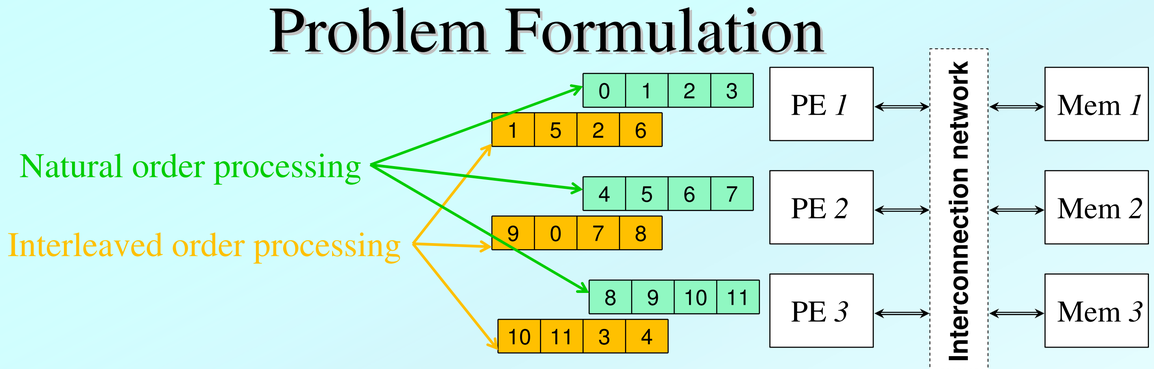
Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architecture

Cyrille CHAVET[2], Philippe COUSSY[2], Pascal URARD[1], Eric MARTIN[2]
[1]STMicroelectronics, Crolles, FRANCE. {FirstName.Surname@st.com}
[2]LabSTICC, Université de Bretagne Sud, CNRS UMR 3192. {FirstName.Surname@univ-ubs.fr}

Problem Formulation



An ***interleaver*** is a component that shuffles data from a given input data order (natural order; e.g. Nat = 0, 1, 2, 3, 4…), to a different output order (interleaved order; e.g. Int = 1, 5, 2, 6, 9...)

The problem is to find an in-place memory mapping avoiding memory access conflicts in both natural and interleaved order

Related Works

	Standard Interleaver	Architectural complexity	Algorithm complexity
Gnaedig *et al.*	NO	Low	High
When *et al.*	YES	High	Medium
Benedetto *et al.*	YES	High	High
Jezequel *et al.*	YES	High	Medium
STAR	YES	Medium	Medium
SAGE	**YES**	**Low / Medium**	**Low**

Proposed Design Flow

In the mapping matrix each element is filled with a memory bank (i.e. the corresponding data will be stored in this memory bank)

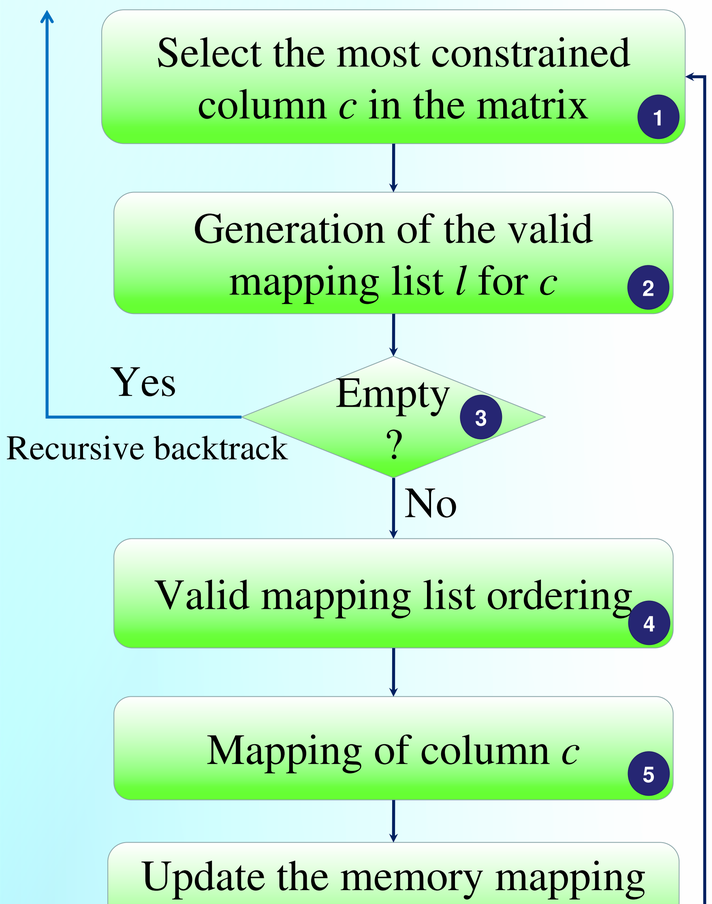
This memory mapping is done according to:

