DDR Memory Interference Optimization on Heterogeneous Multicore Platforms

CONCORDE PROJECT

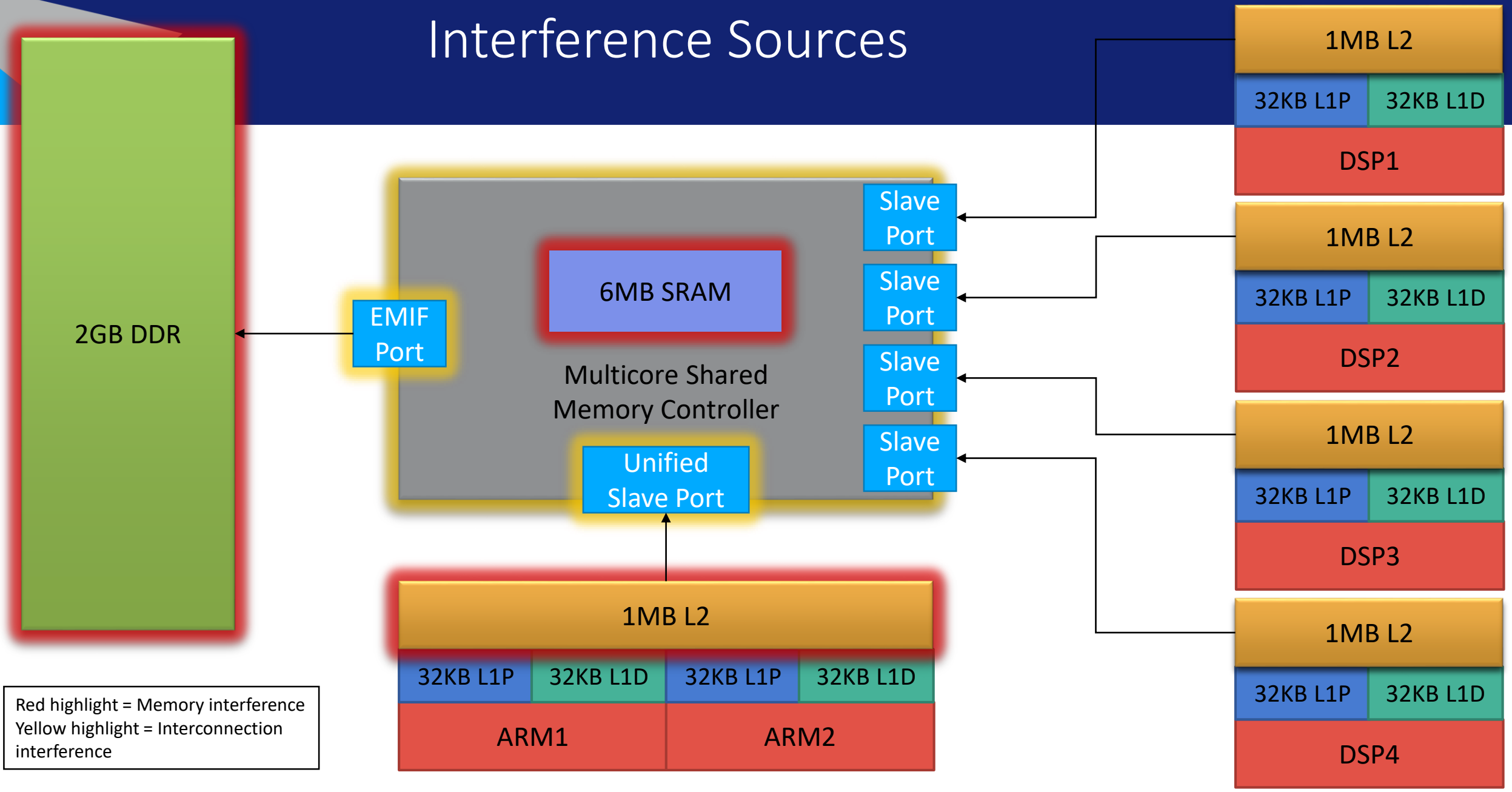
Alfonso Mascareñas González

17/02/2022

ONERA
THE FRENCH AEROSPACE LAB

isae
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

Interference Sources

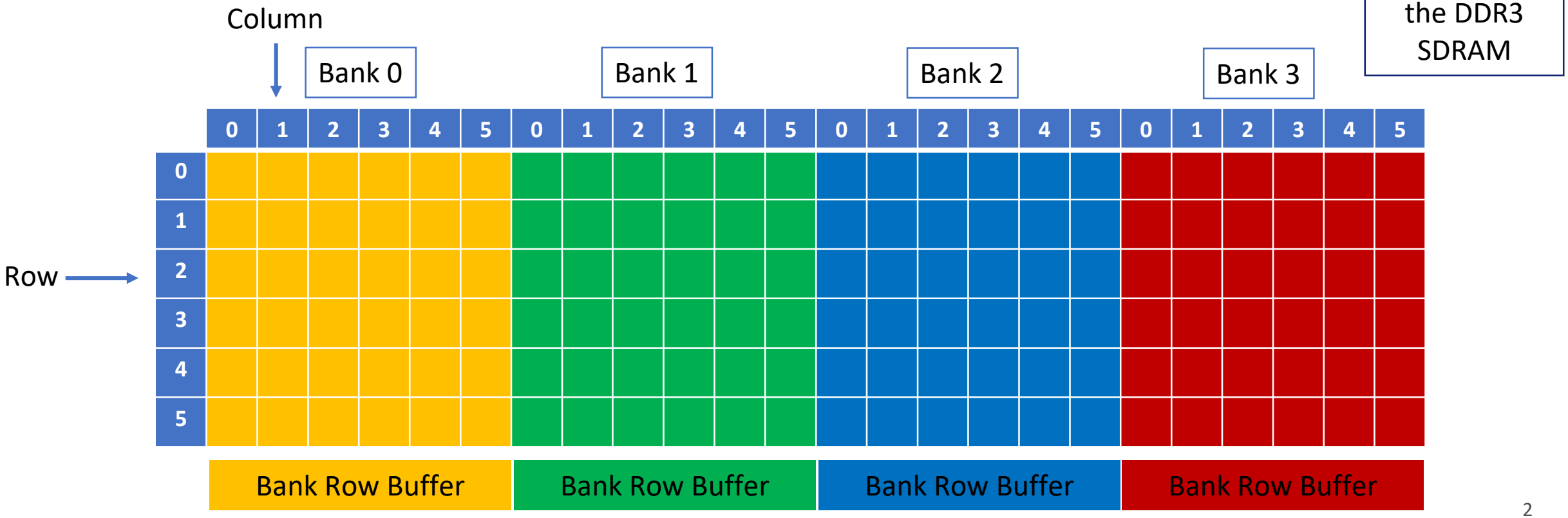


Red highlight = Memory interference
Yellow highlight = Interconnection interference

