LOCAL: Low-Complex Mapping Algorithm for Spatial DNN Accelerators

Midia Reshadi, David Gregg

School of Computer Science and Statistics,

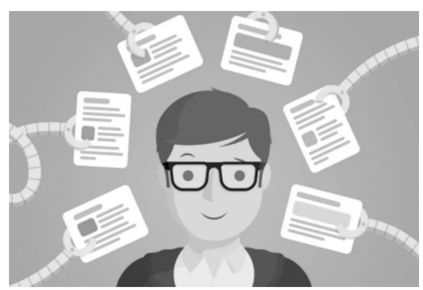
Lero, Trinity College Dublin

Outline

- Introduction
- Motivation
- Problem formulation
- LOCAL mapping algorithm
- Simulation results

Introduction

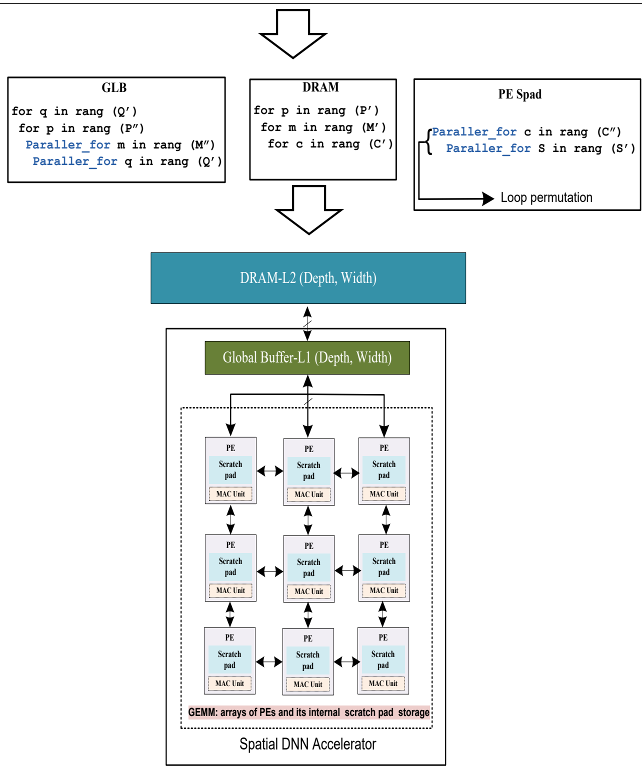
- DNN applications
  - Self-driving cars
  - Recommendation systems
  - Language translation