Structural constraints

- A data must be mapped in the same memory bank in natural and interleaved order (i.e. in matrix MAP_{Nat} and matrix MAP_{Int})
- In any column of MAP_{Nat} and MAP_{Int} each memory has to be used only one time

Architectural objectives

- The memory mapping in any column has to respect the rules of the steering components that compose the network



Natural access matrix

PE *1*	0	1	2	3
PE *2*	4	5	6	7
PE *3*	8	9	10	11

Interleaved access matrix

PE *1*	1	5	2	6
PE *2*	9	0	7	8
PE *3*	10	11	3	4

The data exchanges in natural and interleaved order are formalized through two distinct matrices

Natural mapping matrix — Map_{Nat}

-	-	-	-
-	-	-	-
-	-	-	-

Interleaved mapping matrix — Map_{Int}

-	-	-	-
-	-	-	-
-	-	-	-

The SAGE algorithm starts with two empty memory mapping matrices

Algorithm Iterations

	Map_{Nat}	Map_{Int}
Step I	A - - - B - - - C - - -	- - - - - A - C - - - B
Step II	A - - - B - A - C - - -	- - - A - A - C - - - B
Step III	A - C - B - A - C - B -	- - C A - A - C B - - B
Step IV	A A C - B - A - C C B -	A - C A C A - C B - - B
Step V	A A C - B B A - C C B -	A B C A C A - C B - - B
Step VI	A A C - B B A - C C B C	A B C A C A - C B C - B
Step VII	A A C A B B A B C C B C	A B C A C A B C B C A B

Iteration IV details

The resulting mapping matrix from step 3

Map_{Nat}

A	-	C	-
B	-	A	-
C	-	B	-

Map_{Int}

-	-	C	A
-	A	-	C
B	-	-	B

1 Selection of the most constrained column in Map_{Nat} and Map_{Int}

This selection step should select any of the three first columns of Map_{Int}

Let suppose that the algorithm selects the first column of Map_{Int}

Map_{Nat}

A	-	C	-
B	-	A	-
C	-	B	-

Map_{Int}

-	-	C	A
-	A	-	C
B	-	-	B

2 Generation of the valid mapping list for the selected column

In the selected column of Map_{Int} the memory bank B has been already used for data 10

As a consequence, the possible mapping are:

- ***data 1 in bank C and data 9 in bank A***
- ***data 1 in bank A and data 9 in bank C***

So the valid mapping list is:

$l = \{(C, A), (A,C)\}$

3 Test of the emptiness of the generated list l

$l = \{(C, A), (A,C)\}$ *=> The list is NOT empty*

4 Memory mapping list ordering

The list ordering aims to promote the mapping which respect the targeted interleaving architecture

Let's suppose that we target a barrel-shifter, then the list ordering will be:

$l = \{(A, C), (C,A)\}$

5 Memory mapping of the selected column

The first mapping of list l is then performed

Map_{Nat}

A	-	C	-
B	-	A	-
C	-	B	-

Map_{Int}

A	-	C	A
C	A	-	C
B	-	-	B

6 Update the memory mapping in the other matrix

The mapping performed in step 5 is reported in the second matrix

Map_{Nat}

A	A	C	-
B	-	A	-
C	C	B	-

Map_{Int}

A	-	C	A
C	A	-	C
B	-	-	B

Resulting mapping

Bank **A** = {*0, 1, 6, 3*}
Bank **B** = {*4, 5, 10, 7*}
Bank **C** = {*8, 9, 2, 11*}

Conclusion

SAGE is a memory mapping methodology dedicated to design parallel interleaver architecture

This approach allows to generate a valid memory mapping in any case, and if the interleaving law enables it, then the resulting memory mapping will respect a targeted interconnection network

Patent in France n°0754793 and patent pending in USA n°20090031094:
C.Chavet, P.Coussy, P.Urard, E.Martin, "*Apparatus for data interleaving algorithm*"

« *We can't solve problems by using the same kind of thinking we used when we created them* »

Albert Einstein