2019年3月26日 星期二

标题

Eyeriss

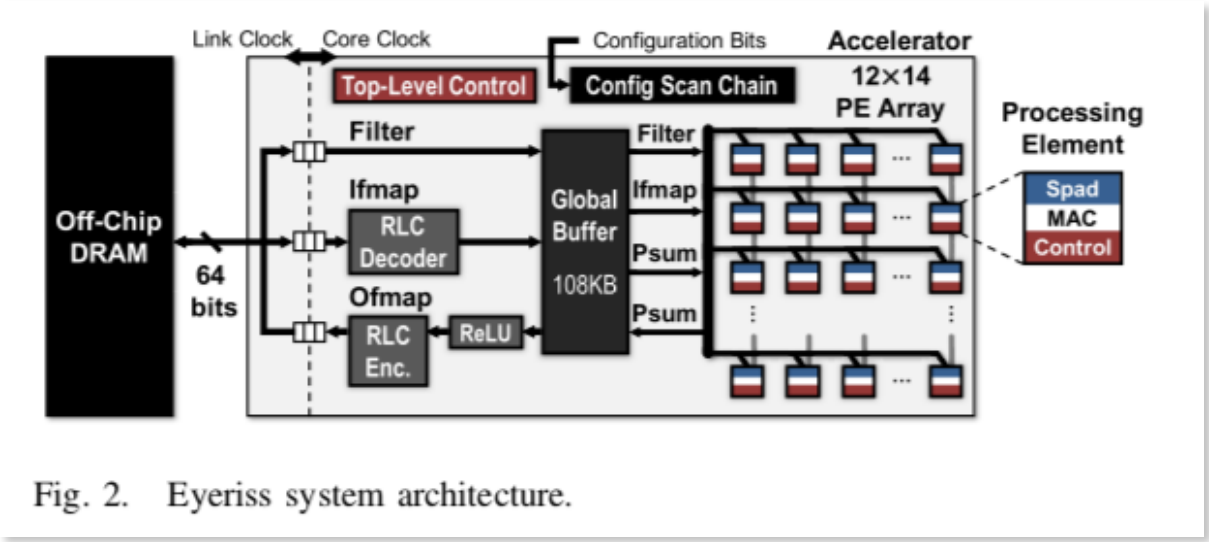

Fig. 2. Eyeriss system architecture.

1.上图是系统整体的架构

——2个时钟频率（link 和 core 时钟频率），两边独立，通过FIFO的交互传递数据

——168个PE(process element)，12x14的阵列，108kb的全局缓冲、一个ReLU激活函数组件，Run-length Compression RLC模块（用来压缩数据中的0）

