Near Memory Computing
Spectral and Sparse Accelerators

Franz Franchetti

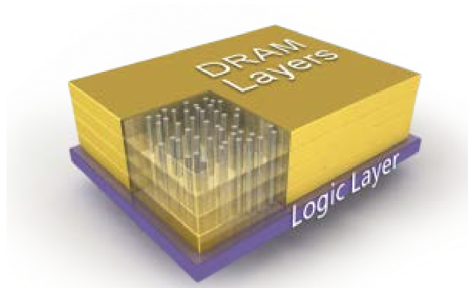
ECE, Carnegie Mellon University
www.ece.cmu.edu/~franzf

Co-Founder, SpiralGen
www.spiralgen.com

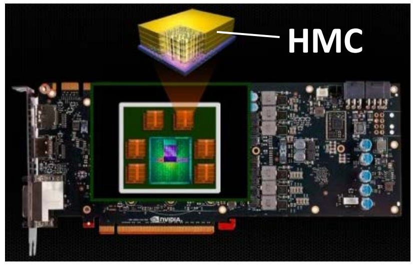
The work was sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement No. HR0011-13-2-0007. The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of DARPA or the U.S. Government. No official endorsement should be inferred. This work was also supported in part by the Intelligence Advanced Research Program Agency and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-12- C-2008, Program Manager Dennis Polla.

DRAM-Optimized Near Memory Acceleration

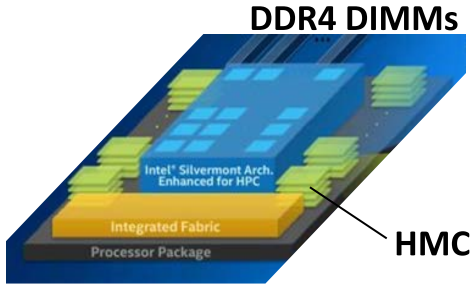
- **Enabling technology: 3D stacked integration**
  - Logic and DRAM layers connected by TSVs
  - Better timing, lower area, advanced IO in the logic layer



Micron HMC



Nvidia Volta



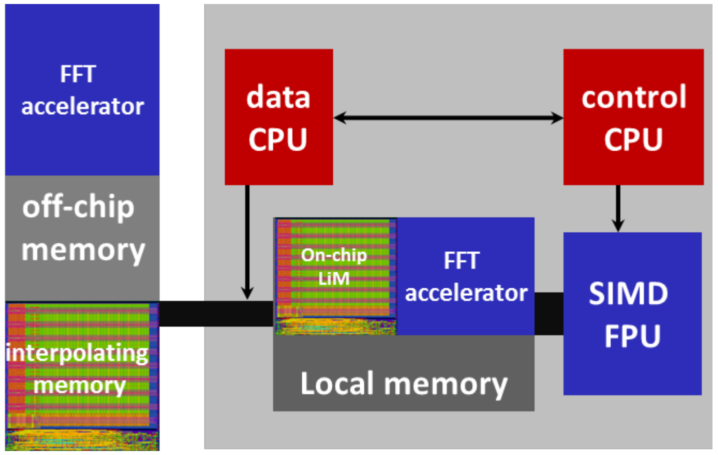
Intel Knights Landing

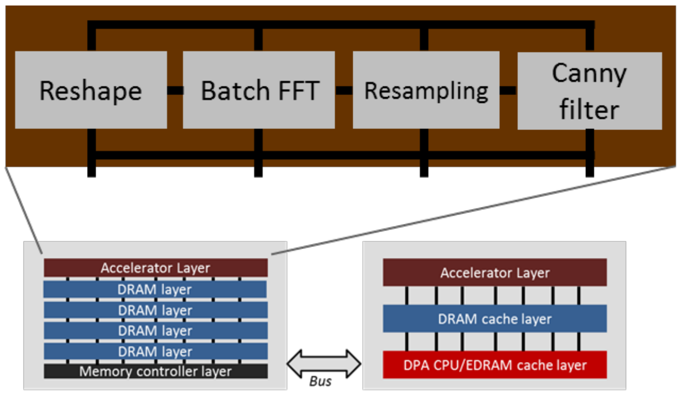
- **Accelerators in 3D DRAM, behind the conventional interface**
  - Bandwidth and latency concerns still exist to off-chip
  - Integration behind conventional interface opens up internal resources