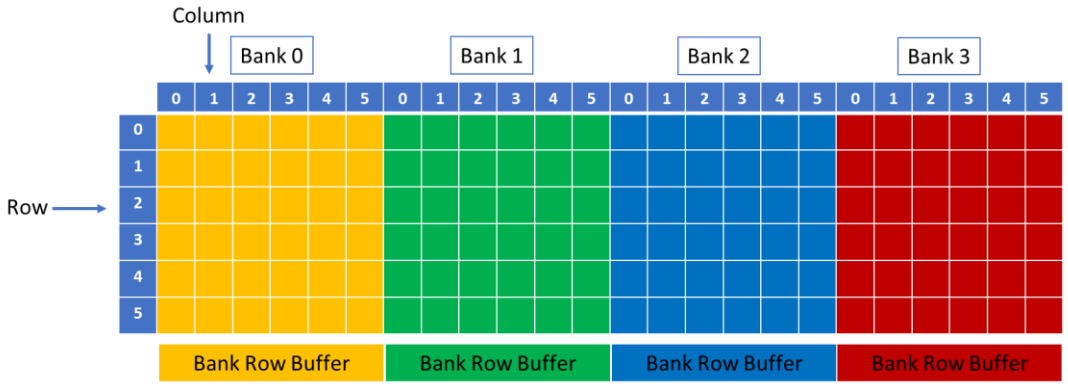
DDR3 SDRAM Device: DDR3 Addressing

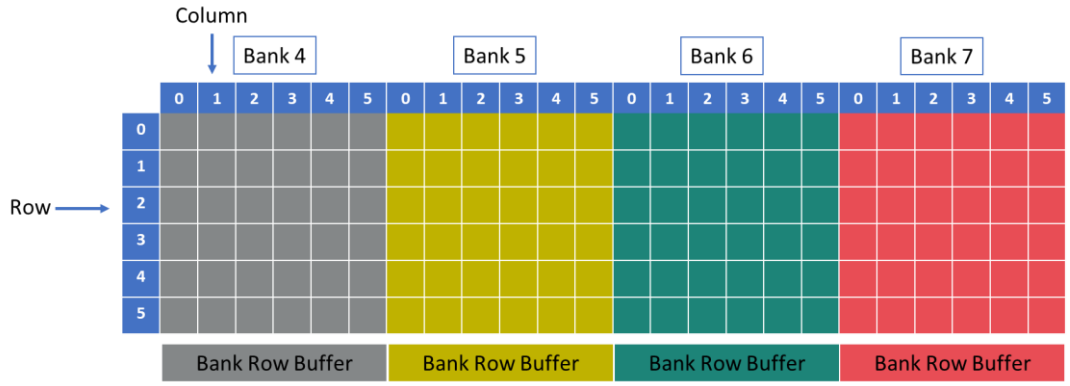
- Minimum division: Columns, rows, banks and ranks.
- Banks can treat commands independently from each other

