Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit Register File	1	10
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit SRAM Cache	5	50
32 bit DRAM Memory	**640**	**6400**

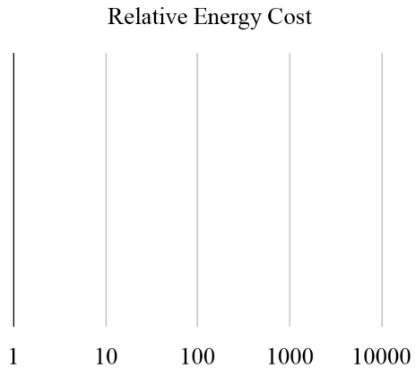

Figure 1: Energy table for 45nm CMOS process
(Adopted figure from [1])

The increasing popularity of DNNs for classification tasks such as computer vision, speech recognition and natural language processing has promted work to accelerate execution using specialised hardware. AI accelerators tend to prioritise improving the performance of networks from two perspectives; increasing compuational throughput, and decreasing energy consumption. Energy consumption is critical to the feasibility of performing inference on mobile devices, the dominant factor in this area is memory access, figure. 1 shows the energy consumption for a 32 bit floating point add operation and a 32 bit DRAM memory access on a 45nm CMOS chip, they note that DRAM memory access is 3 orders of magnitude of an add operation. Hardware is commonly referred to as an AI accelerator, these can be built to accelerate both the *training* and *inference* stages of execution, this section will specifically focus on the *inference* phase, however some of the following are capable of both.

0.0.1 VPU

One commercial hardware accelerator using a VPU architecture is the Intel Movidius Neural Compute Stick. It is a specialised SoC for computer vision applications, with a peak floating-point computational throughput of 1 TOPS, because of reasons described in Section **??** this peak throughput will be hard to achieve in any real world scenario.

- 16 VLIW (very long instruction word) SHAVE (streaming hybrid architecture vector engine) processors, optimized for machine vision and able to run parts of a neural network in parallel.

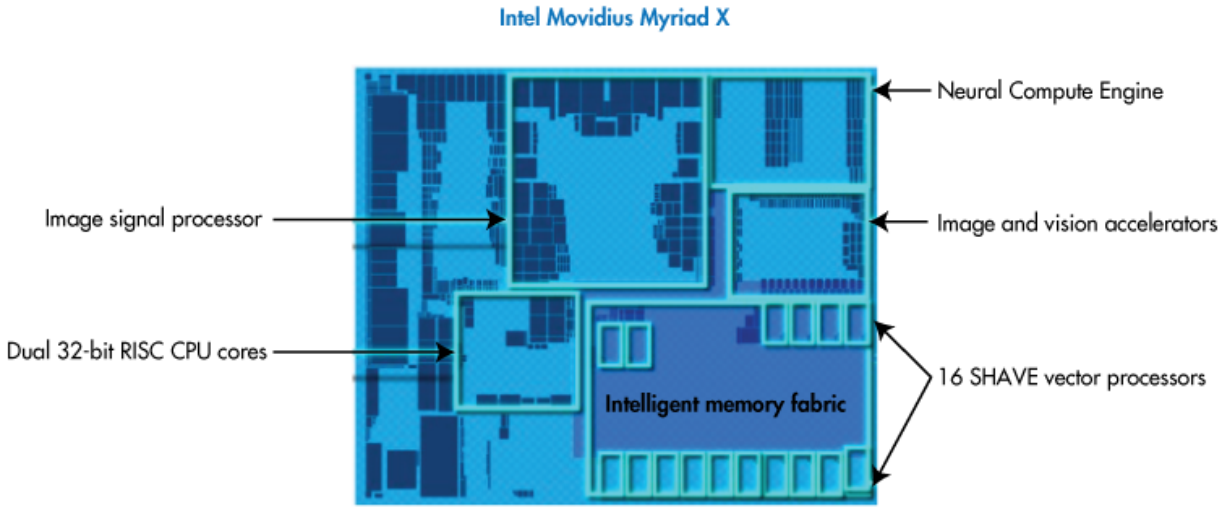

Figure 2: High level view of the Intel Movidius Myriad X VPU

- 2.5 MB On-Chip memory allowing for up to 400GB/s of internal bandwidth.
- 4Gb LPDDR4 DRAM

A key advantage of using hardware like the VPU is a customised computation pipeline that is optimised for high parallelism during inference. This however comes with the caveat that the OpenVINO framework is required to perform inference.

0.0.2 TPU

The TPU is a custom ASIC developed by google, designed specifically for TensorFlow, conventional access to these chips is via a cloud computing service. Google claims [2] the latest 4th generation TPUv4 is capable of more than double the matrix multiplication TFLOPs of TPUv3 (Wang et al. [3] describes a peak of 420 TFLOPs for the TPUv3). The TPU implements data parallelism in a manner prioritising batch size, one batch of training data is split evenly and sent to each core of the TPU, so total on-board memory determins the maximum data batch size. Each TPU core has a complete copy of the model in memory, so the maximum size of the model is determined by the amount of memory avaliable to each core [3].