ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

Optimizing Data Layout for Racetrack Memory in Embedded Systems

Peng Hui, Edwin H.-M. Sha, Qingfeng Zhuge, Rui Xu, Han Wang
School of Computer Science and Technology, East China Normal University

Outline

- **Background**
- **Motivations**
- **Techniques**
- **Evaluation**
- **Conclusion**

Background

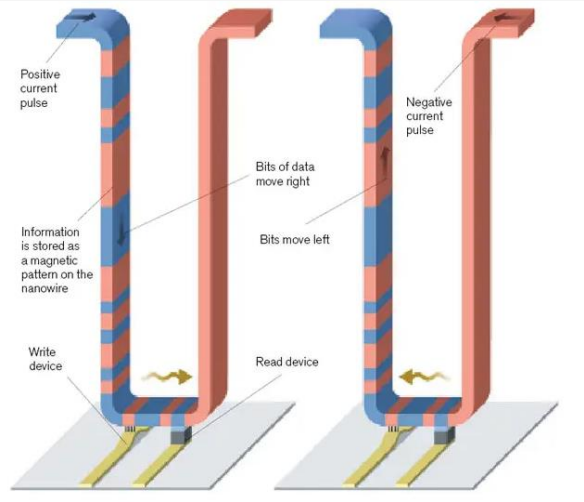
Racetrack Memory(RTM)

Advantages:

Nonvolatile

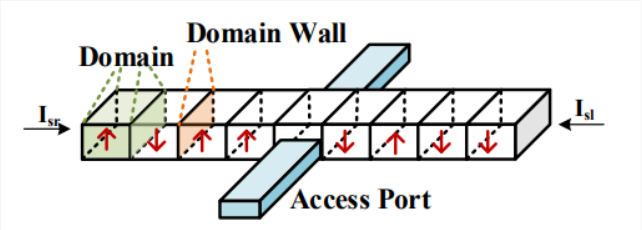
Low latency

High density



Disadvantages:

The data to be accessed needs to be aligned with the access port, otherwise it need to shift data.



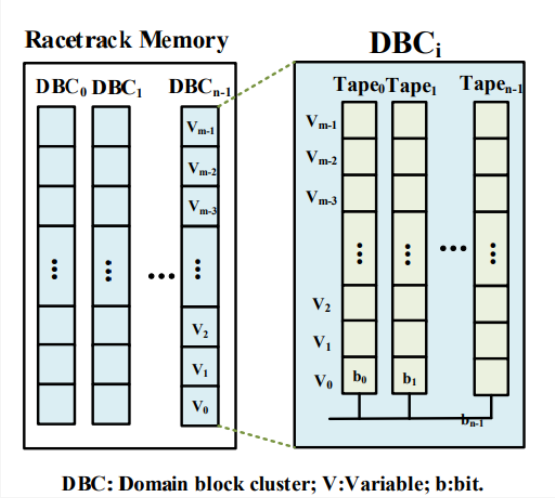
Random data access generates a lot of shifts.

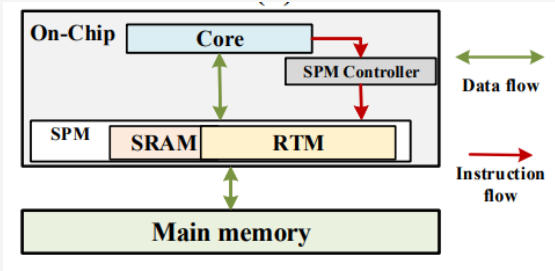
Background

RTM has multiple domain block clusters (DBCs).

Each DBC has multiple tapes.

Using SRAM to reduce shifts.

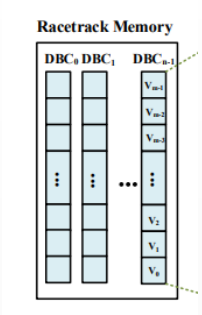




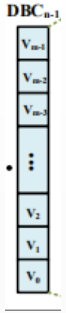
Objective

Reducing shifts (and used DBCs).