8 banks for the DDR3 SDRAM

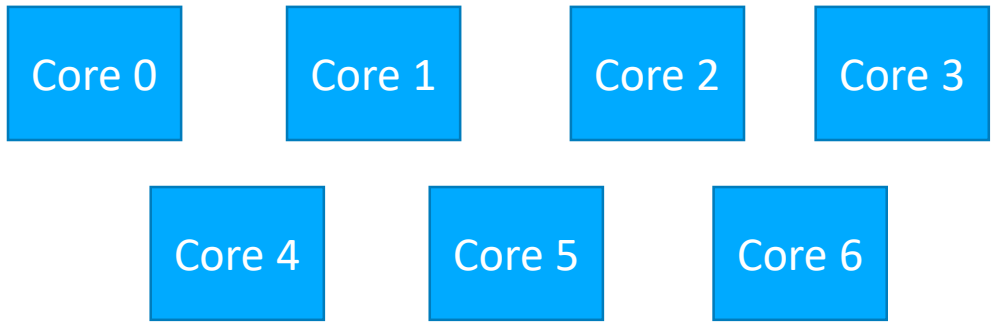


Task and Memory Mapping





Core-Bank Map



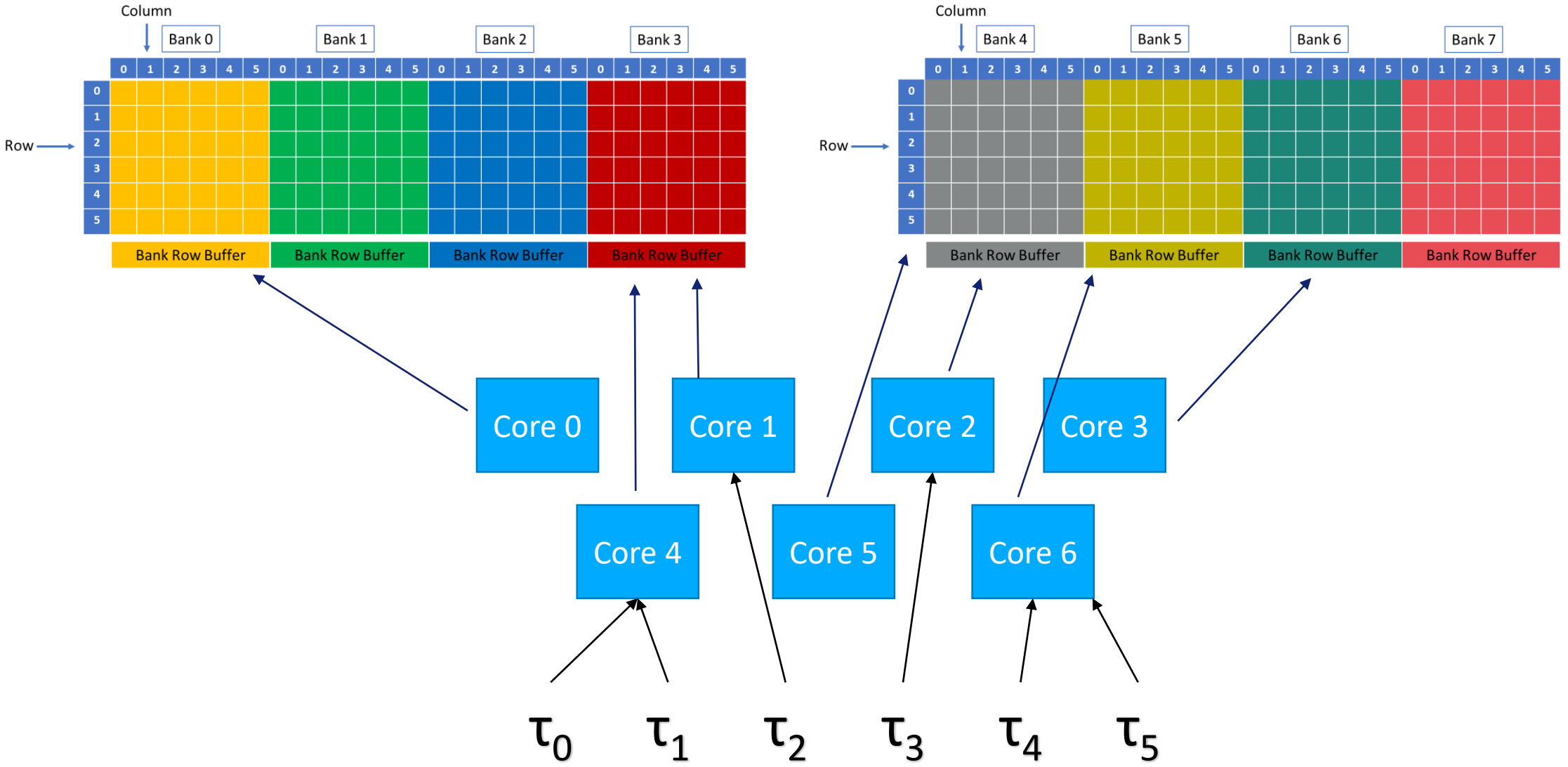
Task-Core Map

$C = core$
τ = task

τ_0 τ_1 τ_2 τ_3 τ_4 τ_5

Task Set

Task and Memory Mapping

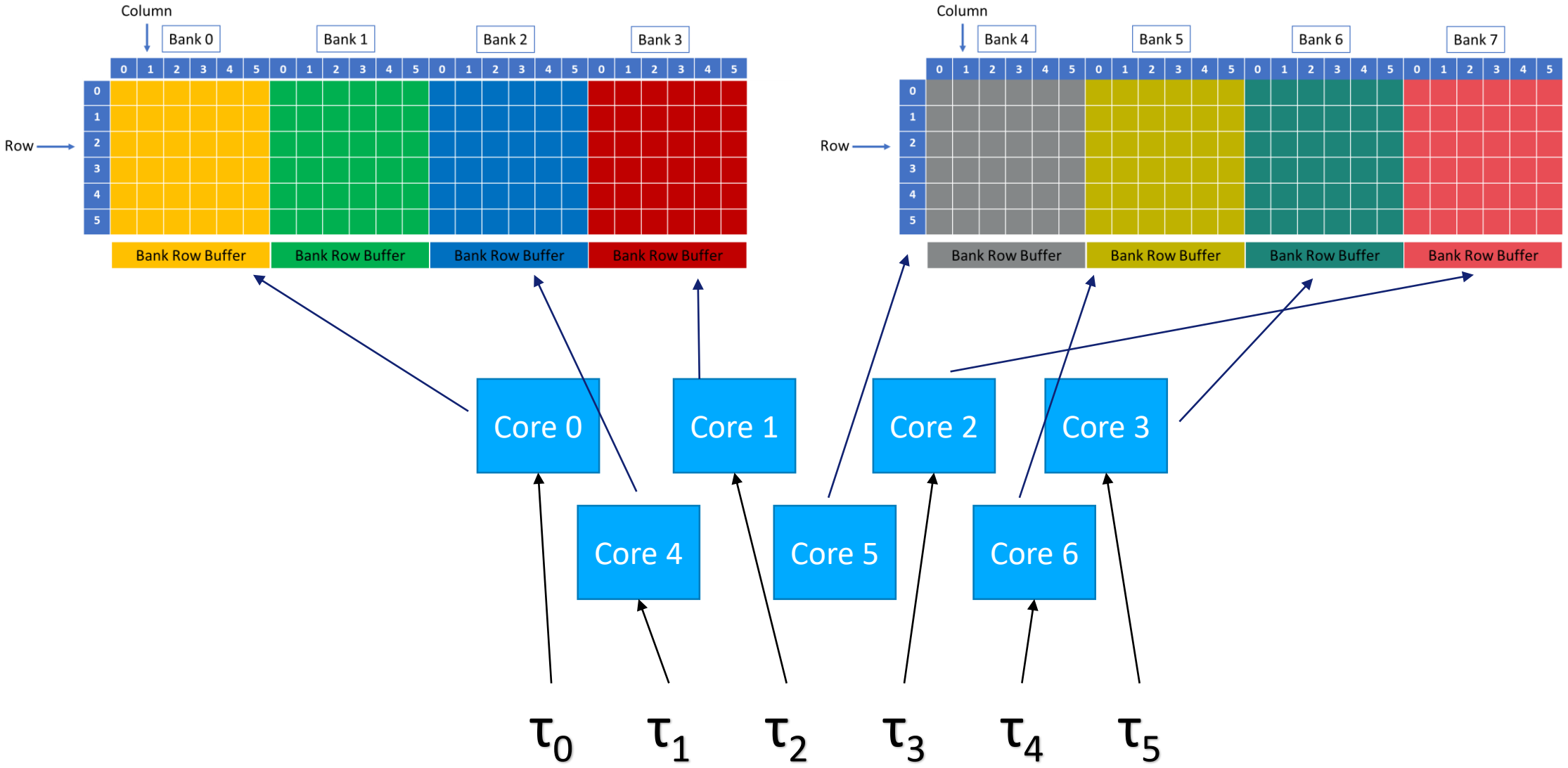


Core-Bank Map

Task-Core Map

Task Set

Task and Memory Mapping

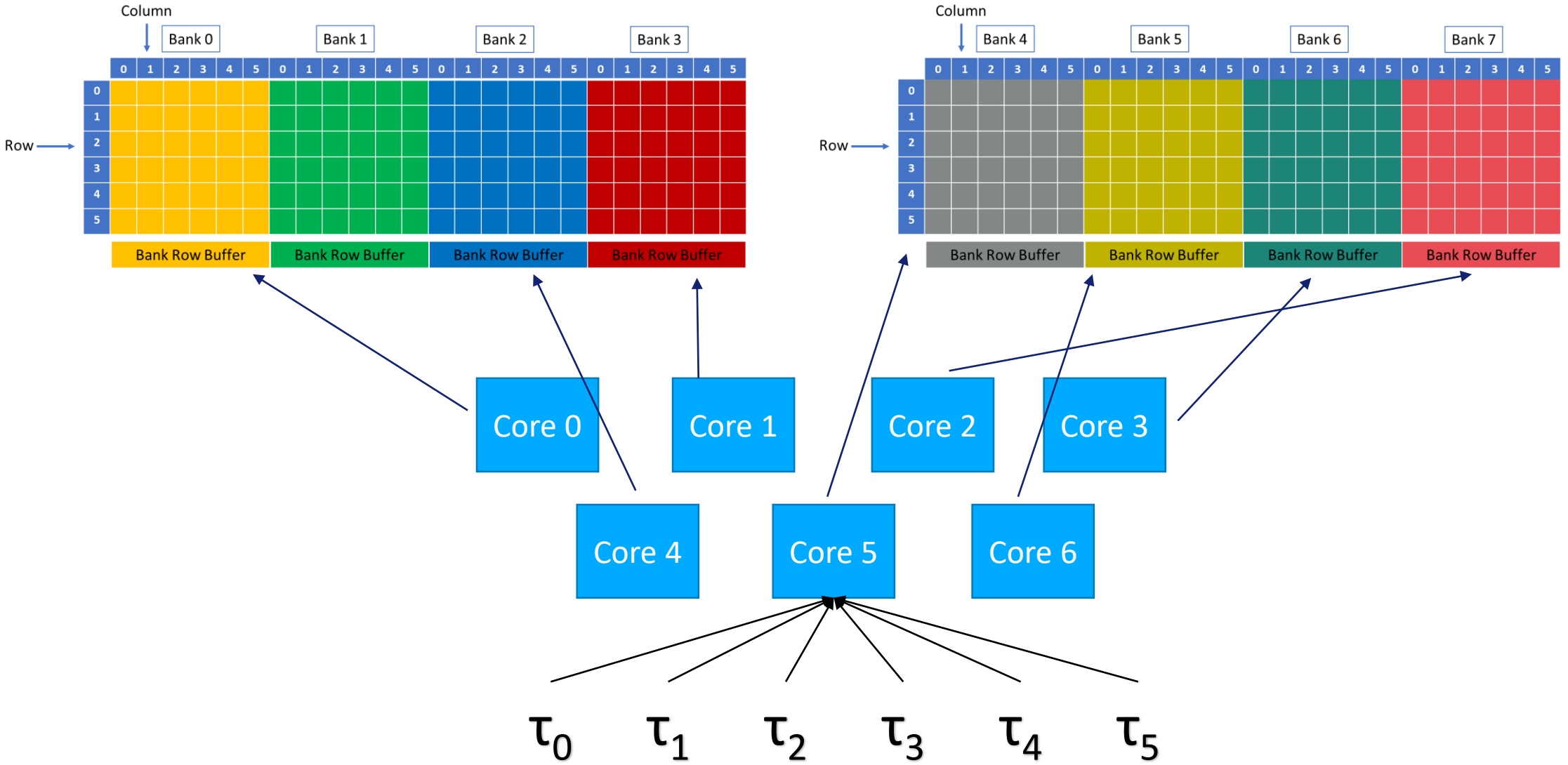


Core-Bank Map

Task-Core Map

Task Set

Task and Memory Mapping



Core-Bank Map

Task-Core Map

Task Set

Task and Memory Mapping Representation

The previous kind of maps can be represented as follows:

Bank	7	0	5	4	4	1	2	1	3	6	6	7
Core	0	7	4	5	5	1	3	1	2	6	6	0
Task	0	1	2	3	4	5	6	7	8	9	10	11

Alternatively, in one single row:

Core Bank

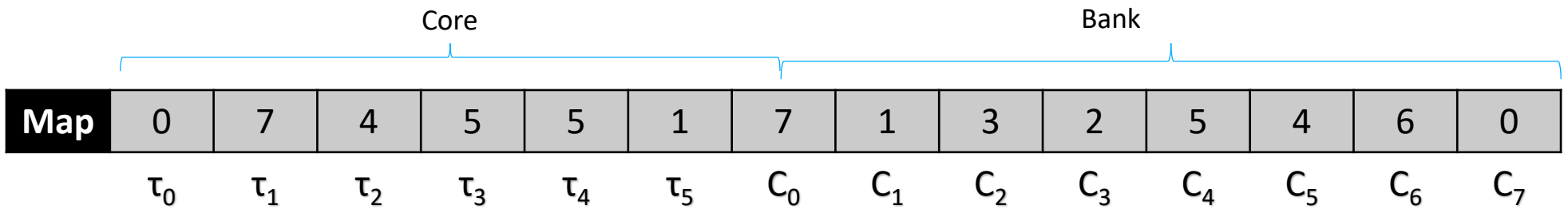
Map	0	7	4	5	5	1	7	0	5	4	4	1
	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5