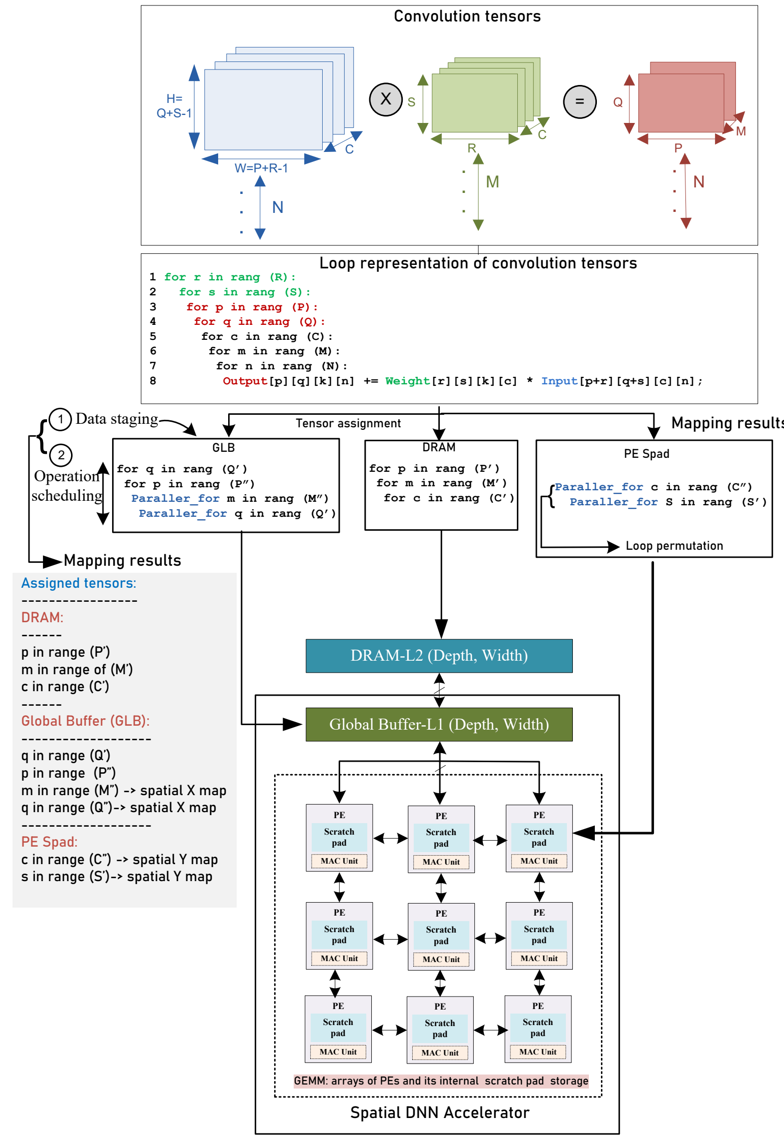


Mapping algorithm

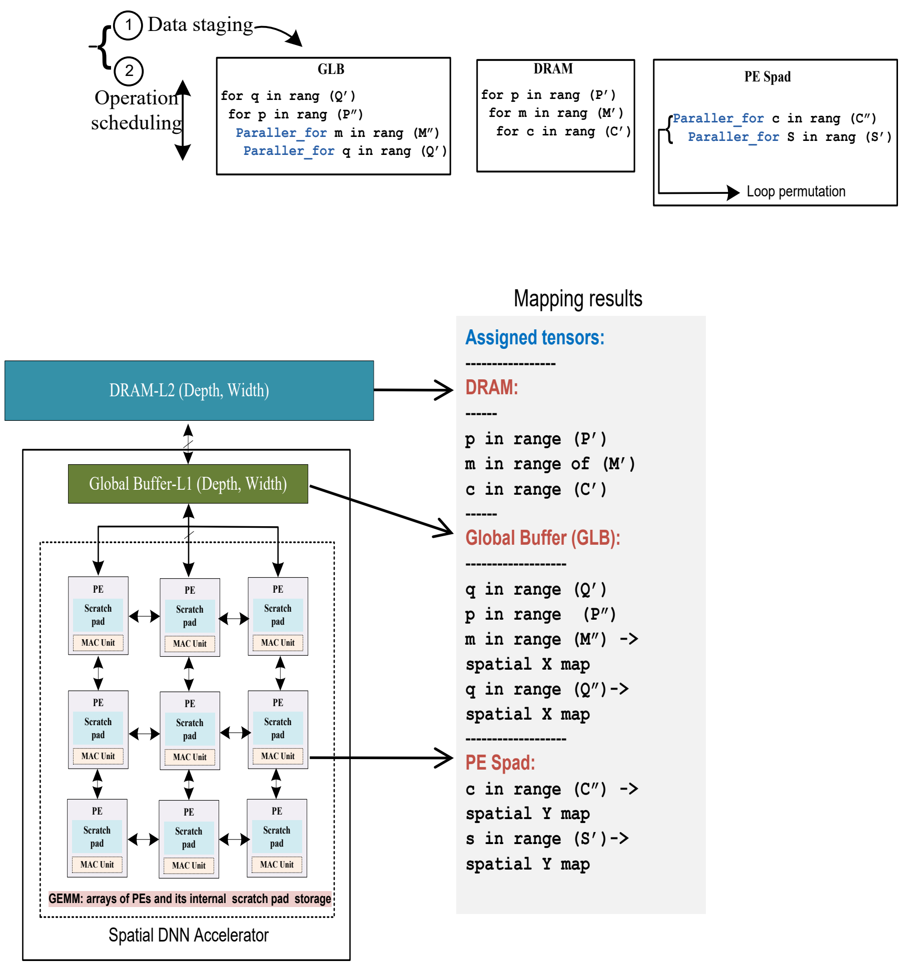
Loop representation of convolution tensors

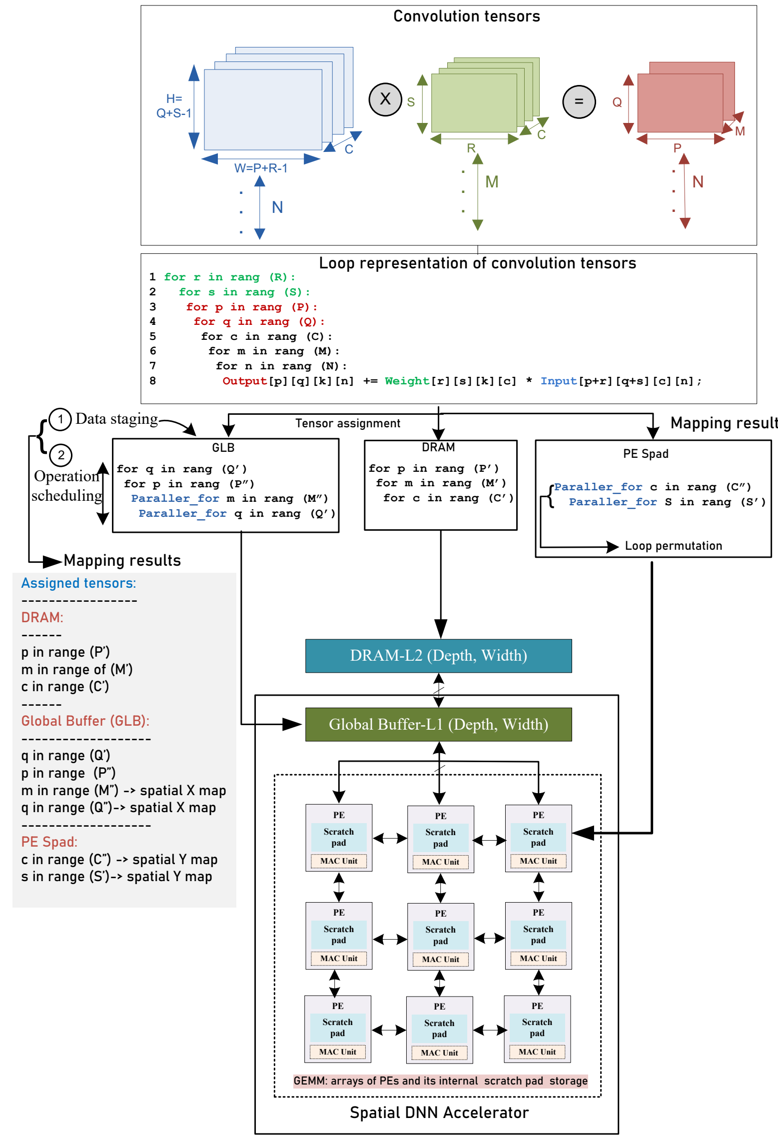
```
for r in rang (R):
  for s in rang (S):
   for p in rang (P):
    for q in rang (Q):
     for c in rang (C):
      for m in rang (M):
       for n in rang (N):
        Output[p][q][k][n] += Weight[r][s][k][c] * Input[p+r][q+s][c][n];
```