Grouping (inter DBCs) → **Placement (intra DBC)** → **Allocation (Hybrid Memory)**



Which DBC should the data be placed?

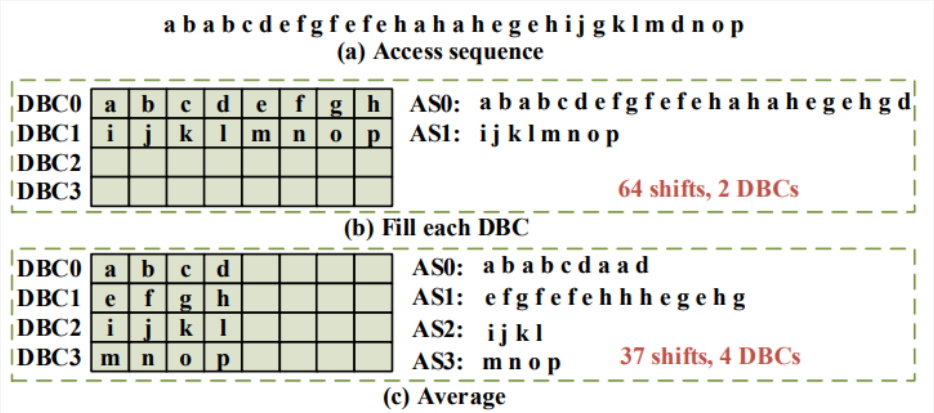


Where the data should be placed in the DBC？

Which data should be placed on SRAM?

Motivation 1: Grouping

It is a good idea to group data equally across the DBC.



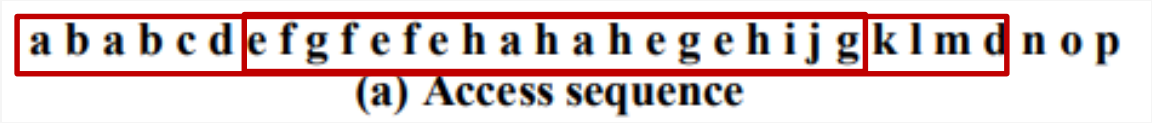
$$average\_shifts = \frac{all\_shifts}{\#DBC}$$

Difficult to reduce in grouping stage

We can increase the number of DBC.

Technique 1

Segment the DBC to increase the number of DBC indirectly.
Select data with **non-overlapping life span** between segments



(a) Access sequence

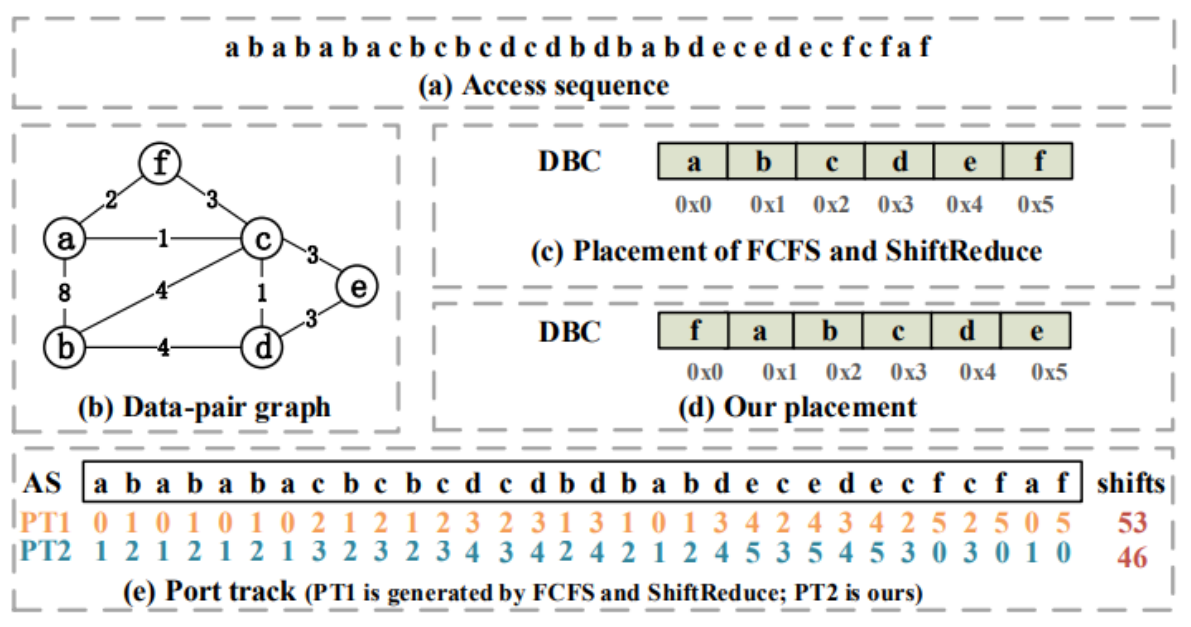
Sorted by first appearance: ~~a b c d~~ e f g h l j k l m ~~n o p~~