——四个信息层次：DRAM、GLB、PE间通信、spads(PE内部的存储单元）

2.系统控制：

——顶层：1.DRAM和GLB；2.GLB和PE；3.RLC和ReLU模块的控制

——底层：PE的控制（各自独立，无需同步）

从DRAM读入数据，建立体系，图像输入，处理后送回DRAM

3.节能——两个路径

——1.减少数据移动　　2.利用数据统计特征

——A.行平稳数据流

2019年3月26日 星期二

将CNN中的算法映射到PE中的权重相连，来提升计算并行性和节约能量

DRAM、GLB的数据操作耗能大，所以尽量利用PE间通讯和PE内的spads

数据类型（输入图像、过滤器、和/输出图像）

平时的问题所在：输入图像和过滤器的卷积重用、与、和的积累运算无法同时进行。

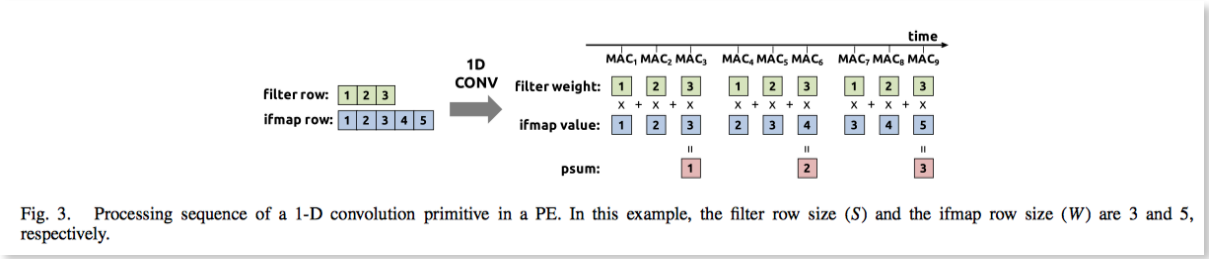

Fig. 3. Processing sequence of a 1-D convolution primitive in a PE. In this example, the filter row size (S) and the ifmap row size (W) are 3 and 5, respectively.

一维卷积如图：PE中的spads存“过滤器的行”，从而高效重用filter 和psum

尺寸取决于过滤器行数

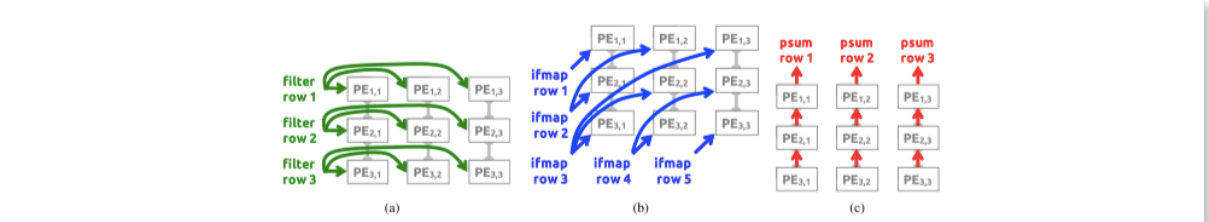

Fig. 4. Dataflow in a PE set for processing a 2-D convolution. (a) Rows of filter weights are reused across PEs horizontally. (b) Rows of ifmap values are reused across PEs diagonally. (c) Rows of psums are accumulated across PEs vertically. Reuse and accumulation of data within a PE set reduce accesses to the GLB and DRAM, saving data movement energy cost. In this example, the number of filter rows (R), ifmap rows (H), and ofmap rows (E) are 3, 5, and 3, respectively. Therefore, the PE set size is 3×3. Filter and ifmap values from different rows are sent to the PE set in a time-interleaved fashion; all the PEs that reuse the same value receive it at the same cycle. The psums generated from one PE are sent to its neighbor PE immediately.

二维卷积：（借助一维图可更好理解）尺寸 RxE (R过滤器行数，E输出图像行数)

（若数量大于168，可取e<E,分布做）

（若数量小于168，但高度太高，目前没有好的办法（maybe 分割做），所以目前其只支持最大高度12）　　下图是一些例子

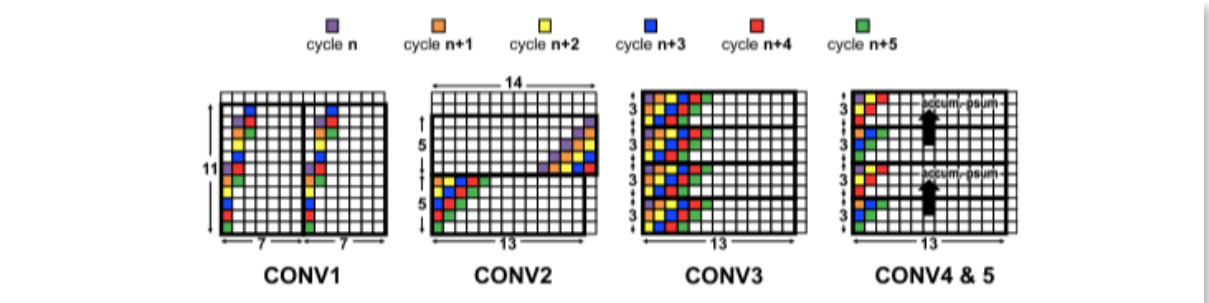

Fig. 5. Mapping of the PE sets on the spatial array of 168 PEs for the CONV layers in AlexNet. For the colored PEs, the PEs with the same color receive the same ifmap value in the same cycle. The arrow between two PE sets indicates that their psums can be accumulated together.