Mapping algorithm

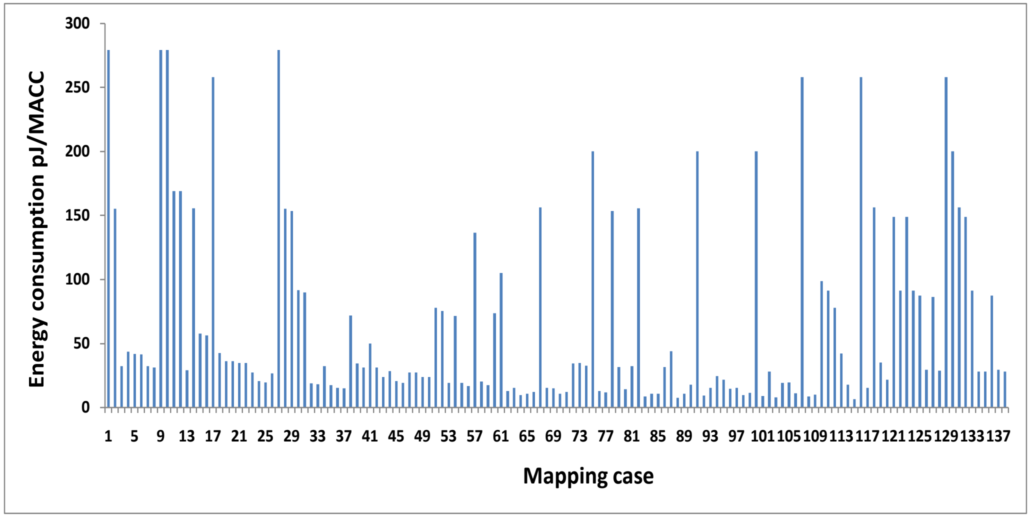




Motivation

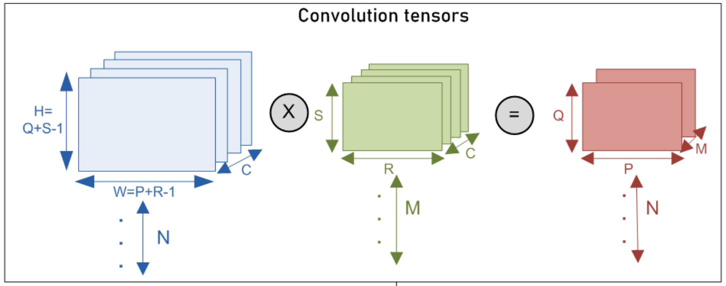
- Random mapping of Layer 5 of VGG16 to Eyeriss
  - 3,000 random mapping cases
  - 77% difference between the *Random_max* and the *Random_med*

SPA architecture	
SPA	Eyeriss
On-chip storage levels	2
DRAM(width)	64
L_1(depth,width)	(16384, 64)
L_0(depth,width)	(16,16)
PE array	(12,14)



Preliminaries

- Convolutions



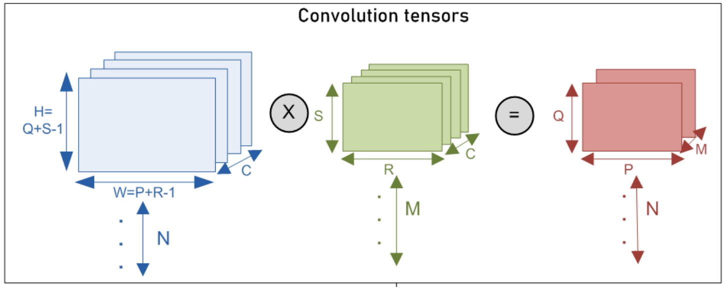
$CT = \{Input, Weight, Output\}$

$CT \in R^{dimension}$

$$dimension = \{N, M, C, W, H, R, S, P, Q\}$$

Preliminaries

- Convolutions



$$dimension = \{N, M, C, W, H, R, S, P, Q\}$$

$$W \in R^{MCRS} \qquad I \in R^{NCHW} \qquad O \in R^{NMPQ}$$

$$ct_i \in CT$$

$$CT = \{ct_1, ct_2, ct_3\}$$

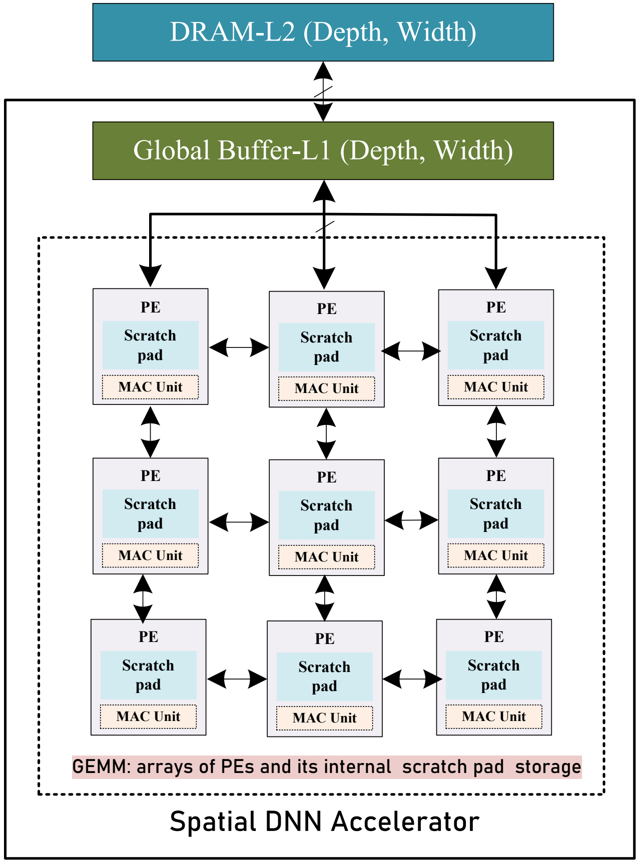
$$CT = \{R^{MCRS}, R^{NCHW}, R^{NMPQ}\}$$