DBC0	a	b	c	d	n	o	p	
DBC1	e	f	g	k	l	m		
DBC2	h	i	j					
DBC3								

31 shifts, 3DBCs(Average grouping: 37 shifts, 4DBCs)

Motivation 2: Placement

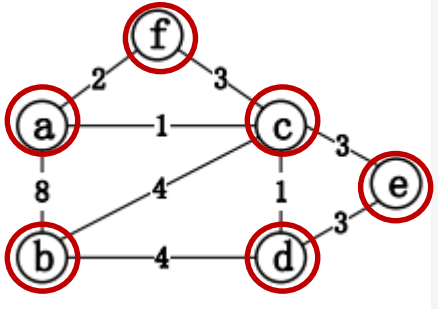
Data placement algorithm is important to reduce shifts on RTM.



Technique 2

The data should be placed next to the data with which it is most continuously accessed.

a b a b a b a c b c b c d c d b d b a b d e c e d e c f c f a f



Placed data

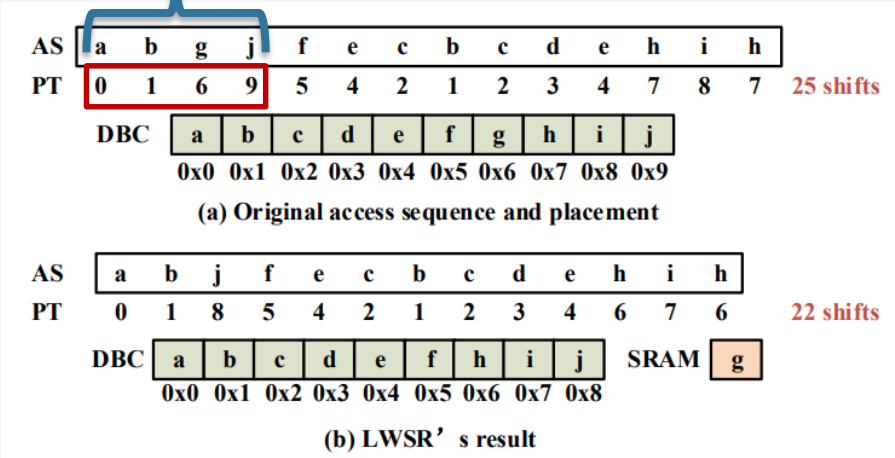
f a b c d e

Divide the placed data from middle, then places data based on the weight with the left group and right group.

Motivation 3: Allocation

It may not get the earning corresponding to the cost of data after moving it to SRAM.

The port track is monotonous.

