Application-Specific Logic-in-Memory for Polar Format Synthetic Aperture Radar *

Qiuling Zhu, Eric L. Turner, Christian R. Berger, Larry Pileggi, Franz Franchetti
{qiulingz, elturner, franzf}@ece.cmu.edu
Dept. Electrical and Computer Engineering, Carnegie Mellon University

Introduction

In the conventional von Neumann model, where computing systems are physically and logically split between memory and CPUs, the "memory wall" and "power wall" are well known bottlenecks that have severely limited the energy efficiency of many applications. Running today's memory intensive applications, modern computers spend most of their time and energy moving data rather than on computation.

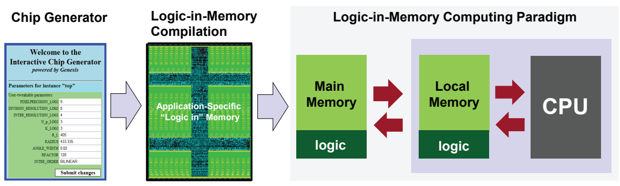


Figure 1: Application-Specific Logic-in-Memory

Application Specific Logic-in-Memory. To enable large savings of energy in memory intensive applications, we propose a novel computing paradigm—*Application Specific Logic-in-Memory*—by blurring the distinction between memory and processing logic. As shown in Fig. 1, appropriate application-specific logic is tightly integrated into the on-chip "dumb" memory to enable localized computation. Compared with the well known "Processing-in-Memory" (PIM) [3], the key is to be application-specific, which benefits from algorithm and problem-level knowledge to optimize the embedded logic and memory to a level that is impossible with traditional processor designs. The proposed logic-in-memory blocks act like special-purpose caches or scratch-pads. From the software perspective they appear as an ordinary memory block, but return data that is computed.

Interpolation Memory. The new memory-centric computational paradigm requires that computational logic is simple enough to be easily embedded into memory arrays for customized localized computation (basic boolean operations, fixed-point additions, shifts, multiplier-less constant multiplications, etc.). While many applications might benefit from this methodology, we initially focus on "interpolation memory", a logic-in-memory unit that combines a seed table with simple arithmetic logic to efficiently evaluate functions [5]. Our preliminary investigation of 1D interpolation memory has shown very promising results in memory-intensive signal processing applications, and we have reason to believe that this technology can be widely used in many high performance embedded applications.

*This work was supported by DARPA/SRC FCRP C2S2. The authors acknowledge the support of the C2S2 Focus Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity.

Case Study: Grid Interpolation in SAR. In Synthetic Aperture Radar (SAR), the commonly used Polar Format Algorithm (PFA) is memory intensive. The majority of the processing time and power is consumed by a grid interpolation involving a resampling of the radar reflectivity function from a curvilinear grid, the polar annulus, to a rectangular grid, the Cartesian grid array [1]. We find that this "re-gridding" can be also achieved using simple interpolation algorithms; e.g., local bilinear or bicubic interpolation. This makes the PFA a promising candidate for the logic-in-memory computing paradigm and its associated design tools. Simulation results show that the proposed localized interpolation technique yields comparable errors as conventional more computation-intensive FFT-based interpolation algorithms.

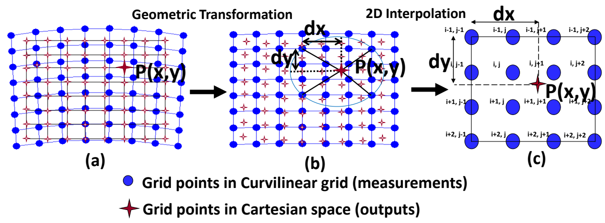
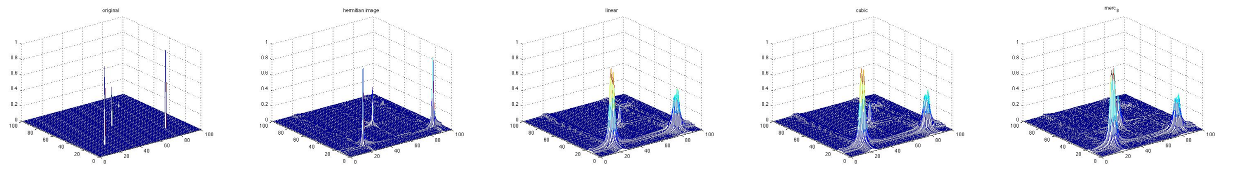


Figure 2: Localized Polar-to-Rectangular Grid Interpolation

Application-Specific Logic-in-Memory for SAR

The proposed implementation of the PFA re-gridding is based on geometric approximations, image transformation, 2D surface interpolation, as well as several advanced automatic design methodologies. A high-level visual representation of the process with bicubic interpolation is shown in Fig. 2.

Coordinate Conversion by Geometric Transformation. The measurements of the radar reflectivity function are taken on partial polar annuli, which need to be converted to outputs on a Cartesian grid, see Fig. 2(a). To apply low complexity local interpolation, it is preferable that the measurements are in a rectangular grid, although then the locations of the tentative outputs will not be. To achieve this, we use a four-corner image geometric mapping, specifically a perspective transformation, see e.g. [7]. For this to be applicable, we approximate the partial polar annuli as straight lines, making the full shape quadrilaterally tiled. The tentative output locations are mapped into the same coordinate system, such that the distances dx and dy to each of the neighboring measurements can be determined, see example in Fig. 2(c). Most of this transformation involves trivial arithmetic logic. Although



(a) Original scene (b) Gold standard (c) Linear interpolation (d) Cubic interpolation (e) FFT upsampling method

Figure 3: An original point target scene, with the outputs using various interpolation methods

the division operation is required for perspective transformations, the denominator is only a linear function of the coordinates in the Cartesian grid. So it can be implemented by a simple 2D linear interpolation followed by a multiplication leading to negligible accuracy loss. Overall, the whole geometric transformation logic can be efficiently embedded into the memory array via the logic-in-memory method.