Preliminaries

- Spatial DNN accelerator

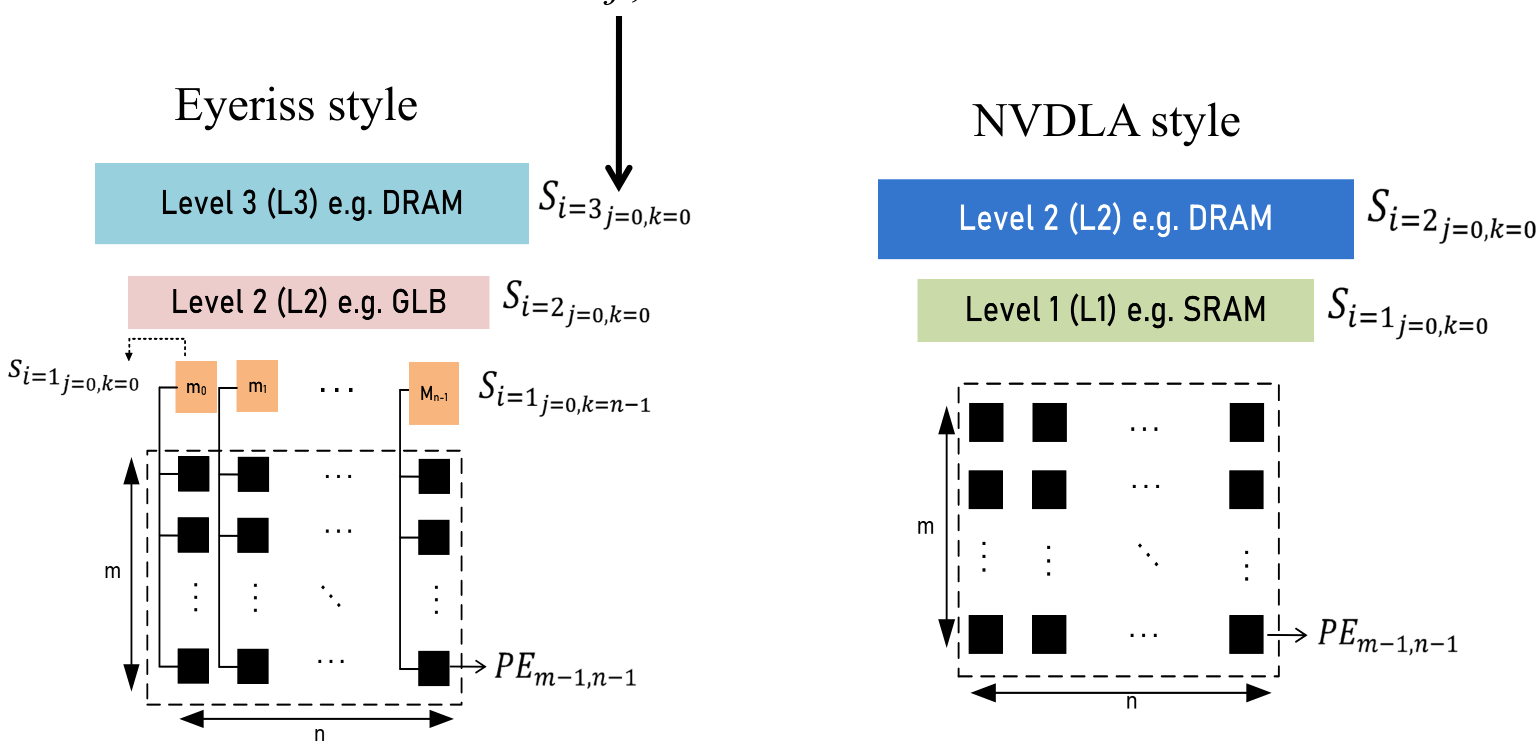
$$SPA = \{Storage[i, j, k], PE[m, n]\}$$

$$PE_{m,n} \in R^{x,y}$$



Preliminaries

$$S_{i\,j,k} \Rightarrow S \in R^{i,j,k}$$



Mapping algorithm

- Assignment $ct_i \in CT \ \mathit{assign\ to}\ s_{i_{j,k}} \in S$
- Bounding

$ct_i\ [0, rang) \in CT\ \ \mathit{assign\ to}\ s_{i_{j,k}} \in S$

$\mathit{for}\ Q\ \mathit{in}\ [0,5)\ \mathit{assigned\ to}\ s_{i=1_{j=0,k=0}}$ EXAMPLE

- Scheduling

$ct_i, ct_j, ct_k \in CT\ \ \mathit{assign\ to}\ L_i$

- Parallelization

$$\text{Order} \left\{ \begin{array}{l} ct_i\ [0, rang_i) \\ \quad ct_j\ [0, rang_j) \\ \qquad ct_k\ [0, rang_k) \end{array} \right.$$

Spatial computing $\rightarrow\ ct_i\ [0, rang) \in CT\ \mathit{on}\ PE_{i\ \mathit{to}\ j\ (x|y)} \in PEs$

Parallel_for S in [0,7) on $PE_{0\text{-}7}$ *Spatial X dimension*

Mapping Problem Formulation

- ***Given:***
  - Convolution tensors *CT*
  - Spatial DNN accelerator *(SPA)*
- ***Find***
  - A mapping function *map( )*

$$min\{Energy\}$$

$$max\{PE\ utilization\}$$

$$utilization\ of\ PEs = \frac{number\ of\ active\ PEs}{number\ of\ PEs}$$

Local Mapping Algorithm

Algorithm1: Tensor to memory assignments $ct_i \ to \ s_{i_{(j,k)}}$

Inputs: All $ct_i \in CT$ and all $s_{i_{(j,k)}} \in S$
Size of ($s_{i_{j,k}}$)
PE dimension ($PE_{m,n}$)