Only 6 tasks are shown

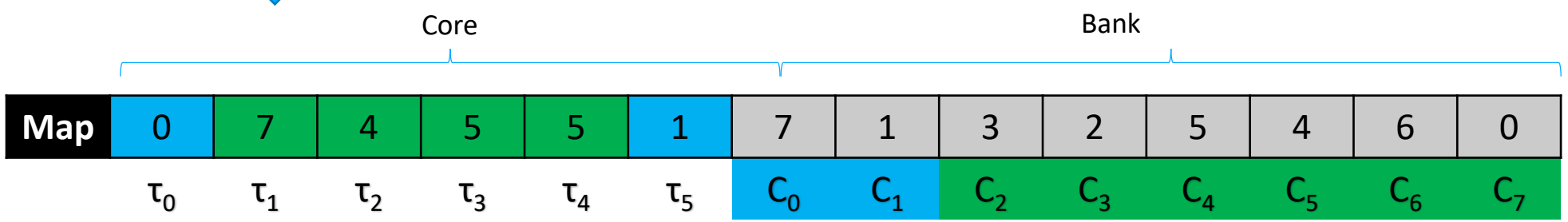
Task and Memory Mapping Representation

$C = core$
τ = task

Or even:



In heterogeneous platforms



Blue = ARM core
Green = DSP

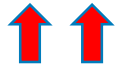
Multi-objective Optimization

How do we map tasks to cores and cores to banks? Is there a logic?

Objectives:

- Increase Task Execution Parallelism → $Minimize(Workload\ Variance)$
- Decrease Maximum DDR Interference → $Minimize(Maximum\ Interference\ Cost)$

$Minimize(Workload\ Variance)$ ⇒ **Maximize** $(Maximum\ Interference\ Cost)$
$Minimize(Maximum\ Interference\ Cost)$ ⇒ **Maximize** $(Workload\ Variance)$



Multi-objective Optimization

$WCET_i$ = Worst Case Execution Time of τ_i in isolation
IC(τ_i) = DDR memory interference cost for τ_i
PE = Processing element = Core

- Cost functions:

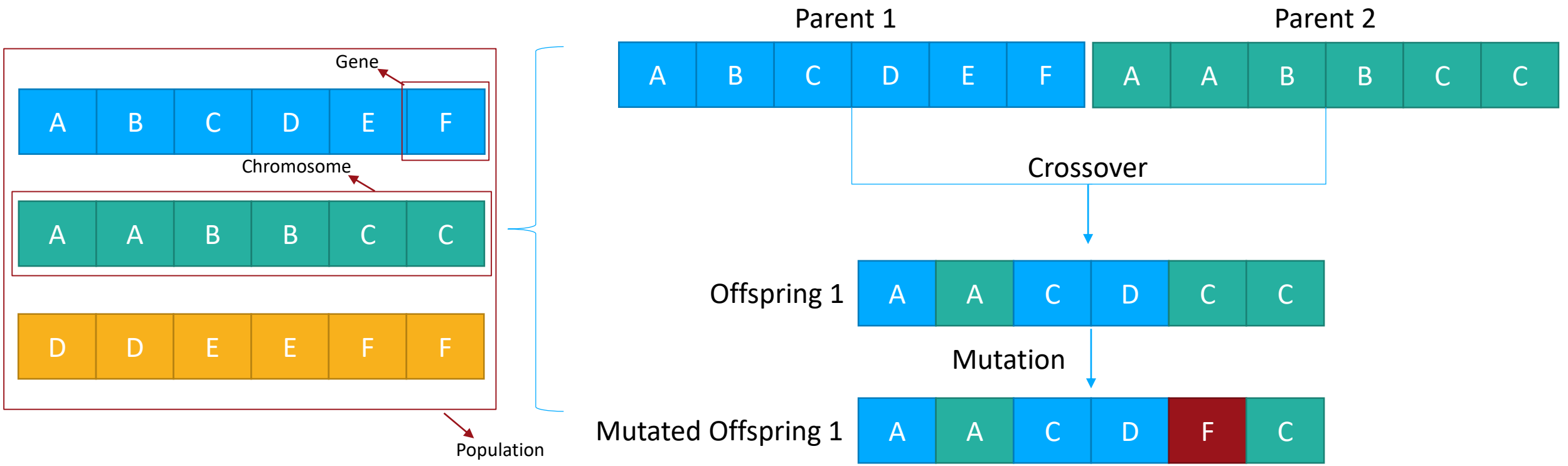
$$Workload\ Variance = \sqrt{\frac{\sum_{i=0}^{n-1}(WCET_i - avg(WCET))^2}{n}} \quad \downarrow\downarrow$$

$$Maximum\ Interference\ Cost = max([IC(\tau_0), \dots, IC(\tau_{n-1})]) \quad \downarrow\downarrow$$

- Constraints:

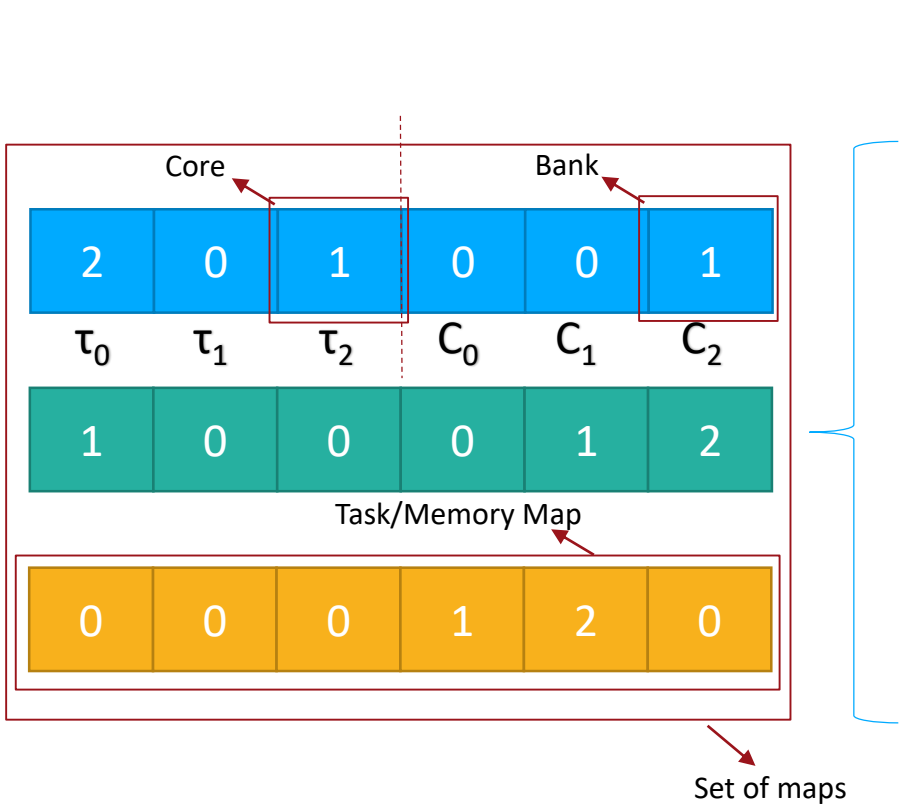
$$Deadline_i > WCET_i + IC(\tau_i) + (WCET_j | PE_j = PE_i)$$