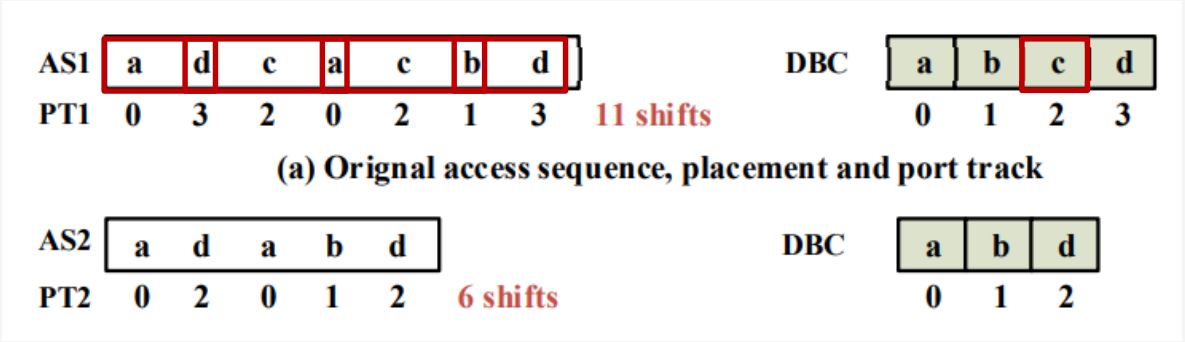


Cost(g) = 5

Earning(g) = 25-22=3

Technique 3

$$cost_v = latency_{v\_on\_RTM} - latency_{v\_not\_on\_RTM}$$



Part1: three variable sub-sequences containing *c*: {*d, c, a*}: 1; {*a, c, b*}: 3-1=2; $|Pos_a - Pos_c| + |Pos_c - Pos_b| = 3$;
Part2: two variable sub-sequence crossing *c*: {*a, d*}: 1; {*b, d*}: 1; $|Pos_a - Pos_b| = 1$

Make the fluctuation of port track smoother to reduce shifts.

Evaluation

Simulator: an in house simulator

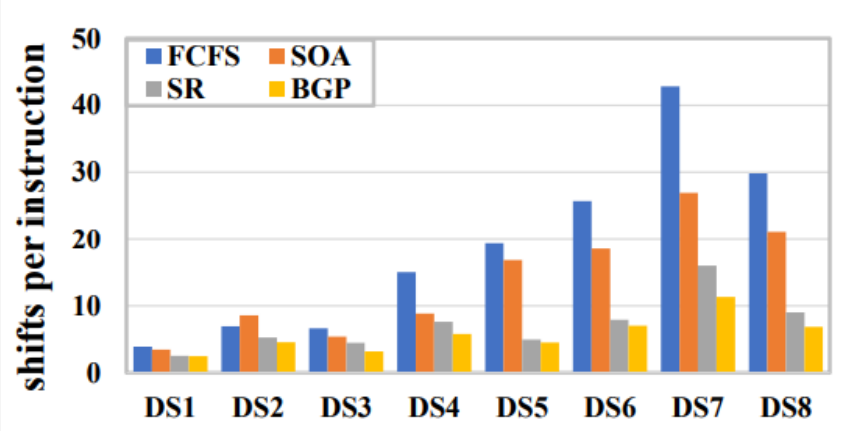
RTM: 32-DBCs, one DBC can store 512 variables

SRAM: can store 256 variables

Workloads: 8 traces from Mibence

Evaluation

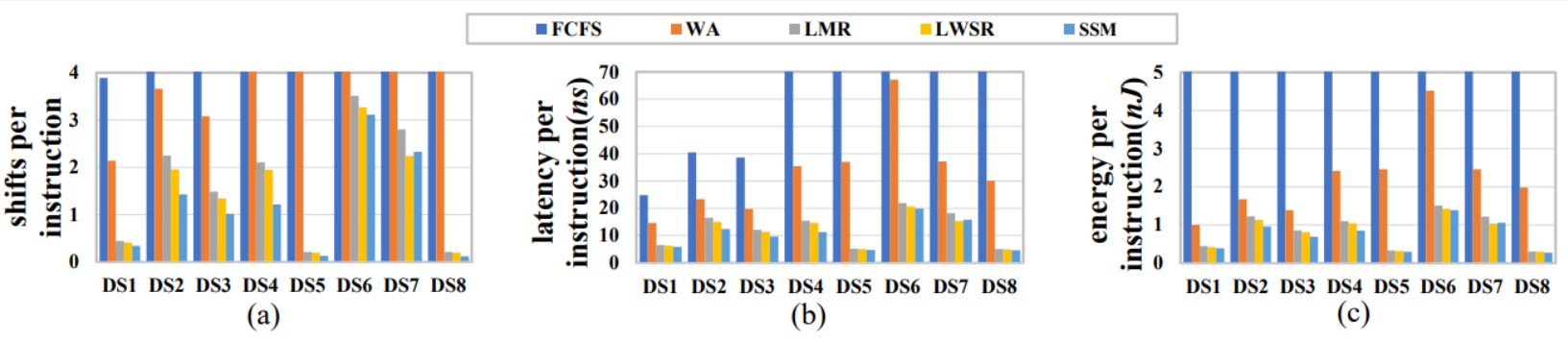
Shifts generated by different **placement** methods.