若要求的尺寸小，从而映射多个有好处，可以减少移入数据，数据重用提高，直接计算psum

多维的卷积

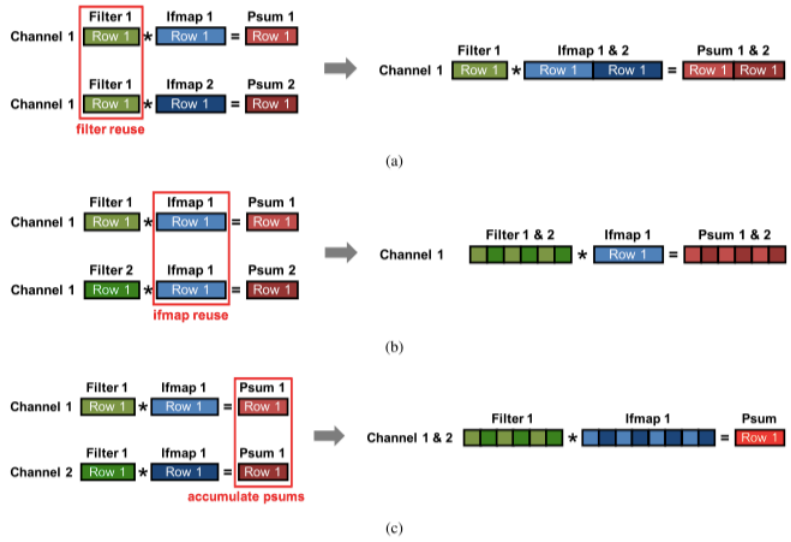

Fig. 6. Handling the dimensions beyond 2-D in each PE by (a) concatenating the ifmap rows, each PE can process multiple 1-D primitives with different ifmaps and reuse the same filter row and (b) time interleaving the filter rows, each PE can process multiple 1-D primitives with different filters and reuse the same ifmap row. (c) By time interleaving the filter and ifmap rows, each PE can process multiple 1-D primitives from different channels and accumulate the psums together.

(a)(b)分别说明了PE中的过滤器时间复用、和、输入图像时间复用

(c)二者可以整合，从而使多维的卷积可以高效重复利用数据

主要就是PE中的过滤器时间交错，切换；然后输入图像的行，不同图像交错。从而可以重复利用filter、ifmap、psum；

由于复用：需提高filter spad、 ifmap spad、psum spad

每个PE持有pxq个单位，q个通道，p个过滤器

different filters. The required spad capacity for each data type is: 1) $p \times q \times S$ for the rows of filter weights from q channels of p filters; 2) $q \times S$ for q sliding windows of ifmap values from q different channels; and 3) p for the accumulation of psums in p ofmap channels. In Eyeriss, where ifmap spad is

