Homework 2
CS6490

Darpan Gaur
CO21BTECH11004

Part 2

SCALE-Sim (SystoliC AcceLErator SIMulator) is a cycle-accurate simulator for DNN accelerators that models compute performance, memory accesses, and interface bandwidth for a given neural network. SCALE-Sim consists of two key components:

- Compute unit based on a systolic array, configurable in size and aspect ratio
- Simple accelerator memory system with three double-buffered SRAMs for storing two input operands and one result.

The simulator takes as input the layer dimensions of a neural network and hardware architecture parameters and can model both scale-up (single partition) and scale-out (multi-partition) configurations.

Implementation

- **Cycle accurate address**: Generates address for elements that goes to PE such that PE array never stalls.
- **Traffic trace parsing**: Determine total runtime for compute and data transfer to and from SRAM.
- **Buffer management**: Uses double buffer mechanism such that data transfer and compute can be done in parallel and no SRAM request is a miss.
- **Bandwidth estimation**: DRAM traces help estimate bandwidth requirements for the given workload and architecture.
- **Compute metric estimation**: Parses trace data to determine on-chip/off-chip requests, compute efficiency, mapping efficiency, and stall cycles.

Part 3

For each run of the SimScale, following files are generated:

- Bandwidth report: Contains average SRAM and DRAM bandwidth per layer for ifmap, filter and ofmap.
- Compute report: Contains total cycles, stall cycles, compute utilization, and mapping efficiency for each layer.
- Access report: Contains SRAM and DRAM ifmap, filter and ofmap reads, start cylce and end cycle for each layer. Also writes for ofmap.

Metrics in bandwidth report and compute report will be used for analysis. Also time to run the simulation will be noted.
Comparing bandwidth among differnet configs:

- OFMAP DRAM bandwidth is highest for google (∼250) than scale and eyeriss (∼10).
- SRAM bandwiths for IFMAP and FILTER are highest for google than scale and lowest for eyeriss.
- OFMAP SRAM bandwidth is highest for google than eyeriss and lowest for scale.
- There is no significant difference in FILTER and IFMAP DRAM bandwidth among the configs.

eyeriss config

Compute Report

Figure 1, 2 show the utilization and efficiency, and total cycles for eyeriss configuration.

- For yolo_tiny, total cycles is highest and for googlenet is lowest.
- Compute utilization is highest for FasterRCNN and lowest for mobilenet.
- Mapping efficiency is highest for alexnet and lowest for mobilenet.

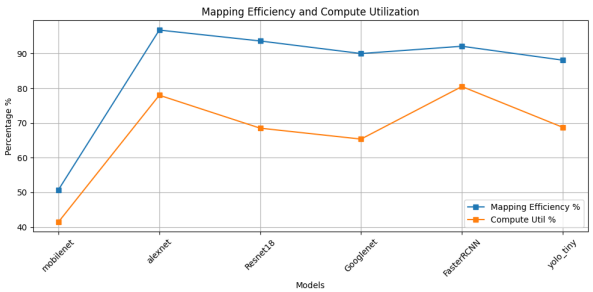

Figure 1: Utilization and efficiency (eyeriss)

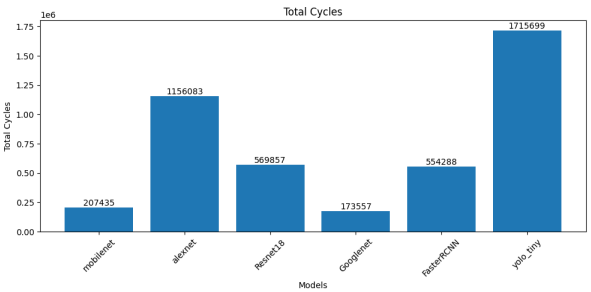

Figure 2: Total cycles (eyeriss)

Bandwidth Report

Table 1 shows the bandwidth report for eyeriss configuration.

	mobilenet	alexnet	ResNet18	Googlenet	FasterRCNN	yolo_tiny
IFMAP SRAM	9.769362	10.110527	9.189589	9.204767	10.570606	9.711927
FILTER SRAM	0.349855	0.692984	1.019175	0.966143	0.419639	0.746595
OFMAP SRAM	6.336004	11.538326	10.184142	9.805424	11.952654	10.326550
IFMAP DRAM	9.605320	10.082670	8.864435	9.387508	10.370170	9.053395
FILTER DRAM	5.995087	0.706960	1.967918	3.354012	2.335077	2.497403
OFMAP DRAM	8.003153	11.545676	10.242832	10.109779	11.995738	10.353721

Table 1: Bandwidth Report (eyeriss)

google config

Compute Report

Figure 3, 4 show the utilization and efficiency, and total cycles for google configuration.

