Google has stated these second generation TPUs will be available on the [Google Compute Engine](https://en.wikipedia.org/wiki/Google_Compute_Engine) for use in [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) applications.

**CHAPTER 13**

## APPLICATION OF TPU

### 13.1 Voice/Sound Recognition

One of the most well-known uses of Tensor Processing Unit are Sound based applications. With the proper data feed, neural networks are capable of understanding audio signals. These can be:

* Voice recognition – mostly used in IoT, Automotive, Security and UX/UI
* Voice search – mostly used in Telecoms, Handset Manufacturers
* Sentiment Analysis – mostly used in CRM
* Flaw Detection (engine noise) – mostly used in Automotive and Aviation

### 13.2 Text Based Applications

Further popular uses of TPU are, text based applications such as sentimental analysis (CRM, Social Media), Threat Detection (Social Media, Government) and Fraud Detection (Insurance, Finance).

* Language Detection is one of the most popular uses of text based applications.
* We all know Google Translate, which supports over 100 languages translating from one to another. The evolved versions can be used for many cases like translating jargon legalese in contracts into plain language.

### 13.3 Image Recognition

Mostly used by Social Media, Telecom and Handset Manufacturers; Face Recognition, Image Search, Motion Detection, Machine Vision and Photo Clustering can be used also in Automotive, Aviation and Healthcare Industries. Image Recognition aims to recognize and identify people and objects in images as well as understanding the content and context.

### 13.4 Time Series

TPU Time Series algorithms are used for analyzing time series data in order to extract meaningful statistics. They allow forecasting non-specific time periods in addition to generate alternative versions of the time series.

The most common use case for Time Series is **Recommendation.** You’ve probably heard of this use from Amazon, Google, Facebook and Netflix where they analyze customer activity and compare it to the millions of other users to determine what the customer might like to purchase or watch. These recommendations are getting even smarter, for example, they offer you certain things as gifts (not for yourself) or TV shows that your family members might like.

### 13.5 Video Detection

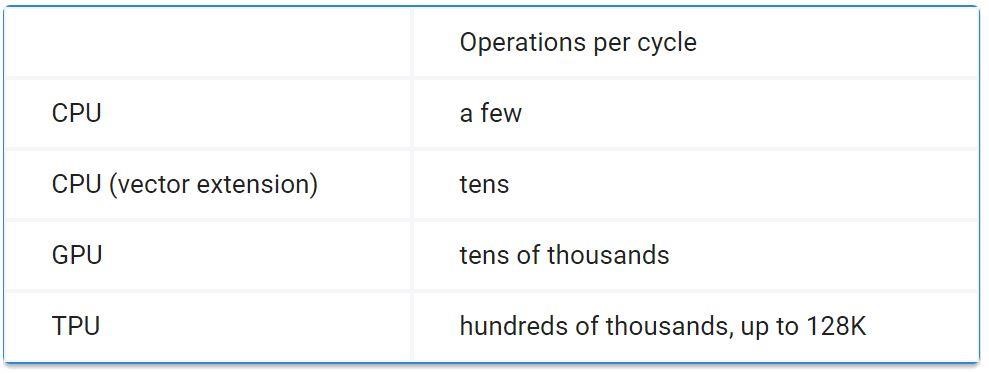
TPU’s neural networks also work on video data. This is mainly used in Motion Detection, Real-Time Thread Detection in Gaming, Security, Airports and UX/UI fields. Recently, Universities are working on Largescale Video Classification datasets like YouTube-8M aiming to accelerate research on large-scale video understanding, representation learning, noisy data modeling, transfer learning, and domain adaptation approaches for video.