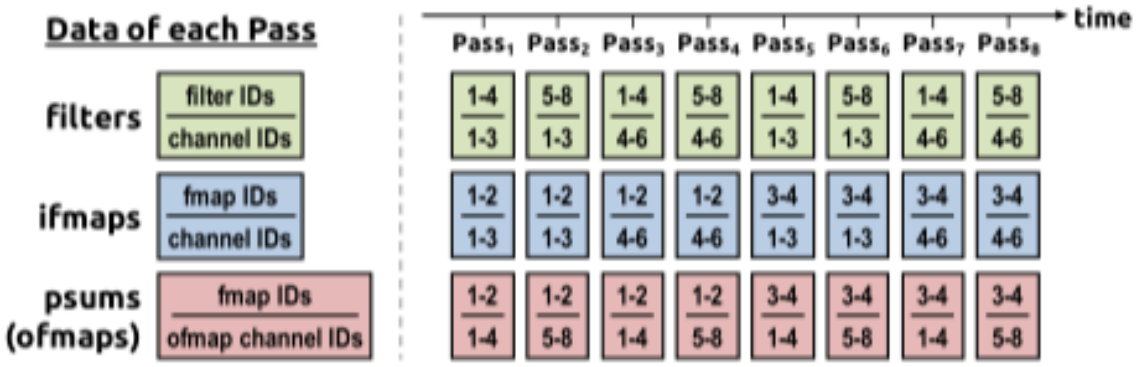

Fig. 7. Scheduling of processing passes. Each block of filters, ifmaps, or psums is a group of 2-D data from the specified dimensions used by a processing pass. The number of channels (C), filters (M), and ifmaps (N) used in this example layer created for demonstration purpose are 6, 8, and 4, respectively, and the RS dataflow uses eight passes to process the layer.

重利用的要求，决定了数在GLB中存在的长短，所以也就决定了GLB要求的大小。

TABLE II

MAPPING PARAMETERS OF THE RS DATAFLOW

Parameter	Description
m	number of ofmap channels stored in the global buffer
n	number of ifmaps used in a processing pass
e	width of the PE set (strip-mined if necessary)
p	number of filters processed by a PE set
q	number of channels processed by a PE set
r	number of PE sets that process different channels in the PE array
t	number of PE sets that process different filters in the PE array

影响映射的硬件参数

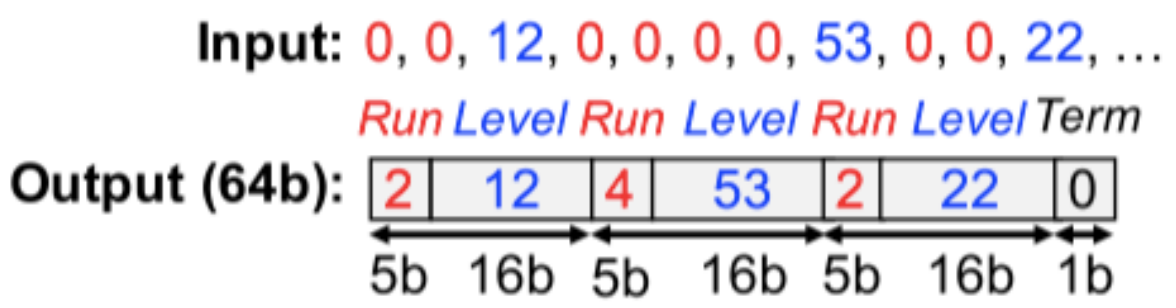

Fig. 8. Encoding of the RLC.

为了节省带宽，来减少GLB和DRAM间的通信，以节约能量，采用RLC来探索数据中的0，压缩数据，使传输量减少，两头都用，进ifmap和出psum;