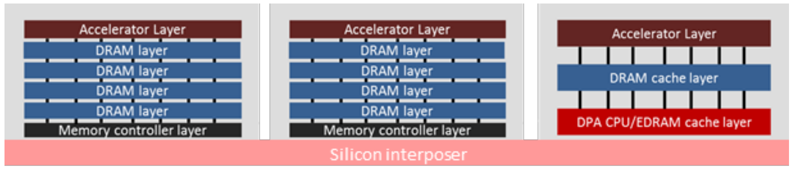
- **DRAM operation and organization aware accelerators**
  - Conventional near memory computing only aim to reduce the communication distance between memory and processor

Concept: Memory-Side Accelerators

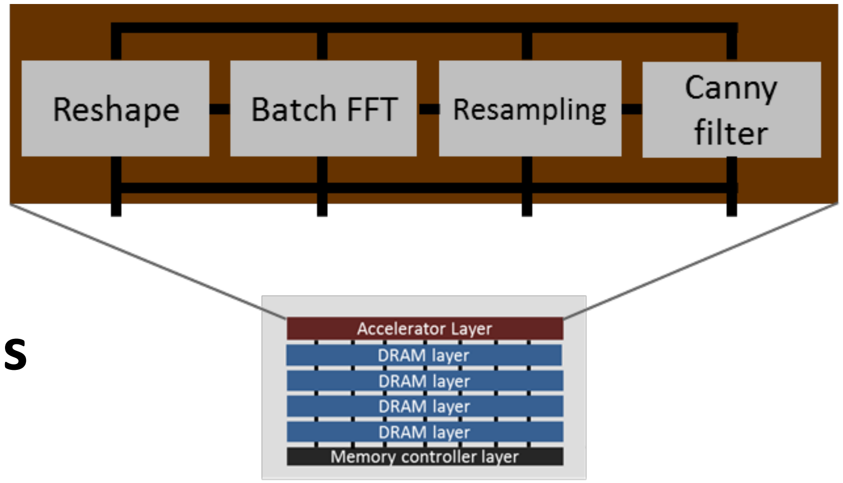
- **Idea: Accelerator on DRAM side**
  - No off-DIMM data traffic
  - Huge problem sizes possible
  - 3D stacking is enabling technology
- **Configurable array of accelerators**
  - Domain specific, highly configurable
  - Cover DoD-relevant kernels
  - Configurable on-accelerator routing
- **System CPU**
  - Multicore/manycore CPU
  - CPU-side accelerators
  - Explicit memory management
  - SIMD and multicore parallelism







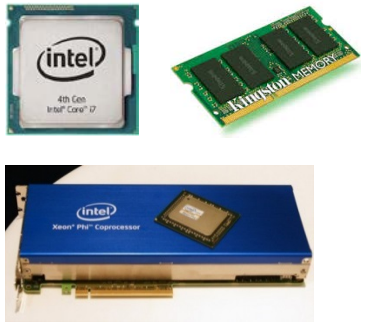
Memory-Side Accelerator Architecture



- **DRAM-side Accelerator**
  - LiM layer in stacked DRAM
  - Array of acceleration engines
  - Domain specific, highly configurable
- **Customized Acceleration Pipelines**
  - DRAM-to-DRAM streaming
  - Connect and configure multiple engines
  - Configurable on-LiM routing
- **Accelerator Cores**
  - **Reshape:** linear-to-block data layout changes
  - **Batch FFT:** primitive for large 1D and 2D FFTs
  - **Resampling:** interpolation, geometric transformations, image alignment
  - **Feature extraction:** edge and shape detection
- **Example:** PFA SAR requires Reshape, Batch FFT and Resampling

Simulation, Emulation, and Software Stack

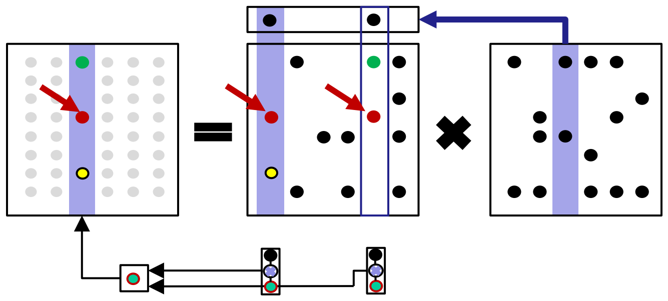
- **Accelerator Simulation**
  - **Timing:** Synopsis DesignWare
  - **Power:** DRAMSim2, Cacti, Cacti-3DD McPAT
- **Full System Evaluation**
  - Run code on real system (Haswell, Xeon Phi) or in simulator (SimpleScalar,…)
  - Normal DRAM access for CPU, but trap accelerator command memory space, invoke simulator
- **API and Software stack**
  - **Accelerator:** memory mapped device with command and data address space
  - **User API:** C configuration library, standard API where applicable
  - **Virtual memory:** fixed non-standard logical-to-physical mapping
  - **Memory management:** Special `malloc/free`, Linux kernel support