Outputs: $ct_i \rightarrow s_{i_{(j,k)}}$ (Assignment & Permutation)
$ct_i \rightarrow PE_{i \ to \ j \ (x|y)}$ (Parallelization)

Parallelization:
$\forall s_{i=1_{j,k}}$:
if (k==0) //NVDLA style
assign: parallel_for C in $Rang(m)$ spatial x dimension to $s_{i=1}$
assign: parallel_for M in $Rang(n)$ spatial y dimension to $s_{i=1}$
else //Eyeriss Style
assign: parallel_for in Q in $Rang(m)$ spatial x dimension to $s_{i=1}$
assign: parallel_for in S in $Rang(n)$ spatial y dimension to to $s_{i=1}$
end_if
Assignment:
$\forall$ all Unassigned tensors: U_{ct_i}
for all remaining U_{ct_i} :
for (i=0 $\rightarrow$ i$\leq$n)
Assign U_{ct_i} to s_i // Assigning with priority from $s_{i=0}$ to $s_{i=n}$
end_for
end_for
Scheduling:
for all Assigned tensors: A_{ct_i}
sort high to low range ct_i
$\forall s_i$ for (i=0 $\rightarrow$ i$\leq$max memory level)
do permutation to allocate higher range tensor to lower s_i
end_for

More details

***Parallelization*:**

$\forall s_{i=1_{j,k}}$:

if (k==0) //NVDLA style

assign: parallel_for C in $Rang(m)$ spatial x dimension to $s_{i=1}$

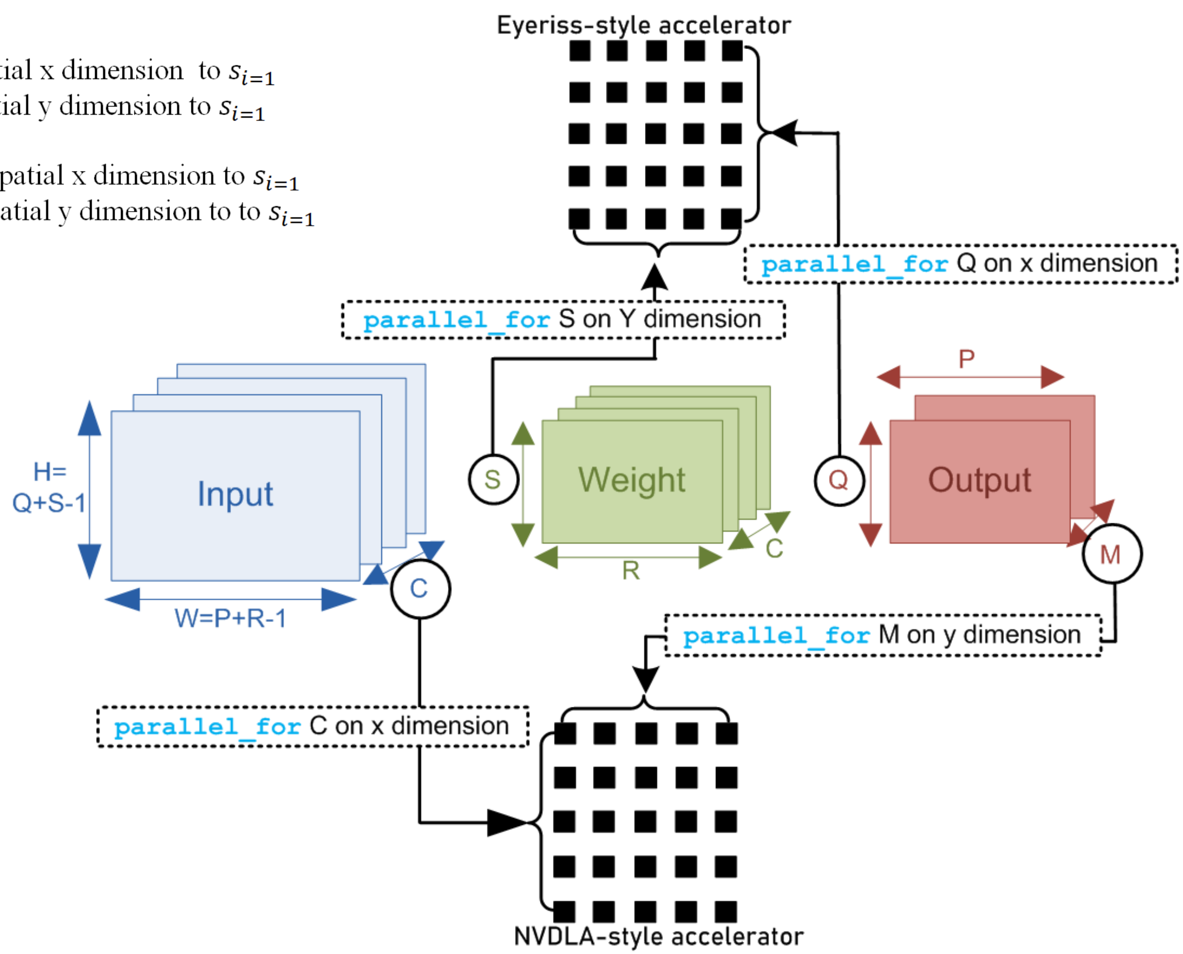
assign: parallel_for M in $Rang(n)$ spatial y dimension to $s_{i=1}$