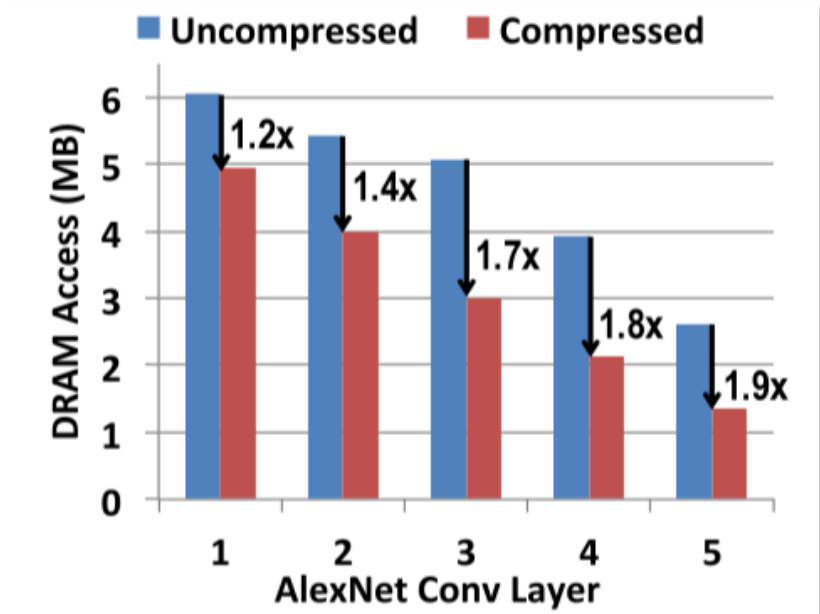

Fig. 9. Comparison of DRAM accesses (read and write), including filters, ifmaps, and ofmaps, before and after using RLC in the five CONV layers of AlexNet.

系统模块：

1.GLB全局缓冲：

2.片上网络 network-on-chip：

（1）全局输入网GIN

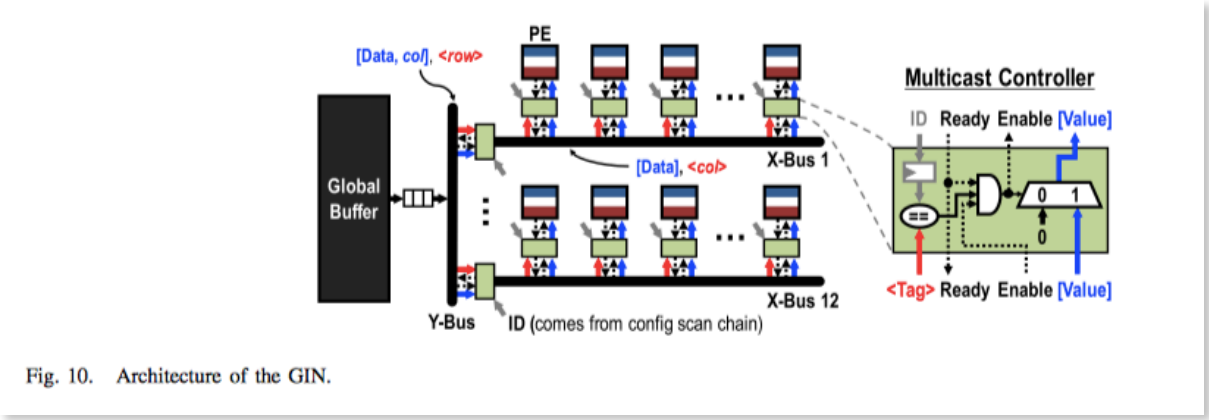

Fig. 10. Architecture of the GIN.

（2）GON全局输出网（结构同GIN）

（3）在同一列的连续两行上的每一对PE之间，实现一个专用的64b数据总线，将psum从底层PE直接传递到顶层PE。因此，一个PE可以从psum GIN或LN中获得它的输入psum。

3.处理单元（processing element)和数据门（data gating)