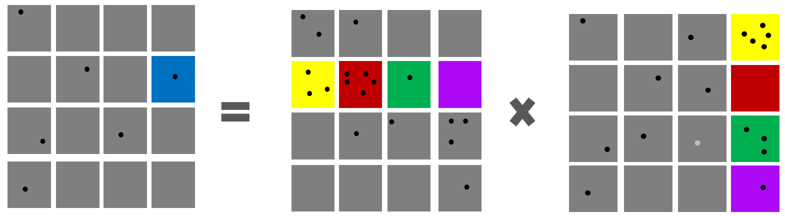
```
#include <accelerator-fftw3.h>
...
{
  fftw_complex *in, *out;
  fftw_plan p;
  ...
  in = (fftw_complex*)
    fftw_malloc(sizeof(fftw_complex) * N);
  out = (fftw_complex*)
    fftw_malloc(sizeof(fftw_complex) * N);
  p = fftw_plan_dft_1d(N, in, out,
    FFTW_FORWARD, FFTW_ESTIMATE);
  ...
  fftw_execute(p); /* repeat as needed */
  ...
  fftw_destroy_plan(p);
  fftw_free(in); fftw_free(out);
}
```

Sparse Matrix Multiplication Accelerator

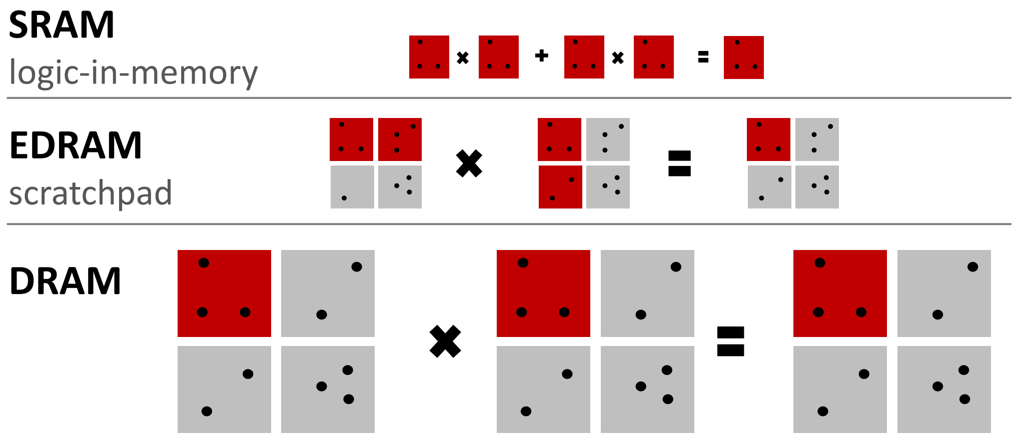
On-chip: regular SpGEMM kernel

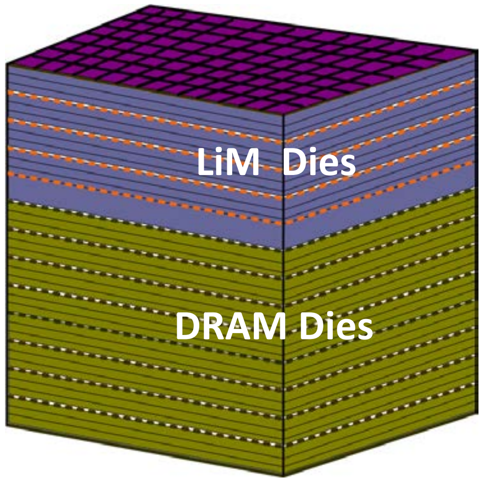


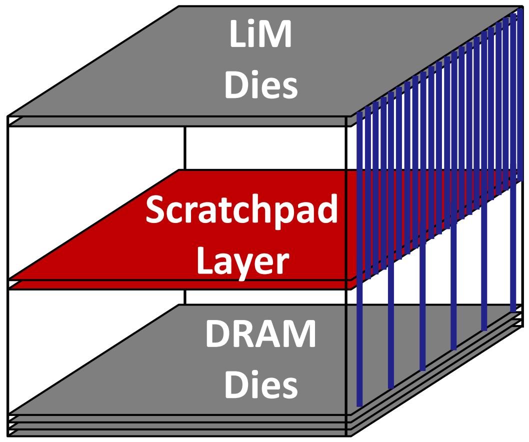
Off-chip: SRUMMA (tiled shared memory algorithm)