else //Eyeriss Style

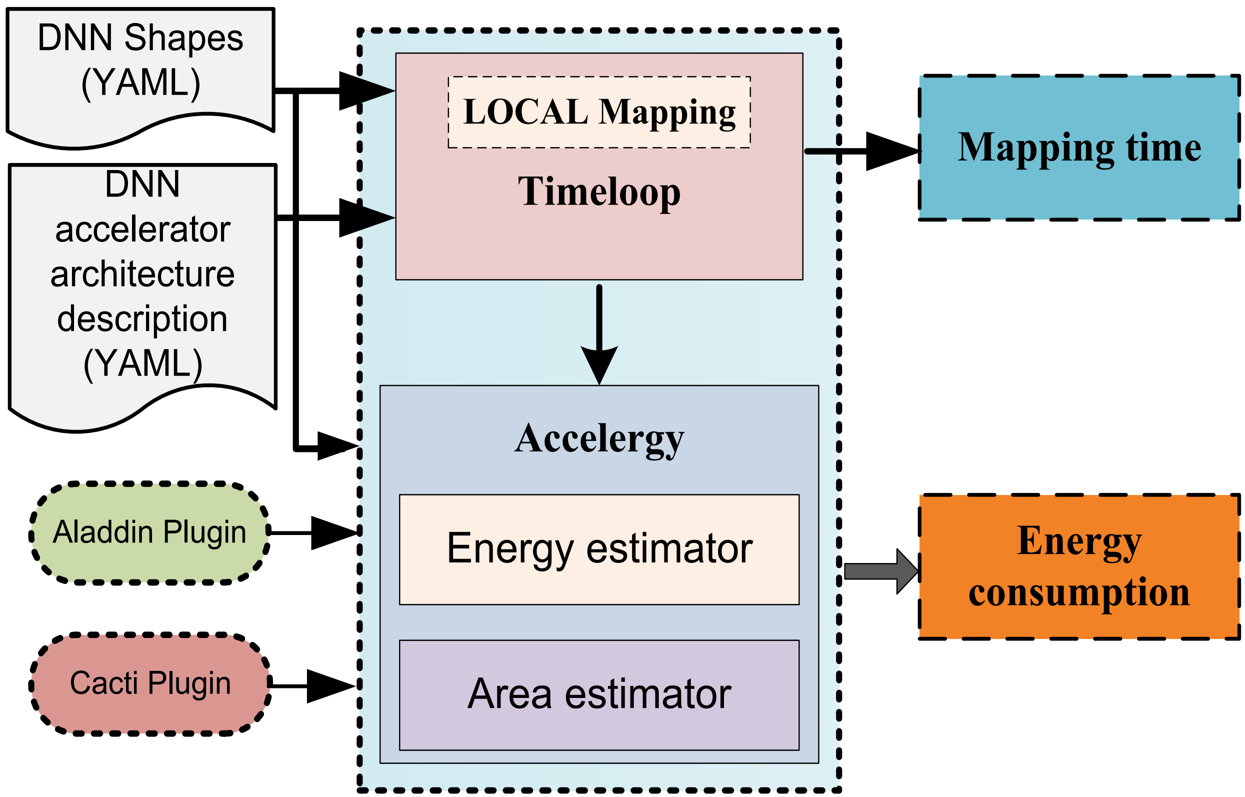
assign: parallel_for in Q in $Rang(m)$ spatial x dimension to $s_{i=1}$

assign: parallel_for in S in $Rang(n)$ spatial y dimension to to $s_{i=1}$

end_if



Evaluation


DNN Shapes (YAML)
DNN accelerator architecture description (YAML)
Aladdin Plugin
Cacti Plugin
LOCAL Mapping
Timeloop
Accelergy
Energy estimator
Area estimator
Mapping time
Energy consumption

Simulation Workload

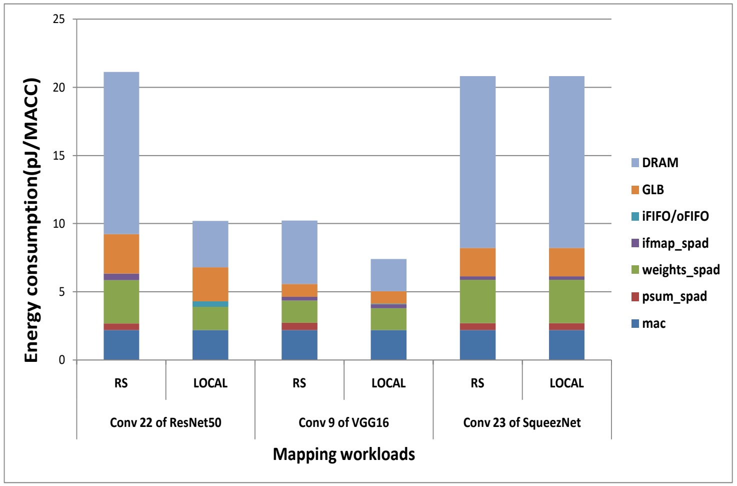
Table 2. Workload categories

Category	Workload	Number of MAC operation
High C value	22nd conv layer of Resnet50	51380224
	23nd conv layer of SqueezNet	5537792
	9nd conv layer of VGG16	1849688064
High M value	25nd conv layer of SqueezNet	24920064
	24nd conv layer of ResNet50	51380224
	8nd conv layer of VGG16	924844032
High P and Q values	1st conv layer of SqueezNet	708083712
	1st conv layer of ResNet50	472055808
	1st conv layer of VGG16	86704128

