

# Application-Specific Embedded Logic-in-Memory



Qiuling Zhu



Franz Franchetti



Larry Pileggi

Recent studies of a sub-20nm construct-based design methodology show opportunities to insert logic patterns reliably next to memory patterns without concern for creating layout hotspots. Leveraging this capability we propose application-specific logic-in-memory (LiM), a novel computing paradigm that is energy-efficient particularly for high-performance data-intensive computing (See Fig. (a)). LiM moves part of a program's computation directly into the memory but keeps the familiar memory interface. The embedded logic computation is highly specialized for a particular application and is integrated as close as possible with the memory array. LiM benefits from algorithm and application knowledge to optimize the hardware to a level that is impossible with general purpose computing or configurable hardware computing.

We have built a design framework enabling the end-to-end hardware generation of LiM blocks (Fig.1 (b)). Our tool provides designers with a graphical user interface to design, customize, and optimize LiM structures. Importantly, the ability to physically synthesize these LiM blocks affordably is enabled by a “smart” memory compiler (see related task) that uses construct-based logic and memory. We show the applicability for a wide range of applications from signal to image processing (e.g., FFT and radar image reconstruction). Synthesis results at 14nm technology show that the resulting LiM hardware is efficient in area, power and latency (See Fig.2 (a)), and our architectural simulation demonstrates orders of magnitude energy savings as compared with traditional processor-centric computing (See Fig.2 (b)).

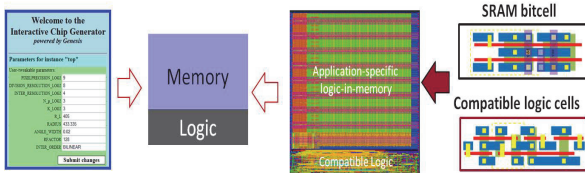


Fig. 1 (a): Application-specific LiM Computing Paradigm.

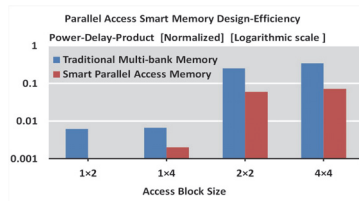


Fig. 2 (a): Hardware Evaluation.

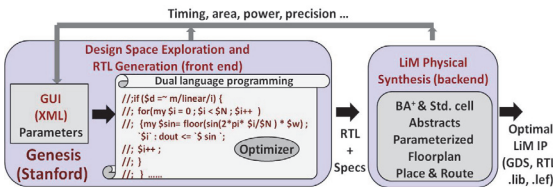


Fig. 1 (b): Application-specific LiM Design Framework.

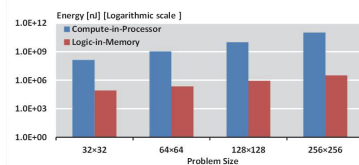


Fig. 2 (b): Energy Evaluation.