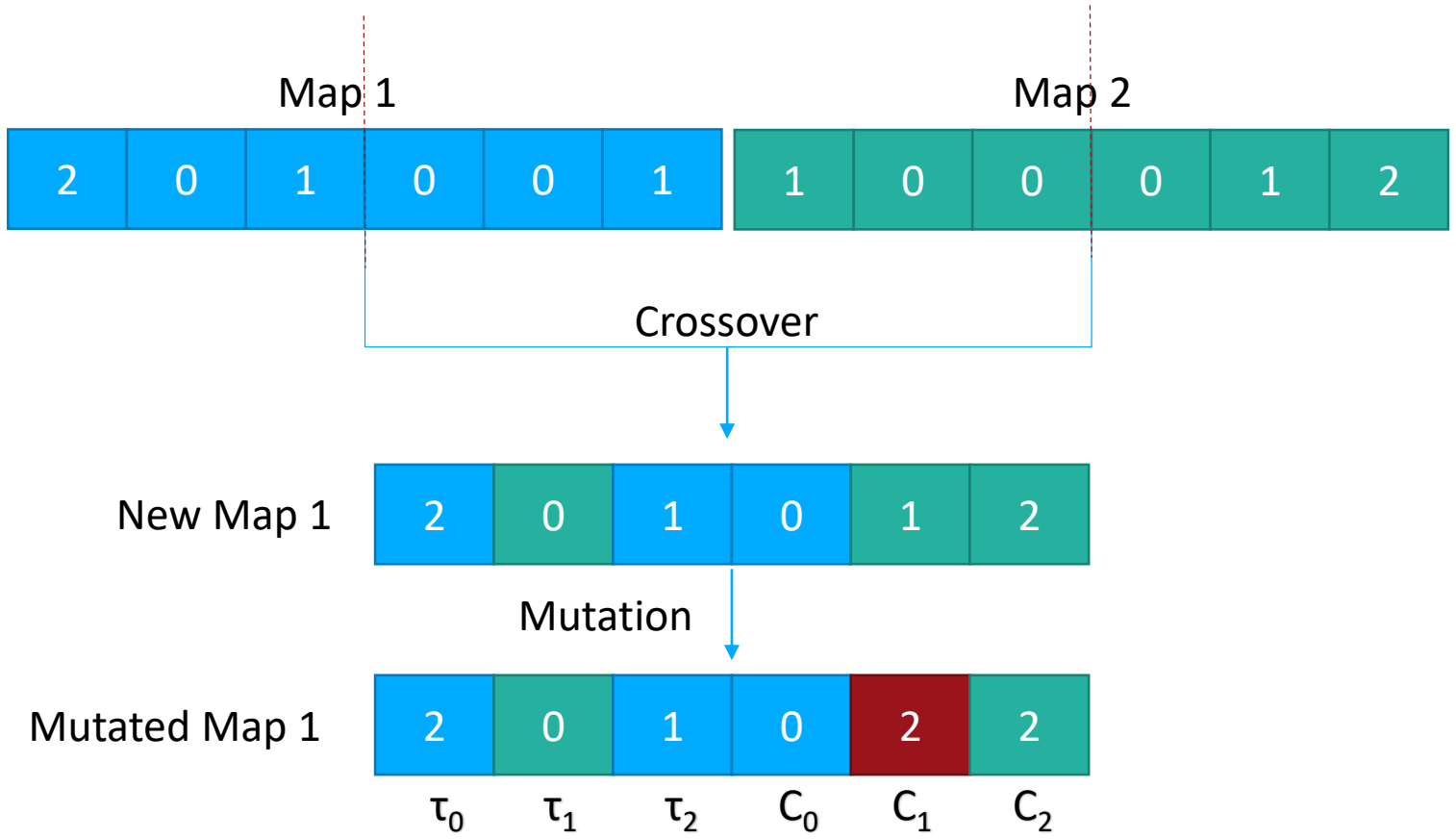
Meta-heuristics Multi-objective Optimization: Genetic algorithm



- Of all candidate offspring, the worst are discarded

Meta-heuristics Multi-objective Optimization: Genetic algorithm

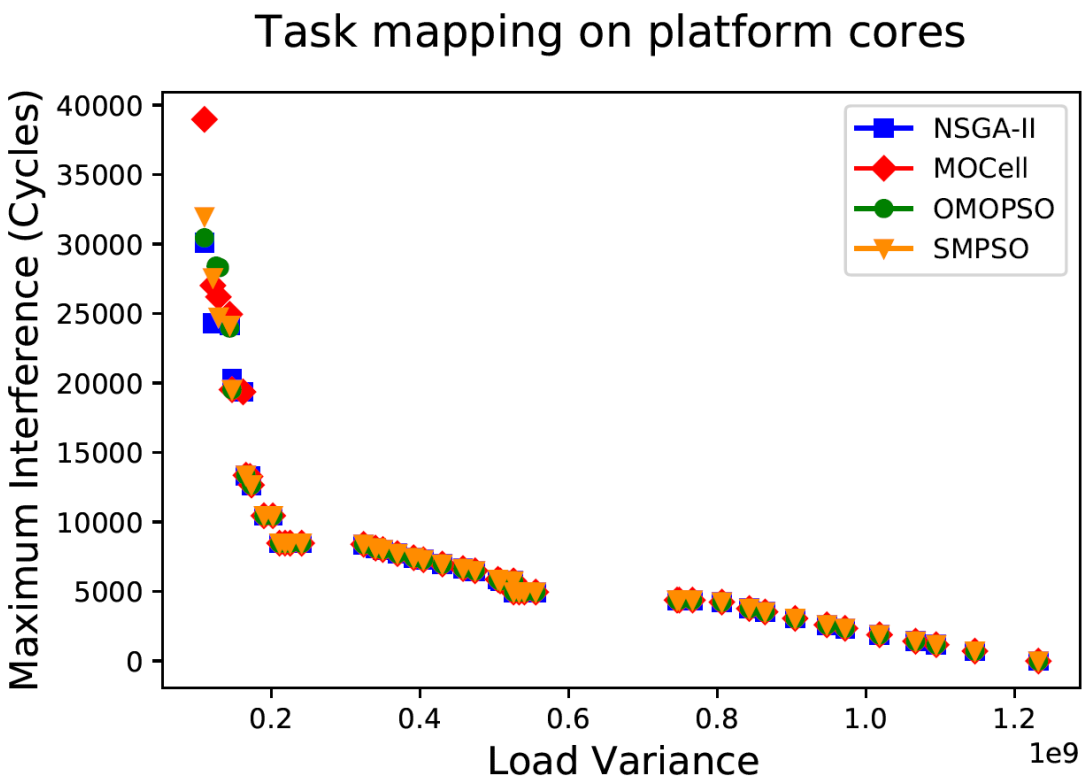




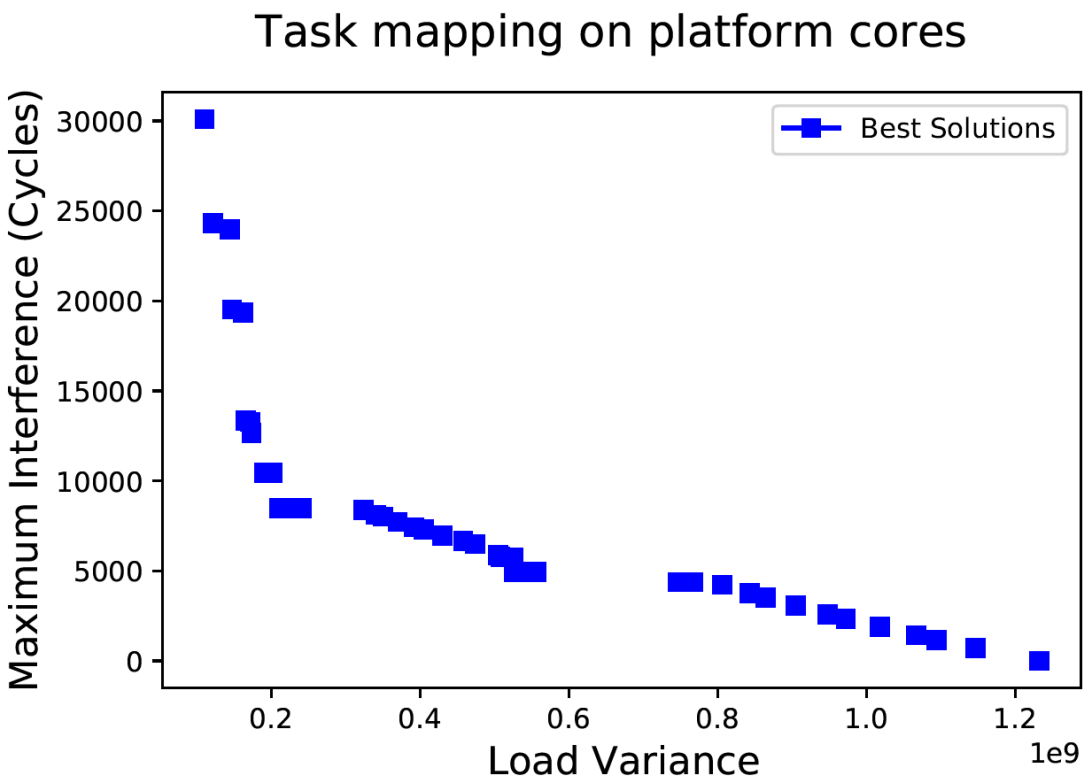
- Of all candidate offspring, the worst are discarded

