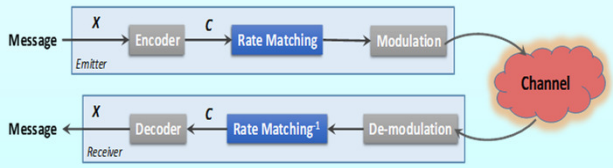
SOLVING MEMORY ACCESS CONFLICTS IN LTE-4G STANDARD

Cyrille CHAVET, Fabrice LOZACHMEUR, Thomas BARGUIL, Awais SANI, Philippe COUSSY
Lab-STICC (*CNRS UMR 6285*)
Université Bretagne Sud, Lorient
{FirstName.LastName@univ-ubs.fr}

Problem Formulation

In order to achieve high throughput requirements, ECC decoders are based on parallel architectures, which results in a major issue: memory access conflicts.
In the **LTE-4G standard**, it seems that this problem have been partially solved. In fact it is known that it is not really the case, but the basic idea was originally to remove memory mapping conflicts in the ECC decoder.

The problem is that in fact, these conflicts has been transferred from the decoder itself, to its channel interface. The question is then how to store data to avoid any conflict between the inverse Rate Matching and the Decoder?



Our goal is to reverse-engineer the LTE-4G Rate Matching algorithm in order to find out the *mathematical key* to solve all conflicts for any case.

Related Works

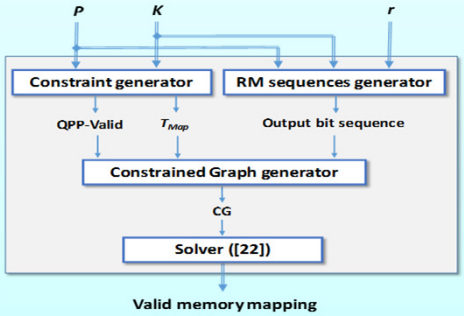
	Online mapping	Any standard	Architecture complexity	Algorithm complexity
Tarable *et al.*	NO	YES	High	High
Wehn *et al.*	NO	YES	High	Medium
Muller *et al.*	NO	YES	High	Medium
Briki *et al.*	NO	YES	High	Medium
Ur Rehman *et al.*	NO	YES	Medium	Medium
Gnaedig *et al.*	YES	NO	Low	Medium
Sani *et al.*	NO	YES	Medium	Low
Our contribution	**YES**	**NO**	**Low**	**Low**

Proposed Approach

We propose to replace the original QPP input order constraint by a custom "Temporary mapping constraint" *TMap*. The idea is to find a mapping constraint that is a simple permutation π of the original QPP data input order, but easier to solve.

This approach relies on a hypothesis resulting from the observation of the behavior of the LTE-4G rate matching algorithm: for a given set of K data, a parallelism P, and a set of $B=P$ memory banks:

(1) each memory banks stores K/P data;
(2) for a given memory address α, the difference between any 2 data in each memory bank at address α is lower than P, and they can be ordered incrementally.



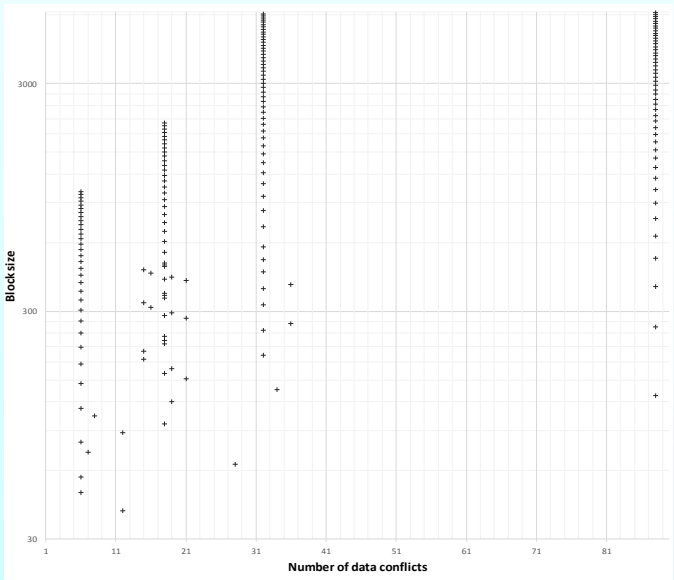
Then, a Constraint Graph CG taking into account all these constraints is defined and a constraint solver (*Ur Rehman et al. DATE 2015*) is applied to find a conflict free memory mapping.

Conclusion

Our proposed conflict free memory mapping approach, based on a two steps algorithm (temporary mapping constraints and a constraint solver), has been validated for any of the 188 block sizes defined in the standard, and for parallelism from 2 to 32.

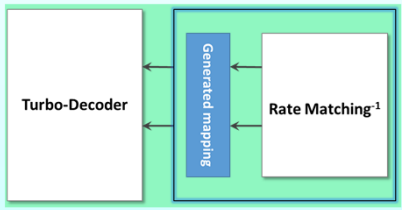
Observation

We designed a Rate Matching algorithm in order to evaluate the number of conflict between the inverse Rate Matching and the decoder input.



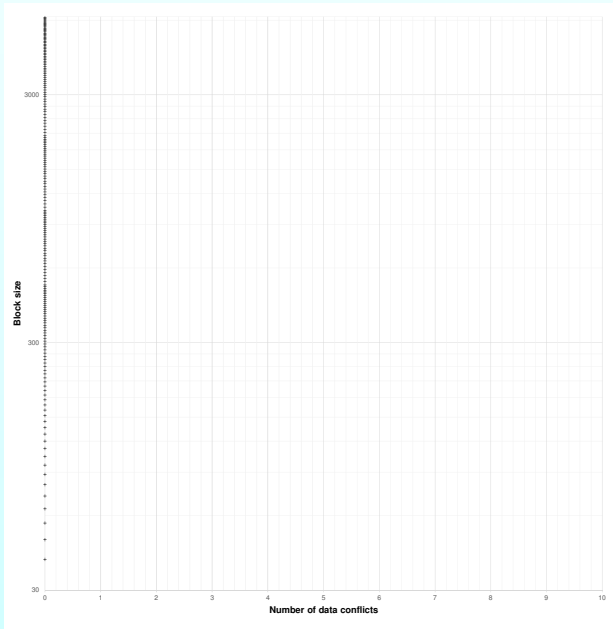
Number of conflicts for the 188 frame lengths (from 40 to 6144) and parallelism P=8.

Targeted architecture



Memory mapping generated by our approach instead of the naïve mapping

No more conflict



Number of conflicts for the 188 frame lengths (from 40 to 6144) and parallelism P=8.