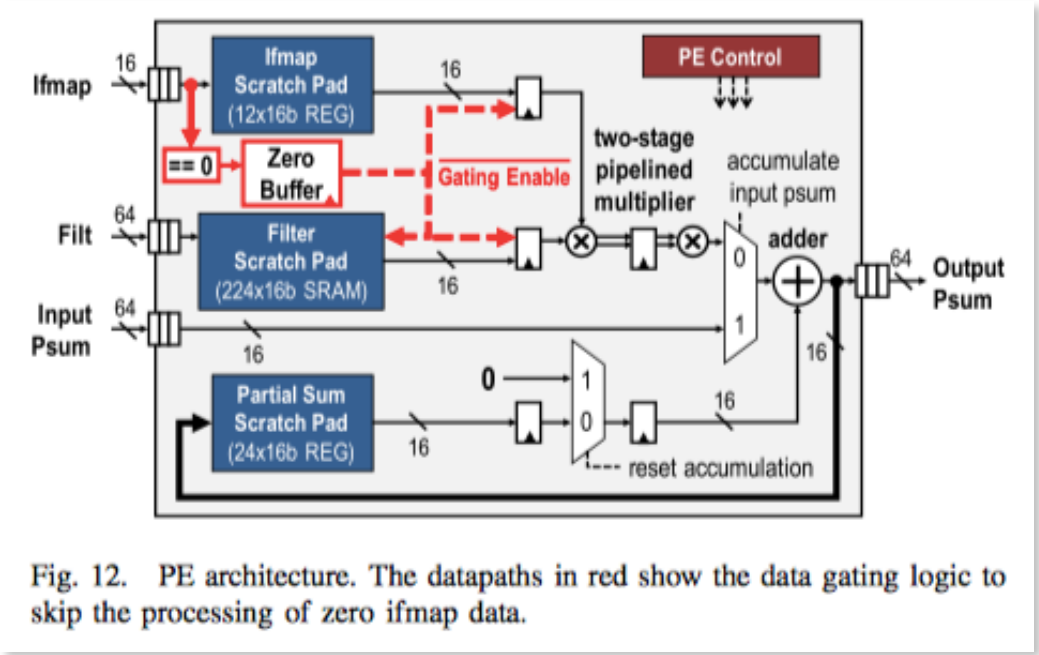

Fig. 12. PE architecture. The datapaths in red show the data gating logic to skip the processing of zero ifmap data.

处理单元的结构如图（很清楚）

实现数据门控逻辑来充分利用ifmap中的零来节省处理能力。一个额外的12-b零缓冲区用于记录ifmap spad中零的位置。如果从zero缓冲区检测到一个zero ifmap值，则选通逻辑将禁用过滤器spad的读取，并防止MAC数据路径切换。与没有数据门控逻辑的PE设计相比，可以节省45%的PE功耗。

下图是芯片的参数

TABLE IV
CHIP SPECIFICATIONS

Technology	TSMC 65nm LP 1P9M
Chip Size	4.0 mm × 4.0 mm
Core Area	3.5 mm × 3.5 mm
Gate Count (logic only)	1176k (2-input NAND)
On-Chip SRAM	181.5K bytes
Number of PEs	168
Global Buffer	108.0K bytes (SRAM)
Scratch Pads (per PE)	filter weights: 448 bytes (SRAM) feature maps: 24 bytes (Registers) partial sums: 48 bytes (Registers)
Supply Voltage	core: 0.82–1.17 V I/O: 1.8 V
Clock Rate	core: 100–250 MHz link: up to 90 MHz
Peak Throughput	16.8–42.0 GMACS
Arithmetic Precision	16-bit fixed-point
Natively Supported CNN Shapes	filter height (R): 1–12 filter width (S): 1–32 num. of filters (M): 1–1024 num. of channels (C): 1–1024 vertical stride: 1, 2, 4 horizontal stride: 1–12