- Total cycles is highest for yolotiny and lowest for googlenet.
- Mapping efficiency is higher for FasterRCNN and alexnet and lower for mobilenet and googlenet.
- Compute utilization varies from 9-18% for all the networks.

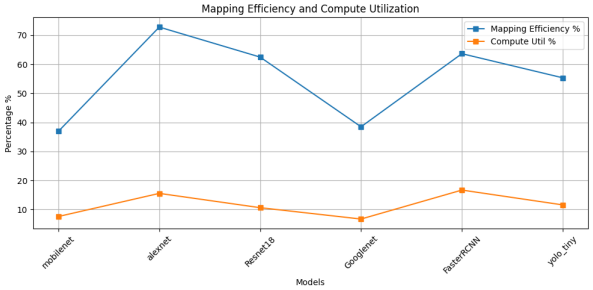

Figure 3: Utilization and efficiency (google)

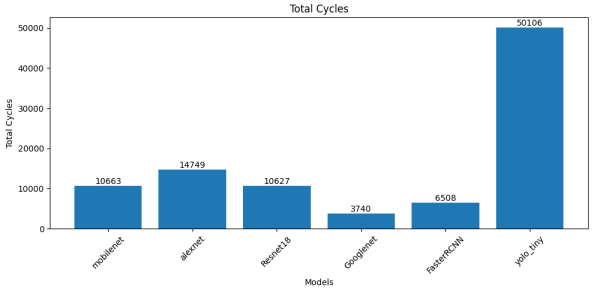

Figure 4: Total cycles (google)

Bandwidth Report

Table 2 shows the bandwidth report for google configuration.

	mobilenet	alexnet	ResNet18	Googlenet	FasterRCNN	yolo_tiny
IFMAP SRAM	64.075448	69.034058	69.457337	51.617591	80.201418	70.926288
FILTER SRAM	23.794946	46.430099	42.778710	26.027584	37.575371	35.931976
OFMAP SRAM	33.475057	53.701242	41.258157	24.960963	72.349675	44.120936
IFMAP DRAM	8.106595	6.145537	6.554707	5.913715	8.129366	8.378022
FILTER DRAM	3.316820	8.728496	7.318920	5.861866	6.454707	7.866377
OFMAP DRAM	246.665867	271.700381	255.699808	255.398188	251.644089	170.471391

Table 2: Bandwidth Report (google)

scale config

Compute Report

Figure 5, 6 show the utilization and efficiency, and total cycles for scale configuration.

- Total cycles are highest for yolo_tiny and lowest for googlenet.
- Compute utilization is highest for alexnet and lowest for mobilenet.
- Mapping efficiency is highest for FasterRCNN and lowest for mobilenet.

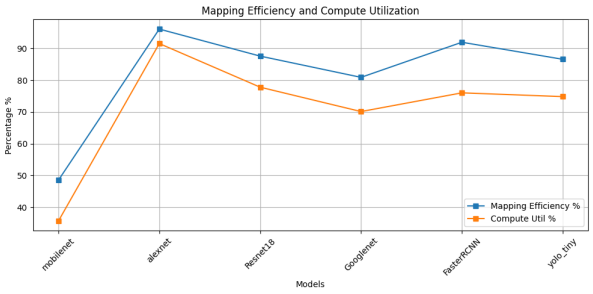

Figure 5: Utilization and efficiency (scale)