**CHAPTER 14**

## ADVANTAGE OF TPU

### 14.1 Number of Operation Per Cycle

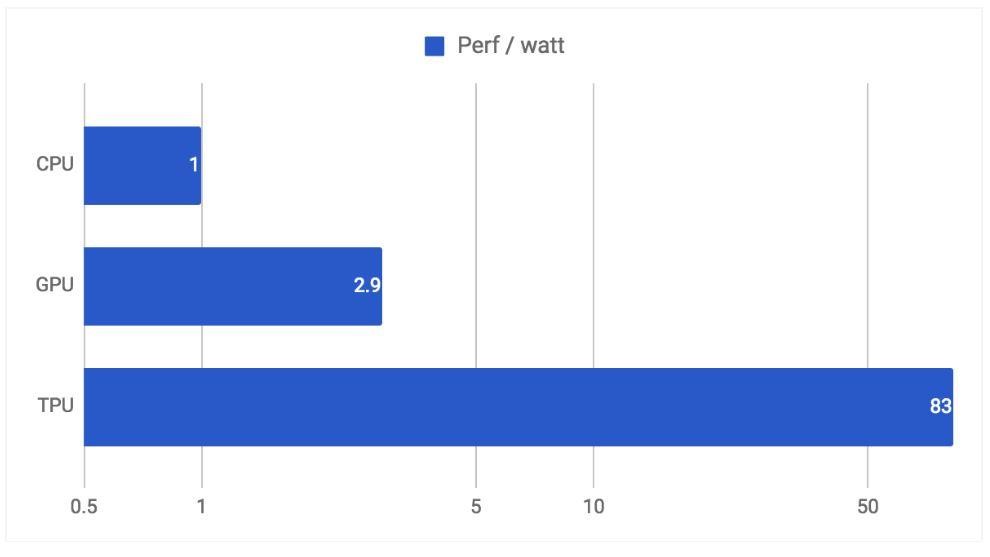
Let's compare the number of operations per cycle between CPU, GPU and TPU.



**Table 14.1: Number of Operation Per Cycle**

In comparison, a typical RISC CPU without vector extensions can only execute just one or two arithmetic operations per instruction, and GPUs can execute tens of thousands of operations per instruction. With the TPU, a single cycle of a Matrix multiply instruction can invoke hundreds of thousands of operations.

During the execution of this massive matrix multiply, all intermediate results are passed directly between 64K ALUs without any memory access, significantly reducing power consumption and increasing throughput. As a result, the CISC-based matrix processor design delivers an outstanding performance-per-watt ratio: TPU provides a **83X** better ratio compared with contemporary CPUs and a **29X**better ratio than contemporary GPUs.