Re-gridding by 2D Surface Interpolation. After the coordinate transformation, the measurements lie on a rectangular grid, while the tentative outputs lie on a quadrilateral in the new coordinate system, see Fig. 2(b). 2D surface interpolation techniques are then used to calculate the radar reflectivity function values at the output locations—Fig. 2(c) shows the example of bicubic interpolation, which involves 16 neighboring measurements. 2D surface interpolation algorithms can be separated into multiple 1D interpolations along orthogonal axes. The bicubic interpolation shown in Fig. 2(b) can be divided into four horizontal 1D interpolations and one vertical 1D interpolation (or vice versa), improving computational efficiency.

Chip Generator. The image formation process requires a series of problem parameters such as the number of grid points (i.e., the size of the grid) and the distances between them. These parameters are determined by SAR image size and geometry as well as radar parameters. Different settings lead to different hardware designs. On the other hand, the proposed localized interpolation is a trade-off problem in terms of performance/accuracy/cost. To automatically build various design points and to allow for algorithm-level design optimization, we used the Genesis2 design tool developed by Stanford University [2, 6] to build an application-specific chip generator. We codified all the combinations of SAR problem specifications and design trade-offs into a design template and built an entire family of the SAR image format processing chip designs to evaluate algorithm/implementation trade-offs.

Smart Logic-in-Memory Compiler. To ensure robust and energy-efficient circuit design in sub-22nm, we map memory and logic onto a set of pre-characterized pattern constructs, enabling logic-in-memory [4]. Exploiting the opportunities provided by modern process and physical design tools, applications can be compiled into the smart memory modules where logic is integrated into the embedded memory. Since 2D interpolation requires one-cycle functional access of a rectangular image window, we create a rectangular-access smart memory with the proposed logic-in-memory compiler and it provides rectangular accessibility at any pixel position. This helps to save a significant area and power overhead of memory peripheral circuits compared with the conventional multi-banking memory design approach. With this new design methodology and associated suite of design tools, design optimization and customization will be enabled at all the levels of abstraction (i.e. architectural, logic, and physical).

Experimental Results

While the proposed implementation uses a localized bilinear or bicubic interpolation, typically a different interpolation scheme is used in existing implementations. Commonly, two 1D interpolation steps are used that upsample the polar grid uniformly and then use a nearest-neighbor approach to find values on the rectangular grid [1]. Since uniform upsampling can be done efficiently using the fast Fourier transform, this method is seen as advantageous, although the memory access pattern is not suitable for logic-in-memory. Comparing the output of the common approach with that of the proposed bilinear and bicubic interpolation, we see that the distortion caused by these interpolation methods are statistically indistinguishable from one another.

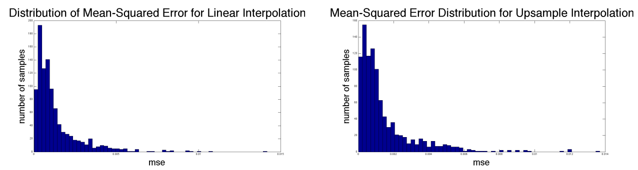


Figure 4: Probability Distribution of Mean-Squared-Error (MSE) for linear interpolation (left) and FFT-based upsampling interpolation (right)

For quantitative comparison we simulated a randomized radar scene of point targets and performed the re-gridding using each interpolation method, see Fig. 3. A reference "gold standard" is included that is based on the non-uniform inverse Fourier transform of the polar samples, which can be computed for point targets. The MSE in Fig. 4 is computed relative to the output of this gold standard. This comparison shows that each of the three interpolation methods considered result in the same level of output image distortion.

References

[1] W. G. Carrara, R. S. Goodman, and R. M. Majewski. *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*. Artech House, 1995.

[2] Available [online] http://genesis2.stanford.edu/mediawiki/index.php.

[3] P. M. Kogge, T. Sunaga, H. Miyataka, K. Kitamura, and E. Retter. "Combined DRAM and Logic Chip for Massively Parallel Systems,". *Conf. Advanced Research in VLSI*, 1995.

[4] D. Morris, V. Rovner, L. Pileggi, A. Strojwas, and K. Vaidyanathan. "Enabling Application-Specific Integrated Circuits on Limited Pattern Constructs,". *Symp. VLSI Technology*, June 2010.

[5] A. S. Noetzel. "An Interpolating Memory Unit for Function Evaluation: Analysis and Design,". *IEEE Trans. Computers*, 38(3):377–384, 1989.

[6] M. Wachs W. Qadeer Z. Asgar K. Kelley J. P. Stevenson S. Richardson O. Shacham, O. Azizi and M. Horowitz. "Rethinking Digital Design: Why Design Must Change,". *IEEE Micro*, 30(6):9–24, 2010.

[7] G. Wolberg. *Digital Image Warping (Systems)*. IEEE Computer Society Press, 1990.