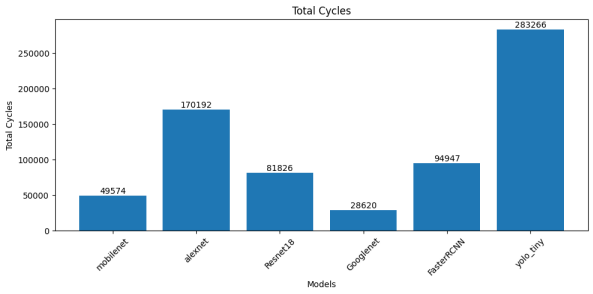

Figure 6: Total cycles (scale)

Bandwidth Report

Table 3 shows the bandwidth report for scale configuration.

	mobilenet	**alexnet**	**ResNet18**	**Googlenet**	**FasterRCNN**	**yolo_tiny**
IFMAP SRAM	23.872831	29.284604	24.890660	23.446267	24.795266	24.539143
FILTER SRAM	12.793304	30.512103	28.535848	26.772964	26.091816	26.311474
OFMAP SRAM	2.280134	0.800293	1.726237	1.909498	2.802965	2.103390
IFMAP DRAM	14.620969	5.272993	7.654913	15.133052	16.470893	7.357785
FILTER DRAM	9.109089	24.652434	18.770750	11.125210	13.675473	19.151726
OFMAP DRAM	16.049865	7.508420	11.633236	24.657264	4.997324	5.571559

Table 3: Bandwidth Report (scale)

Part 4

Change the dataflow architecture

Dataflow architecture used: WS, IS, OS
Figure 7, 8, 9 show the bandwidth for mobilenet, FasterRCNN, and ResNet18 respectively.

- WS: Higher bandwidth for IFMAP and OFMAP, lower for FILTER.
- IS: Higher bandwidth for FILTER and OFMAP, lower for IFMAP.
- OS: For DRAM bandwidth almost same for all, for SRAM higest for IFMAP and lowest for OFMAP.

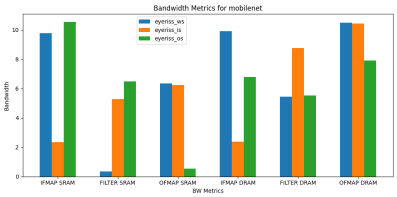

Figure 7: mobilenet

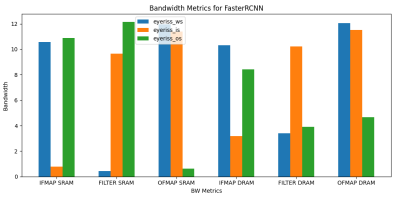

Figure 8: FasterRCNN

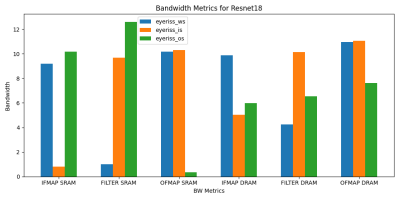

Figure 9: ResNet18

Figure 10, 11, 12 show the total cycles, compute utilization, and mapping efficiency for mobilenet, FasterRCNN, and ResNet18 respectively.

- OS has highest compute utilization and less cycles as compared to IS and WS for all the networks.
- IS has higher mapping efficiency as compared to OS and WS for mobilenet and FasterRCNN.

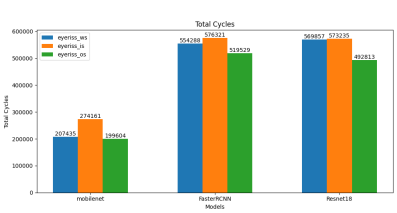

Figure 10: Total cycles

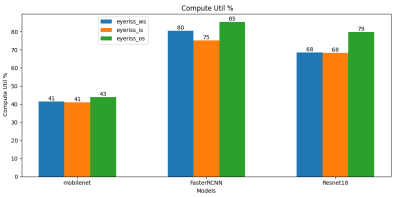

Figure 11: Compute Utilization

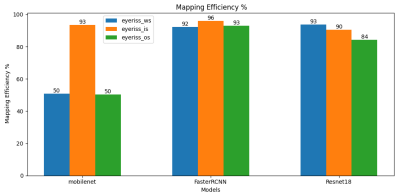

Figure 12: Mapping Efficiency

Change the size of SRAM

SRAM size used: 36KB, 108KB, 216KB, 324KB
Figure 13, 14, 15 show the bandwidth for mobilenet, FasterRCNN, and ResNet18 respectively.