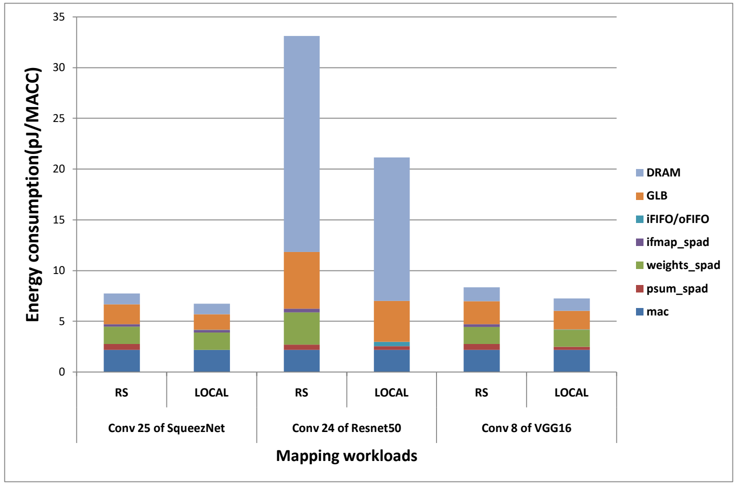
Mapping time

Workload	***Convolution***	***Mapping mechanism based on Eyeriss***	***Mapping time (sec)***	***Mapping mechanism based on ShiDianNao***	***Mapping time (sec)***	***Mapping mechanism based on NVDLA***	***Mapping time (sec)***
High C value	Resnet50: Conv 22	RS	87	OS	**576**	WS	127
		LOCAL	16.2	LOCAL	**15**	LOCAL	6
	VGG16: Conv 9	RS	170	OS	137	WS	68
		LOCAL	10	LOCAL	15	LOCAL	9
	SqueezNet: Conv 23	RS	17	OS	125	WS	21
		LOCAL	16	LOCAL	67	LOCAL	18
High M value	SqueezNet: Conv 25	RS	**230**	OS	126	WS	996
		LOCAL	**6.6**	LOCAL	16	LOCAL	31
	Resnet50: Conv 24	RS	74	OS	116	WS	42
		LOCAL	22	LOCAL	28	LOCAL	12
	VGG16: Conv 8	RS	351	OS	98	WS	411
		LOCAL	12	LOCAL	32	LOCAL	24
High P and Q values	SqueezNet: Conv1	RS	60	OS	20	WS	**2238**
		LOCAL	5.1	LOCAL	7	LOCAL	**45**
	Resnet50: Conv 1	RS	90	OS	60	WS	140
		LOCAL	6	LOCAL	13	LOCAL	23
	VGG16: Conv 1	RS	81	OS	24	WS	113
		LOCAL	6.6	LOCAL	6	LOCAL	17

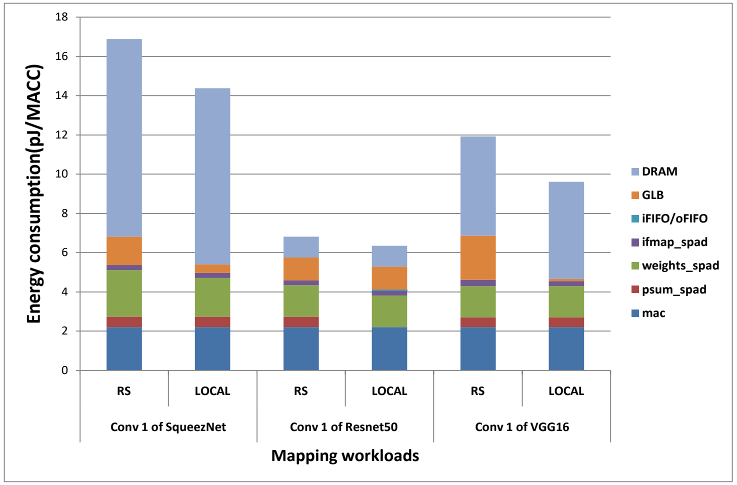
Simulation Results



Energy consumption of row stationary and LOCAL mapping in Eyeriss with High C value workload

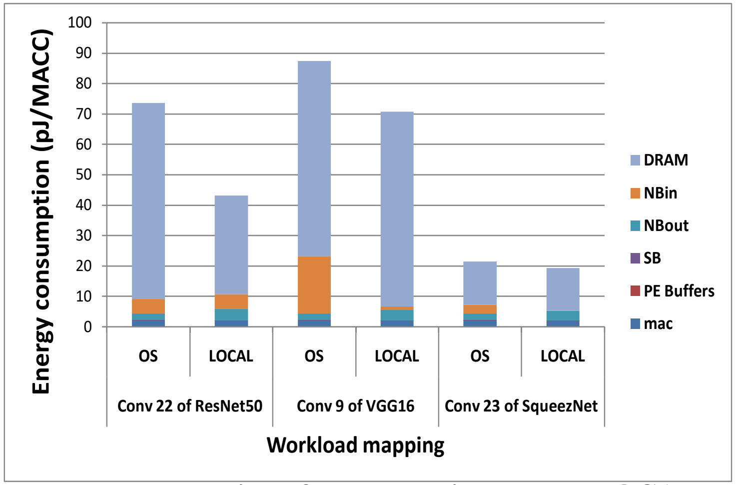


Energy consumption of row stationary and LOCAL mapping in Eyeriss with High M value workload

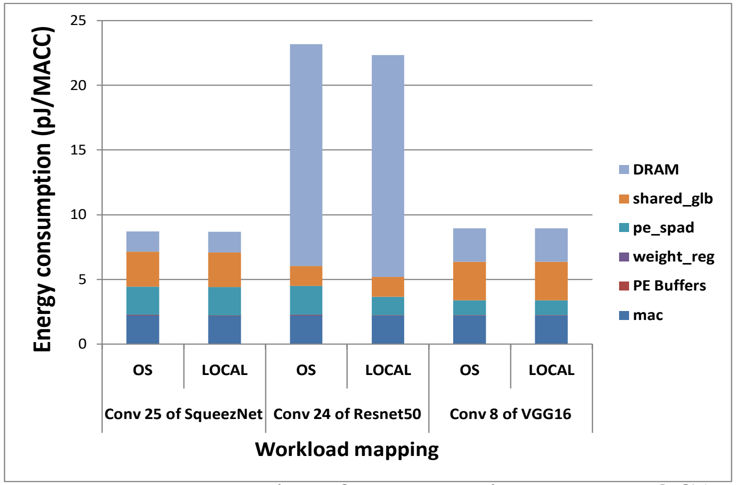


Energy consumption of row stationary and LOCAL mapping in Eyeriss with High P and Q values workload