星期二

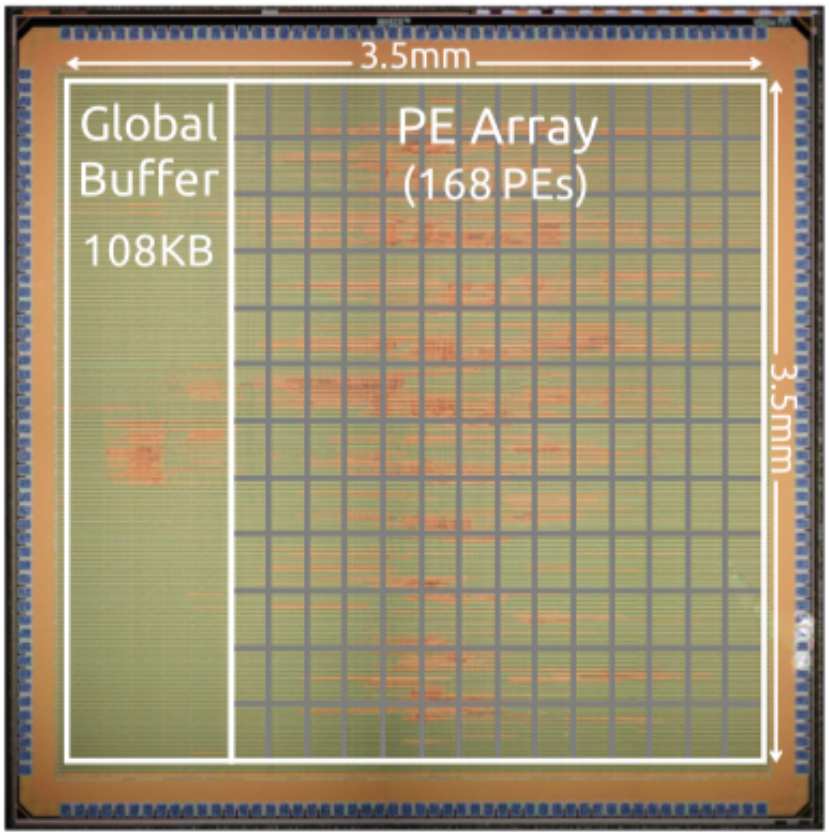

Fig. 13. Die micrograph and floorplan of the Eyeriss chip.

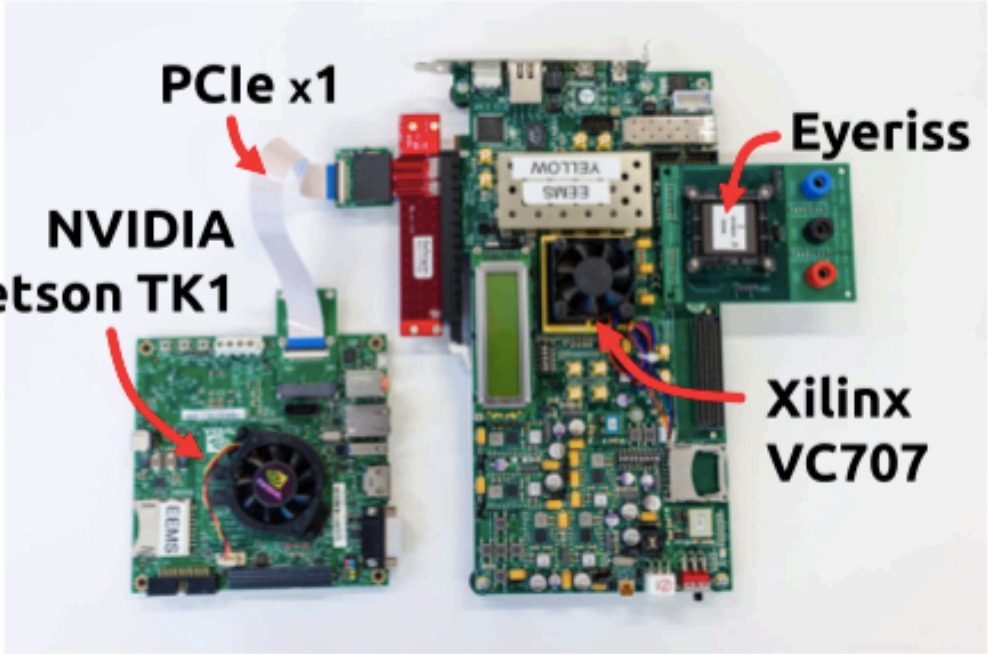

Fig. 14. Eyeriss-integrated deep-learning system that runs Caffe [28], which is one of the most popular deep-learning frameworks. The customized Caffe runs on the NVIDIA Jetson TK1 development board, and offloads the processing of a CNN layer to Eyeriss through the PCIe interface. The Xilinx VC707 serves as the PCIe controller and does not perform any processing. We have demonstrated an 1000-class image classification task [27] using this system, and a live demo can be found in [29].