Scratchpad: Hierarchical tiling

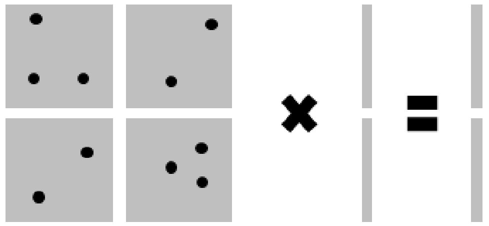






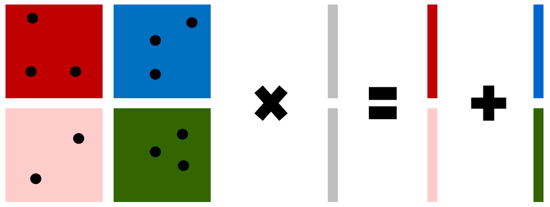
Sparse Matrix-Vector Product Accelerator

SpMV Kernel



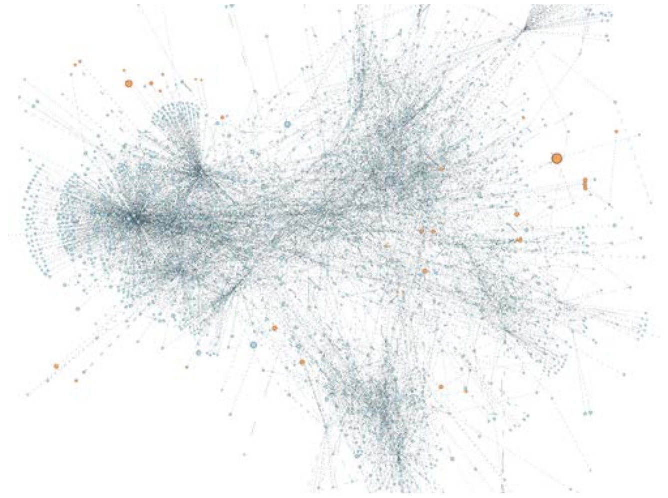
- Standard sparse matrix times dense vector
- Matrix is very sparse (a few non-zeroes per row)
- Matrix and vector size is large (>10M x 10M/GB)

Algorithm

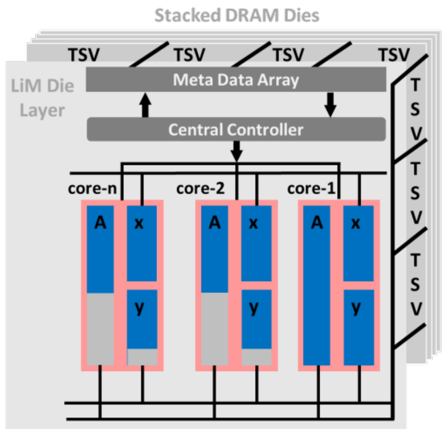


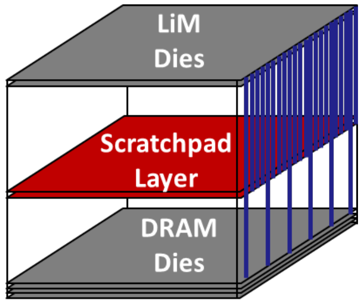
- Block-column multiply + merge step
- Partial results are streamed to DRAM
- Matrix streams from DRAM
- Vector segment is held in scratchpad/EDRAM