Task/Memory Pareto Front – Keystone II

At the end of the algorithms execution, we obtain different Pareto front

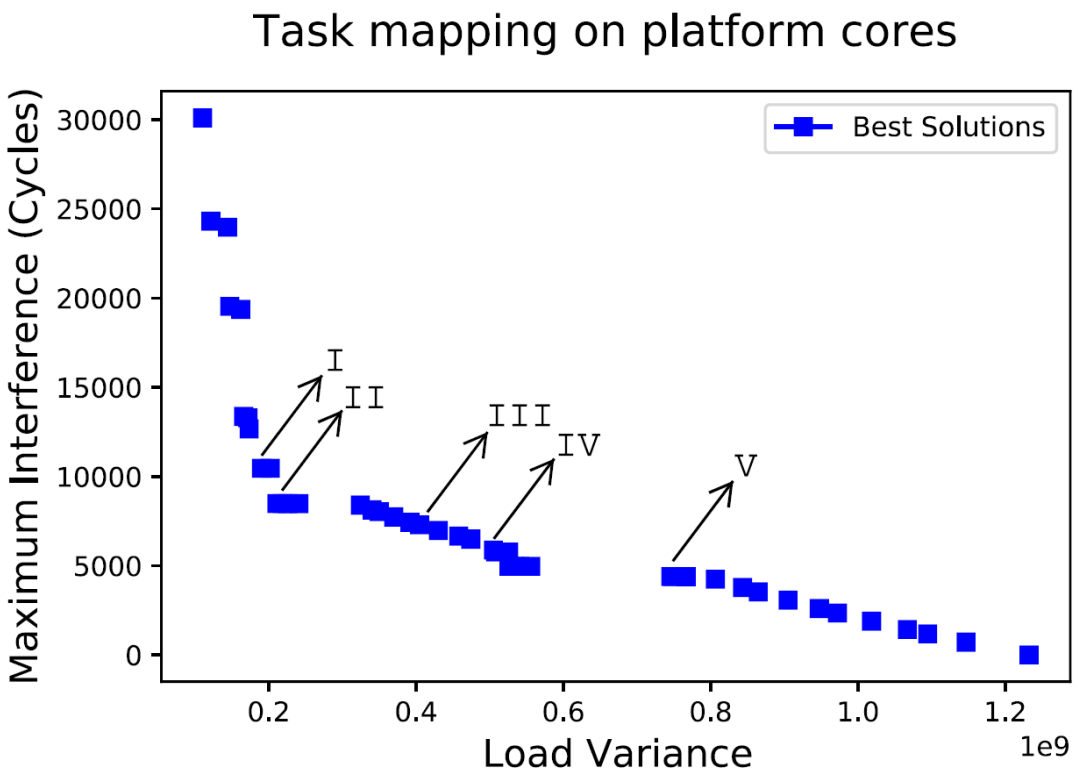


The dominated solutions from all the algorithms are removed. Another Pareto front is obtained