- For DRAM bandwidth of FILTER and OFMAP there is increase as SRAM size increases for all the models. Especially for FILTER DRAM bandwidth.
- There is no change in SRAM bandwidth for IFMAP, OFMAP and FILTER.

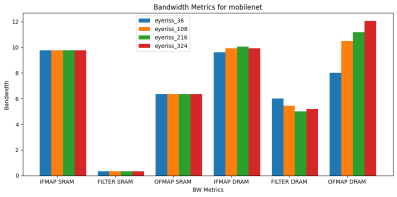

Figure 13: mobilenet

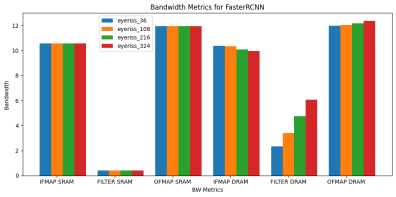

Figure 14: FasterRCNN

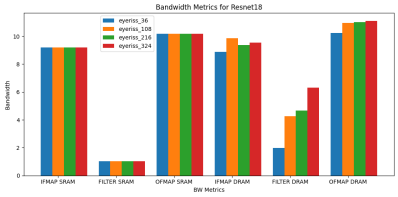

Figure 15: ResNet18

Change the array size

Array size used: H[12, 24, 48, 96], W[14, 28, 56, 112]
Figure 16, 17, 18 show the bandwidth for mobilenet, FasterRCNN, and ResNet18 respectively.

- Bandwidth increases as array size increases for all the models, and is highest for H=96, W=112.
- This is becase as array size increases, more data can be processed in parallel.

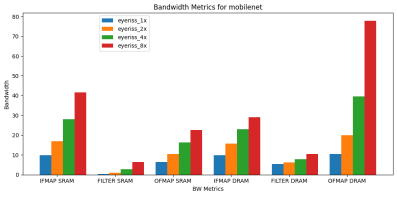

Figure 16: mobilenet

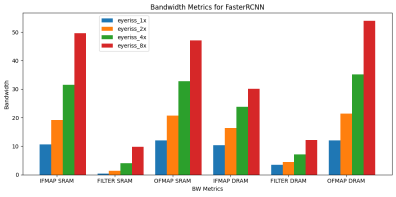

Figure 17: FasterRCNN

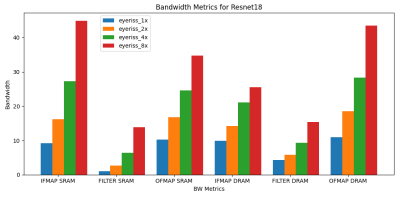

Figure 18: ResNet18

Figure 19, 20, 21 show the total cycles, compute utilization, and mapping efficiency for mobilenet, FasterRCNN, and ResNet18 respectively.

- As array size increases, total cycles, compute utilization and mapping efficiency decreases.
- Total cycles and mapping efficiency for mobilenet is lowest and for ResNet18 is highest.
- Compute utilization for mobilenet is lowest and for FasterRCNN is highest.

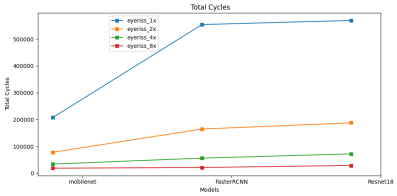

Figure 19: Total cycles

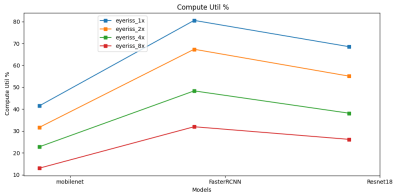

Figure 20: Compute Utilization

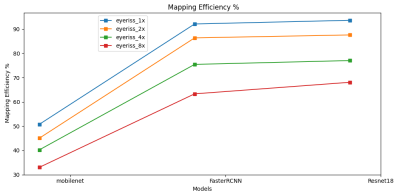

Figure 21: Mapping Efficiency