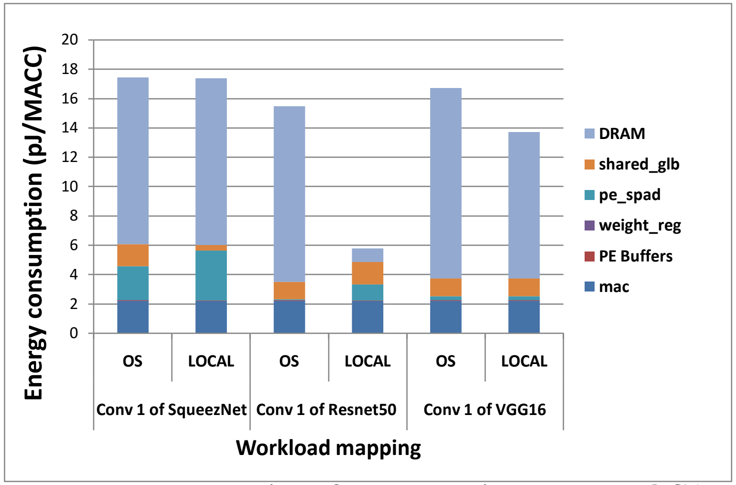
Simulation Results



Energy consumption of output stationary and LOCAL mapping in Shi-diannao with High C value workload

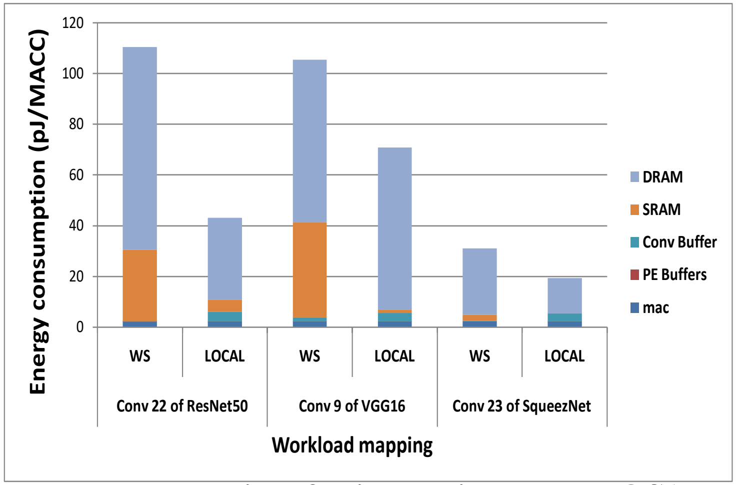


Energy consumption of output stationary and LOCAL mapping in Shi-diannao with High M value workload

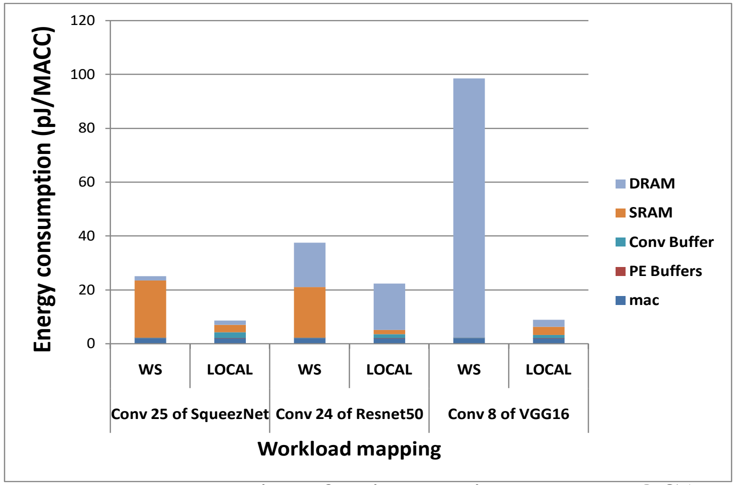


Energy consumption of output stationary and LOCAL mapping in Shi-diannao with High P and Q values workload

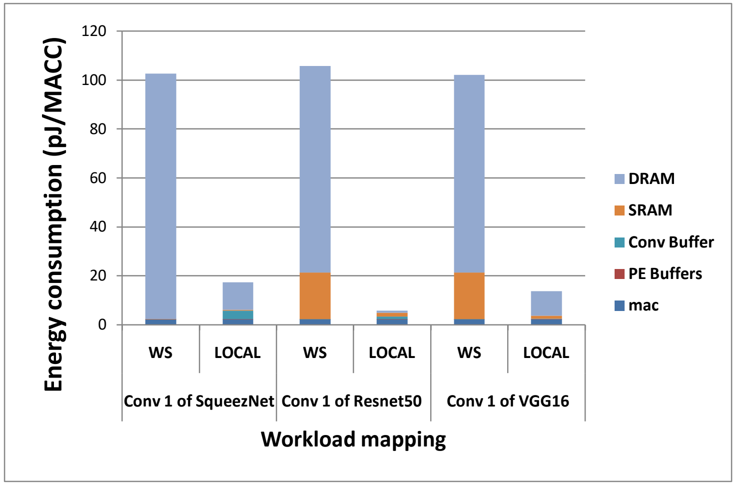
Simulation Results



Energy consumption of weight stationary and LOCAL mapping in NVDLA with High C values workload



Energy consumption of weight stationary and LOCAL mapping in NVDLA with High M values workload



Energy consumption of weight stationary and LOCAL mapping in NVDLA with High P and Q values workload

Thank you for your attention