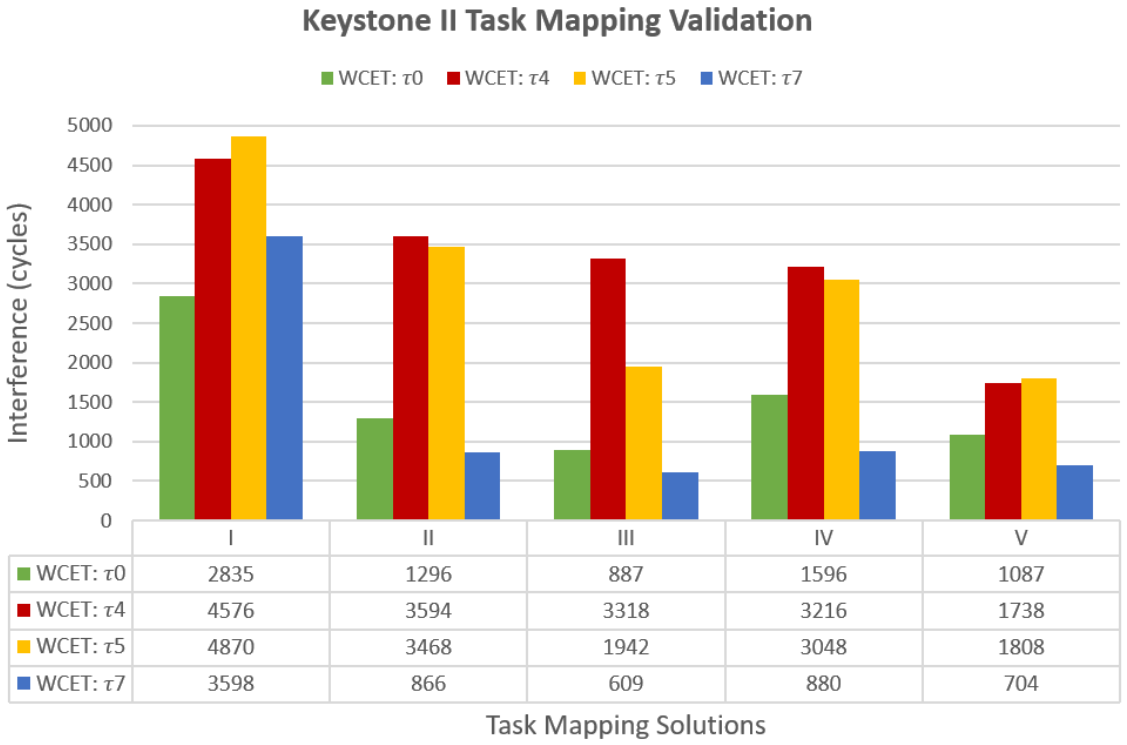


Task/Memory Map Evaluation – Keystone II

Some solutions are tagged for their evaluation



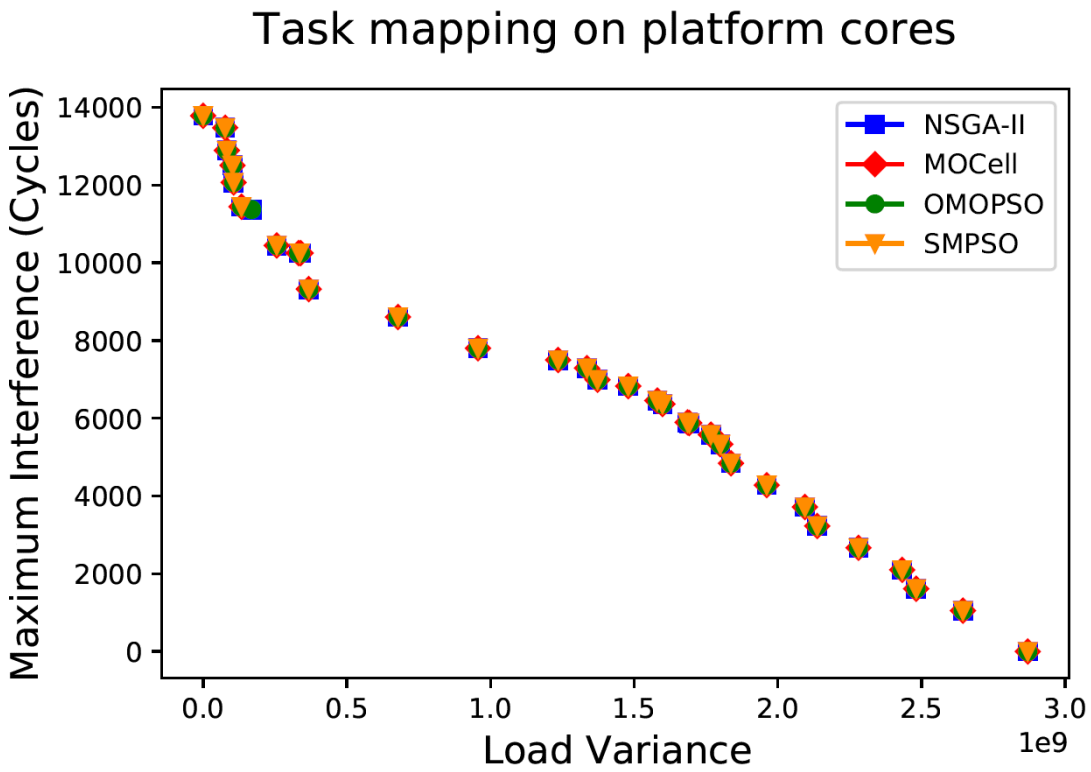
The maximum interference is measured for each solution



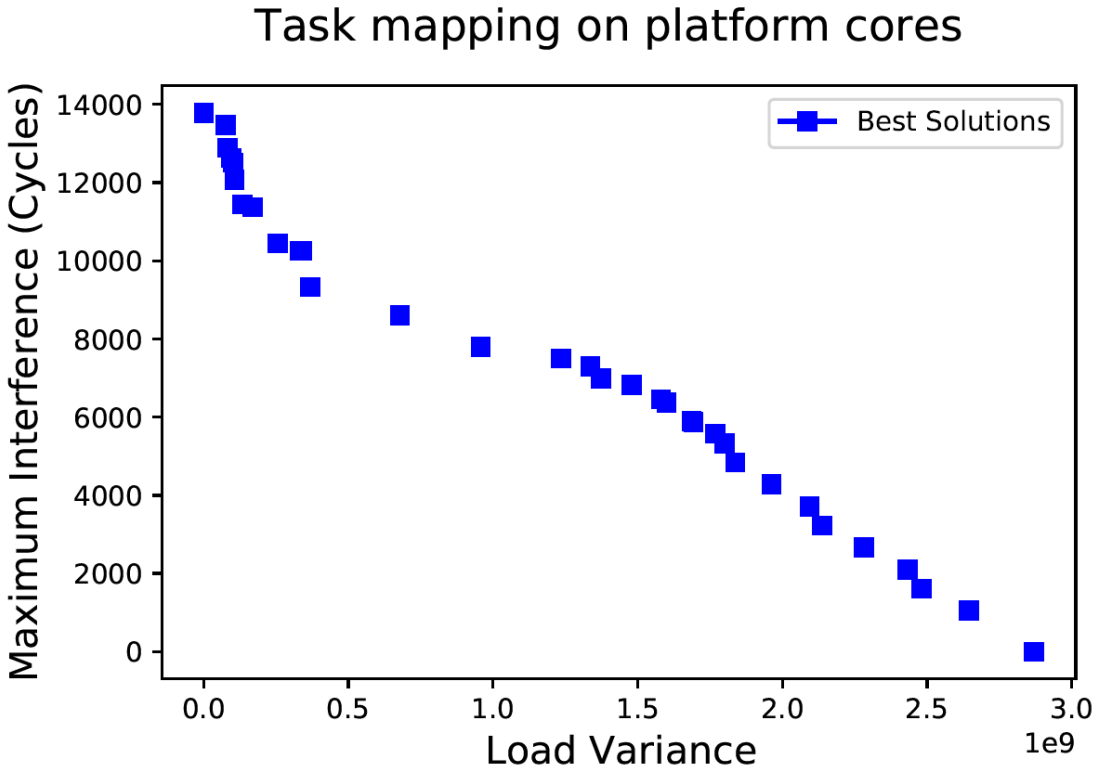
	I	II	III	IV	V
WCET: τ0	2835	1296	887	1596	1087
WCET: τ4	4576	3594	3318	3216	1738
WCET: τ5	4870	3468	1942	3048	1808
WCET: τ7	3598	866	609	880	704

Task/Memory Pareto Front – Sitara AM5728

At the end of the algorithms execution, we obtain different Pareto front

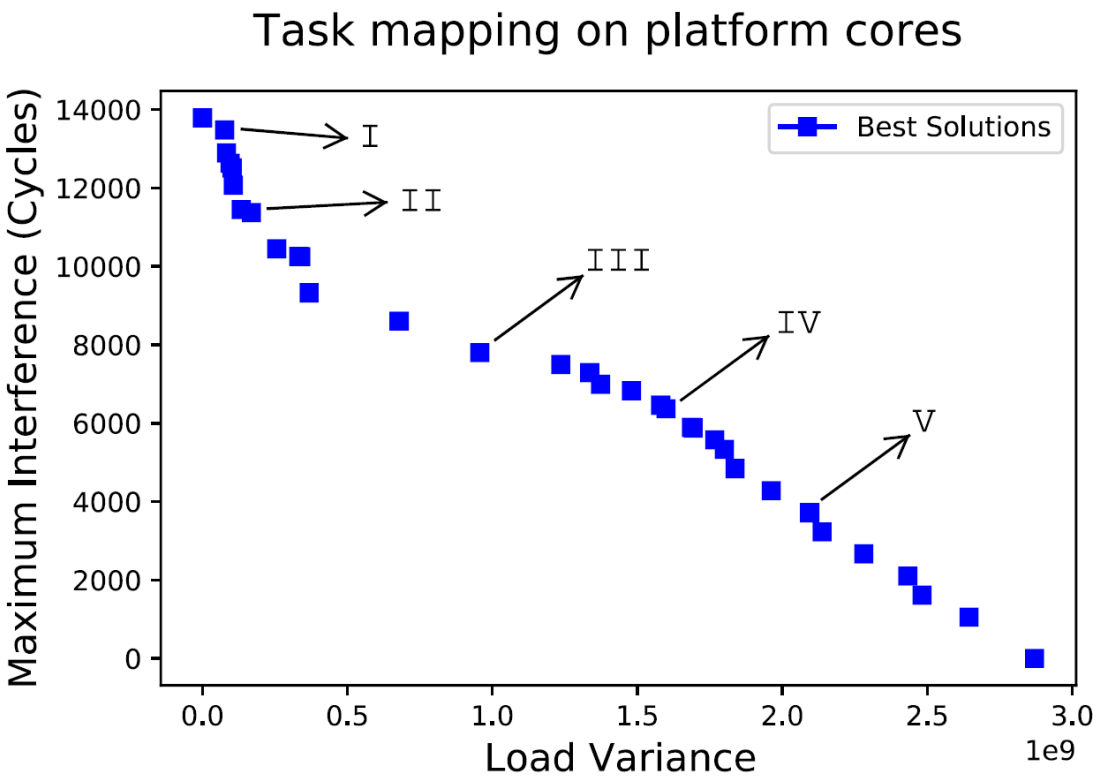


The dominated solutions from all the algorithms are removed. Another Pareto front is obtained