**Figure 14.1: Performance/Watt Relative to Contemporary CPUs and GPUs**

#### 14.2 Minimal and Deterministic Design

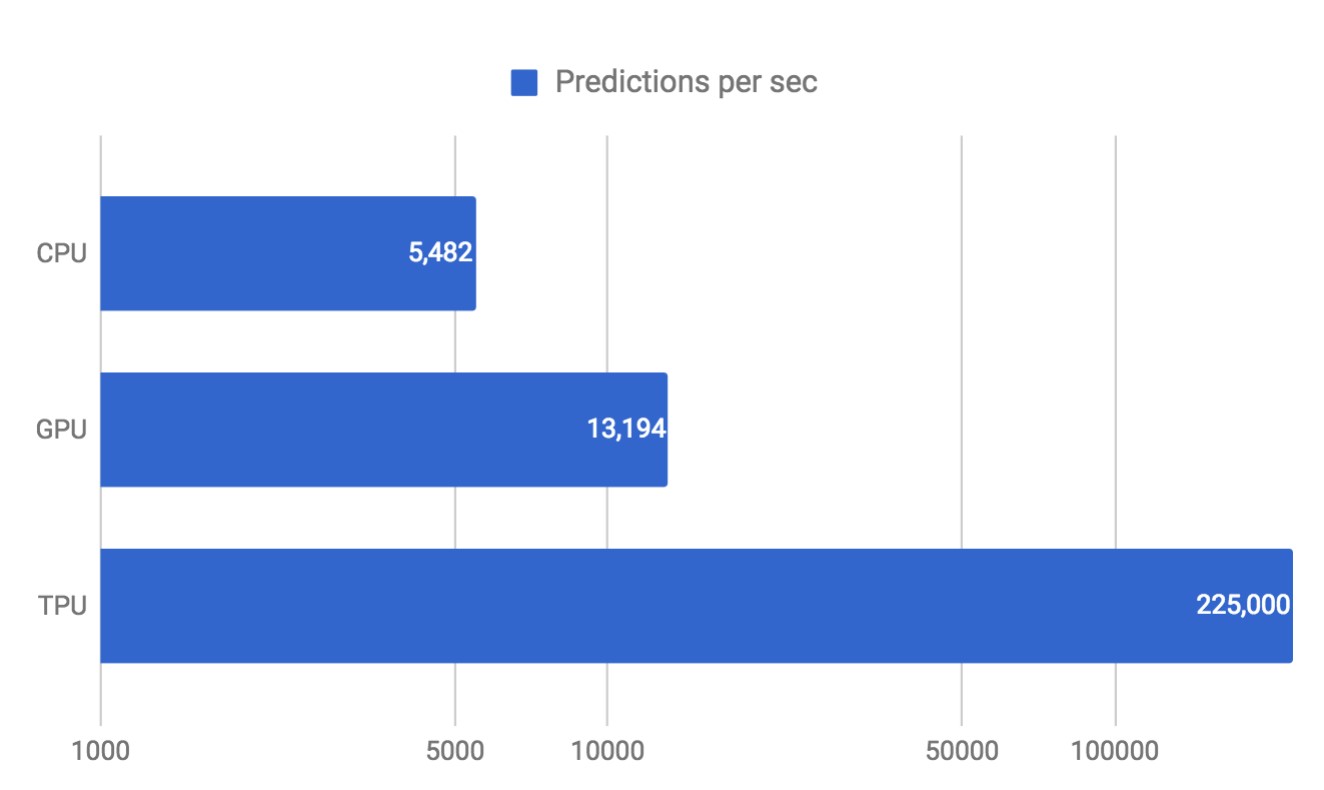
General-purpose processors such as CPUs and GPUs must provide good performance across a wide range of applications, they have evolved myriad sophisticated, performance-oriented mechanisms. As a side effect, the behavior of those processors can be difficult to predict, which makes it hard to guarantee a certain latency limit on neural network inference. In contrast, TPU design is strictly **minimal and deterministic** as it has to run **only one task at a time: neural network prediction**. You can see its simplicity in the floor plan of the TPU die.

If you compare the floor plans (Figure of CPUs and GPUs with TPU, you'll notice the red parts (control logic) are much larger (and thus more difficult to design) for CPUs and GPUs since they need to realize the complex constructs and mechanisms mentioned above. In the TPU, the control logic is minimal and takes under 2% of the die.

More importantly, despite having many more arithmetic units and large on-chip memory, the TPU chip is half the size of the other chips. Since the cost of a chip is a function of the area3 — more [smaller chips per silicon wafer](http://anysilicon.com/die-per-wafer-formula-free-calculators/) and higher yield for small chips since they're less likely to have manufacturing defects**\*** — halving chip size reduces chip cost by roughly a factor of 8 (23).