FCFS: First Come First Store
SOA: a heuristic algorithm for Single Offset Assignment problem in DSP stack's frame
SR: a group based algorithm which exploits the locality of accesses in the access sequence and assigns offset accordingly
BGP: ours

Evaluation

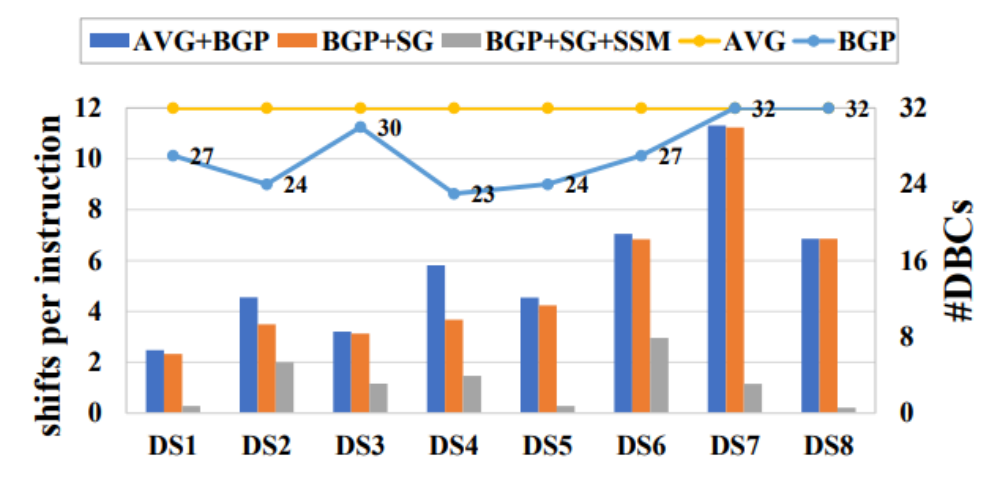
Shifts, latency and energy generated by different **allocation methods under hybrid SPM**



FCFS: First Come First Store
WA: moves the data with most write times to SRAM
LMR: most write times to SRAM
LWSR: selects the variables with the maximum sum of the number of shifts and writes each time and puts it on SRAM
SSM: ours

Evaluation

Shifts generated by combining our methods, and the amount of DBC required by different **grouping** methods.



AVG+BGP: average grouping first, then place data using BGP inside DBC
GBP+SG: grouping data using SG first, then place data using BGP inside DBC
BGP+SG+SSM: grouping data using SG first, then place data using BGP inside DBC, and move some data to SRAM using SSM.

Conclusion

Proposed:

For pure RTM:

A placement method to placement data close to its most continuously accessed.

A grouping strategy to reducing shifts and used DBCs by using one DBC as multiple DBCs.

For hybrid SPM:

A cost evaluation metric is proposed to determine which variables should be placed on SRAM, and an allocation method based on the metric is proposed to reduce shifts.

Experiments show that BGP reduces shifts by 17.64% compared with SR on average; SSM is 22.12% better than LWSR. In addition, SG+BGP reduces shifts by up to 36.87% while only using 23 DBCs (32 DBCs in default) compared with the average grouping method paired with BGP.

ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

Optimizing Data Layout for Racetrack Memory in Embedded Systems

Thank you!

Questions?

If any questions, please contact us!

Qingfeng Zhuge *qfzhuge@cs.ecnu.edu.cn*
Peng Hui *penghui@stu.ecnu.edu.cn*