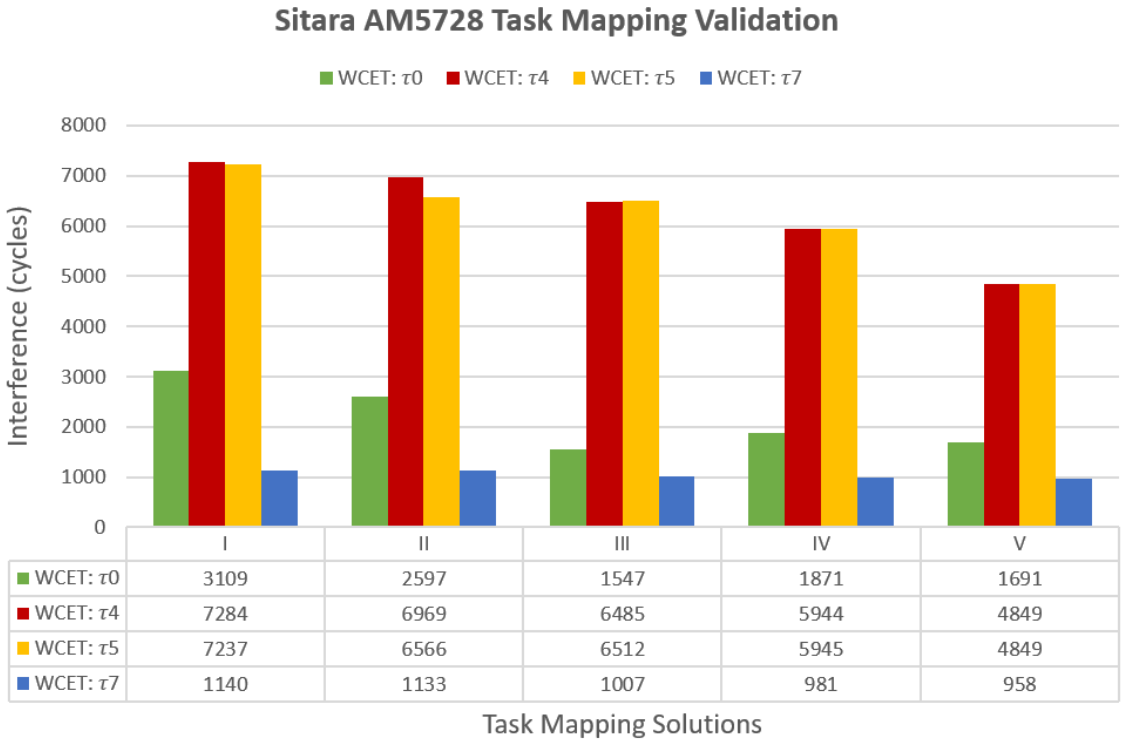


Task/Memory Map Evaluation – Sitara AM5728

Some solutions are tagged for their evaluation



The maximum interference is measured for each solution



	I	II	III	IV	V
WCET: $\tau 0$	3109	2597	1547	1871	1691
WCET: $\tau 4$	7284	6969	6485	5944	4849
WCET: $\tau 5$	7237	6566	6512	5945	4849
WCET: $\tau 7$	1140	1133	1007	981	958

Task/Memory Mapping Logic

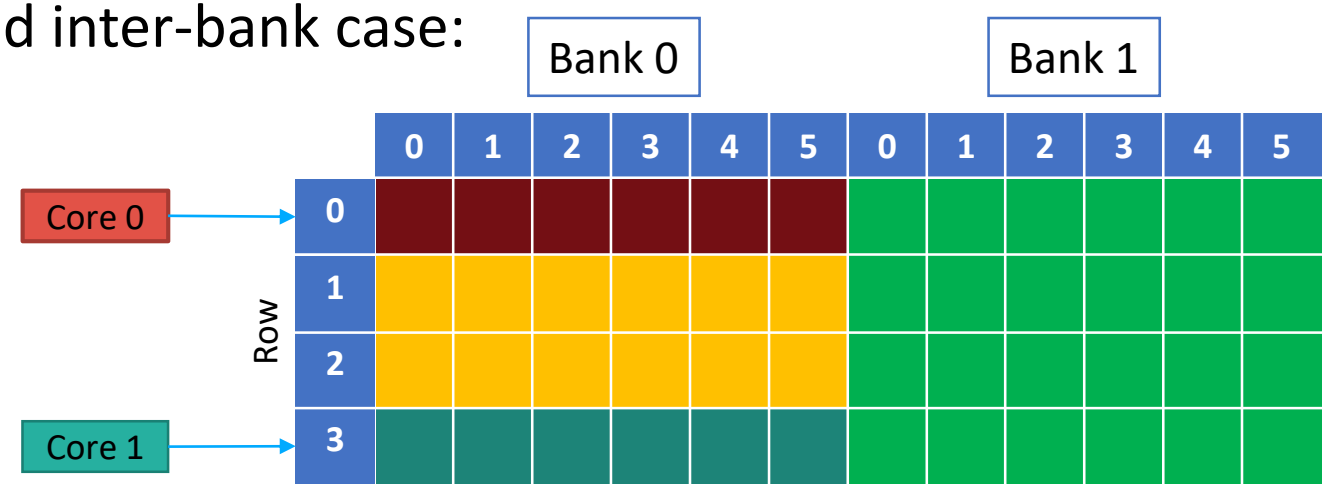
- Task-Core mapping:
  - Stack most intensive tasks on the same core (sequential execution, i.e., no interference) → Drawback = Less parallelism
  - Equally spread the tasks among the cores → Drawback = memory Interference
  - Increase the number of active cores (more parallelism) → Drawback = Generally memory Interference
  - Select the correct core type for the task (ARM,DSP): the execution time and interference vary → Drawback = None

- Core-bank mapping: Always try with a private bank and, when all of these are occupied (N°C > N°B), share bank

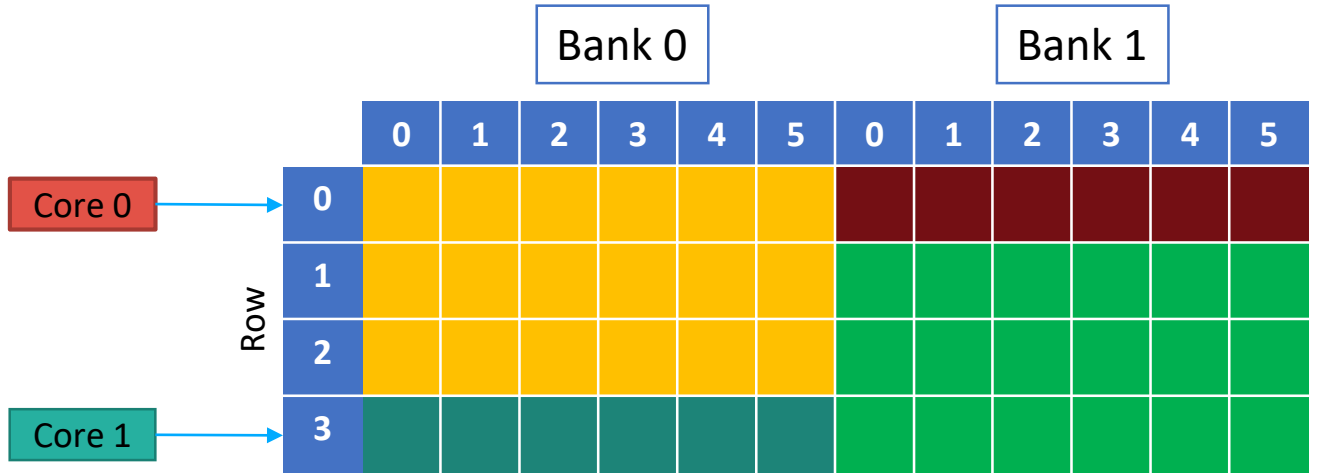
Why should we go for the private banks?

- For instance, let's compare the worst-case read transmission time cost, for the intra-bank and inter-bank case:

$$RT^{\text{intra}} = CL + \frac{BL}{2} + t_{RTW}$$



$$RT^{\text{inter}} = CL + \frac{BL}{2} + t_{RTW} - WL$$



Why should we go for the private banks?

- Another example is the DDR memory PREcharge command worst-case interference cost (PRE):
  - $PRE^{\text{intra}} = t_{RP}$
  - $PRE^{\text{inter}} = 1$

  where $t_{RP} \gg 1$

- And yet another example related to the DDR memory ACTive command worst-case interference cost (ACT):
  - $ACT^{\text{intra}} = t_{RCD}$
  - $ACT^{\text{inter}} = t_{RRD}$

  where $t_{RCD} > t_{RRD}$

Thank you for your attention