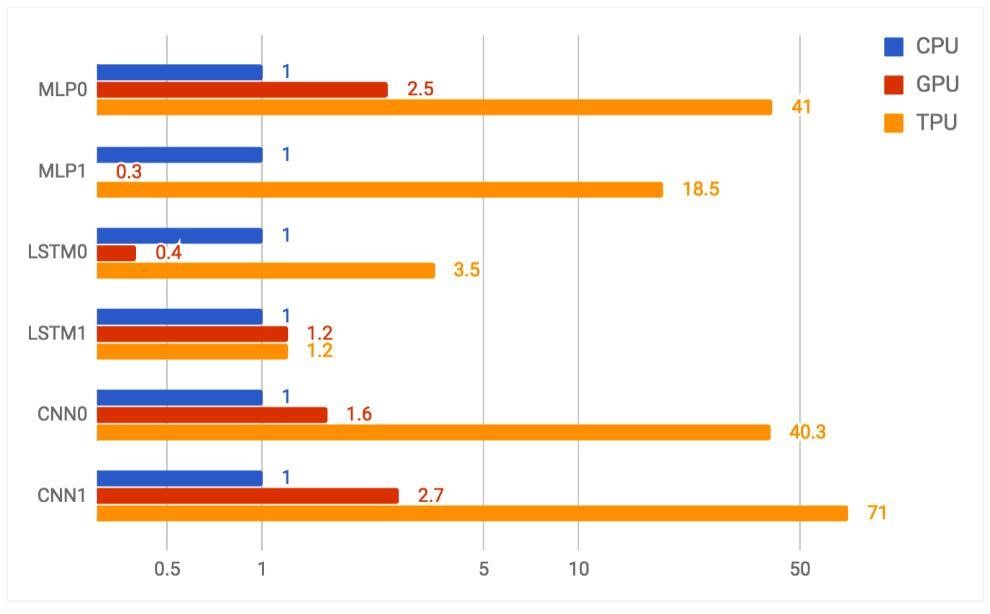
#### 14.3 Make Prediction

With the TPU, we can easily estimate exactly how much time is required to run a neural network and make a prediction. This allows us to operate at near-peak chip throughput while maintaining a strict latency limit on almost all predictions. For example, despite a strict **7ms** limit in the above-mentioned application, the TPU delivers **15–30X** more throughput than contemporary CPUs and GPUs.



**Figure 14.2: Throughput Under 7 ms Latency Limit**

We use neural network predictions to support end-user-facing products and services, and everyone knows that users become impatient if a service takes too long to respond. Thus, for the MLP0 application, we limit the 99th-percentile prediction latency to around 7 ms, for a consistently fast user experience from TPU-based Google services. The following is an overall performance (predictions per second) comparison between the TPU and a contemporary CPU and GPU across six neural network applications under a latency limit. In the most spectacular case, the TPU provides 71X performance compared with the CPU for the CNN1 application.



**Figure 14.3: CPU, GPU and TPU Performance On Six Reference Workloads**

29

## CONCLUSION

Although most researches are optimizing TPU, they represent just 5% of goolge’s datacenter workload. Since Inference apps are user-facing, they emphasize response-time over throughput. Due to latency limits, the K80 GPU is just a little faster than the CPU, for inference. Despite having a much smaller and lower power chip, the TPU has 25 times as many MACs and 3.5 times as much on-chip memory as the K80 GPU. The TPU is about 15X – 30X faster at inference than the K80 GPU and Haswell CPU. Four of the six NN apps are memory-bandwidth limited on the TPU; if the TPU were revised to have the same memory system as the K80 GPU, it would be about 30X – 50X faster than the GPU and CPU. The performance/Watt of TPU is 30x – 80X that of contemporary products; the revised TPU with K80 memory would be 70X – 200X better.

30

## REFERENCES

1. https://www.nextplatform.com/2017/04/05/first-depth-look-googles-tpuarchitecture/

1. https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googlesfirst-tensor-processing-unit-tpu

1. [https://www.extremetech.com/computing/247199-googles-dedicatedtensorflow-processor-tpu-makes-hash-intel-nvidia-inference-workloads](https://www.extremetech.com/computing/247199-googles-dedicated-tensorflow-processor-tpu-makes-hash-intel-nvidia-inference-workloads)

1. <https://en.wikipedia.org/wiki/Tensor_processing_unit>

1. [https://techcrunch.com/2017/05/17/google-announces-second-generation-oftensor-processing-unit-chips/](https://techcrunch.com/2017/05/17/google-announces-second-generation-of-tensor-processing-unit-chips/)

1. In-Datacenter Performance Analysis of a Tensor Processing Unit TM paper by Google.

1. http://www.techradar.com/news/computing-components/processors/google-stensor-processing-unit-explained-this-is-what-the-future-of-computing-lookslike-1326915