Target: Larges Sparse Graphs



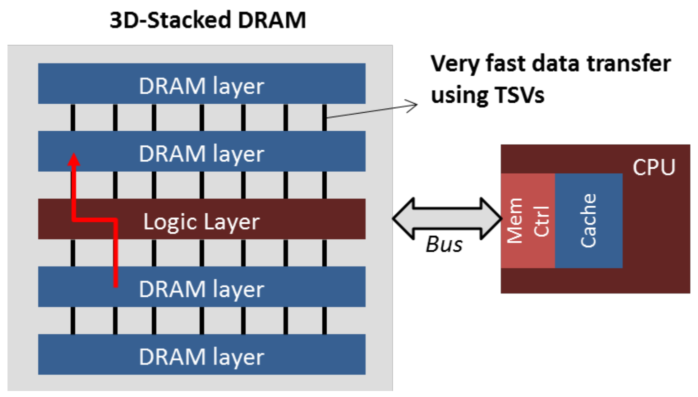
3DIC Memory Side Accelerator



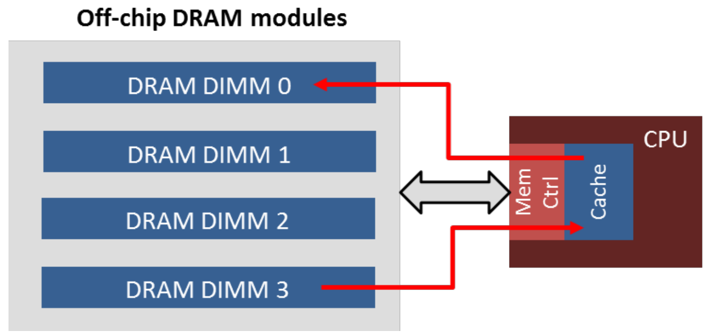


In-DRAM Reshape Accelerator

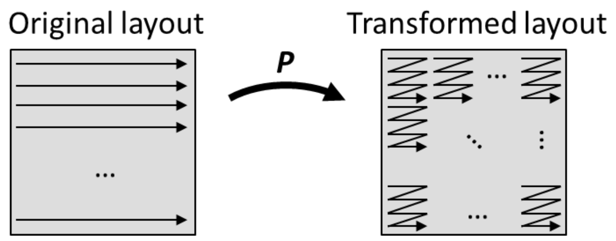
3DIC Reshape Stack



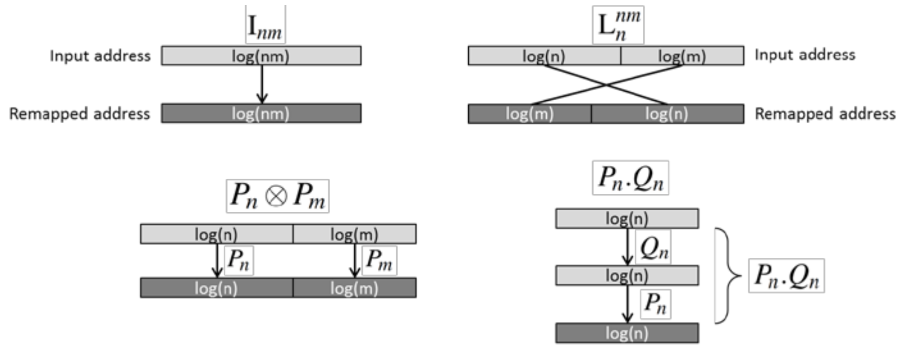
Traditional System



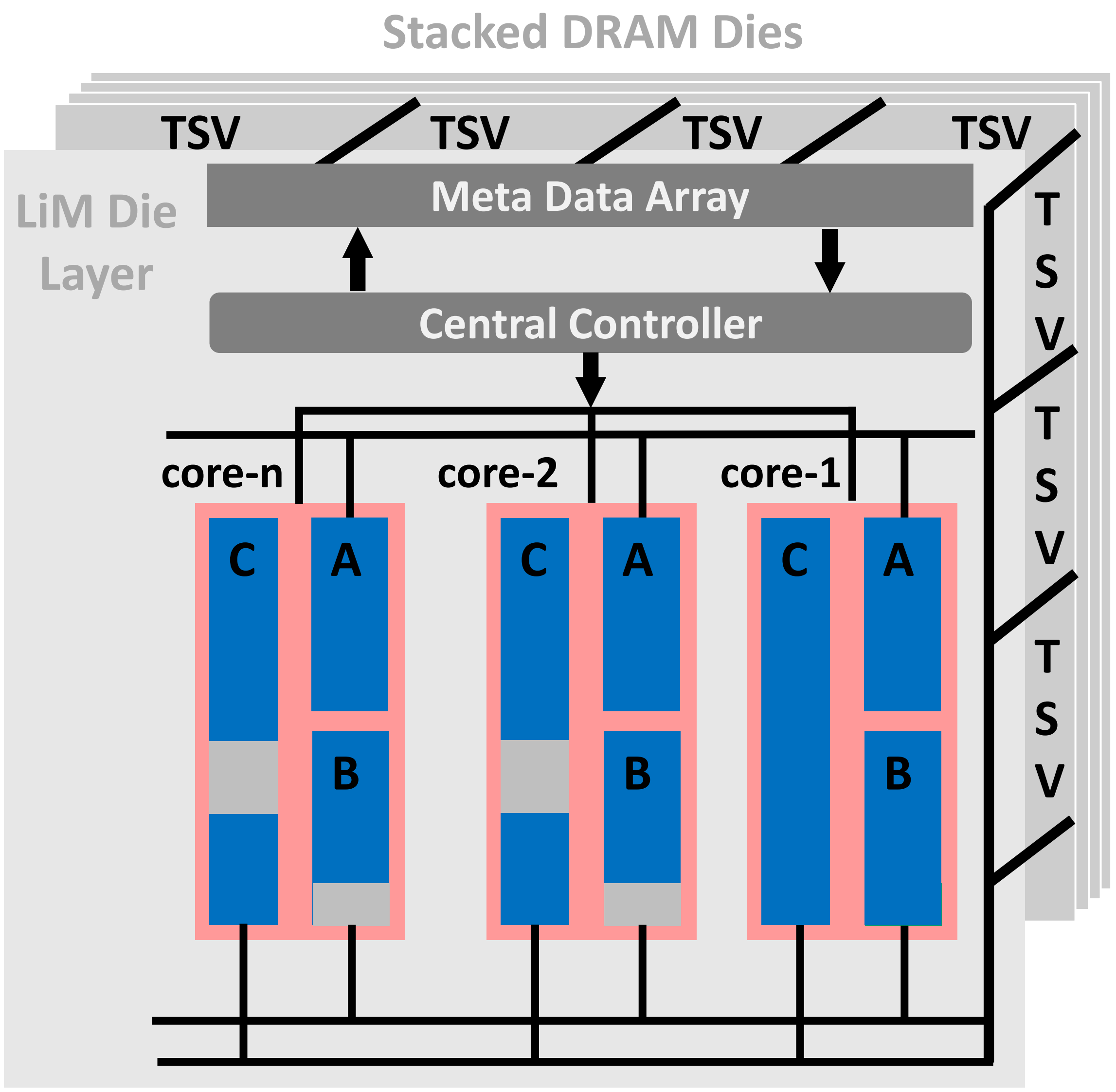
Reshape Operation



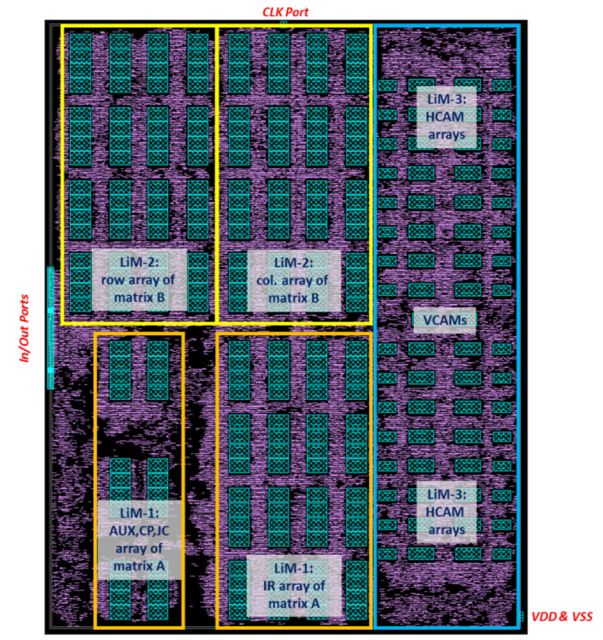
Abstraction: Bit Permutations



3DIC DRAM + SpGEMM Multicore Design



65nm test chip


CLK Port
In/Out Ports
LiM-3: HCAM arrays
LiM-2: row array of matrix B
LiM-2: col. array of matrix B
VCAMs
LiM-1: AUX,CP,JC array of matrix A
LiM-1: IR array of matrix A
LiM-3: HCAM arrays
VDD & VSS

- Taped out (65nm)
- Core Area: 1.003 x 1.292 mm^2
- Transistor count > 1M
- Frequency: 525MHz (SS @ICC)
- Power: 150mW (@DC)

Tape-out is funded under IARPA (PI Pileggi, Co-PIs Fedder, Franchetti, Piazza)

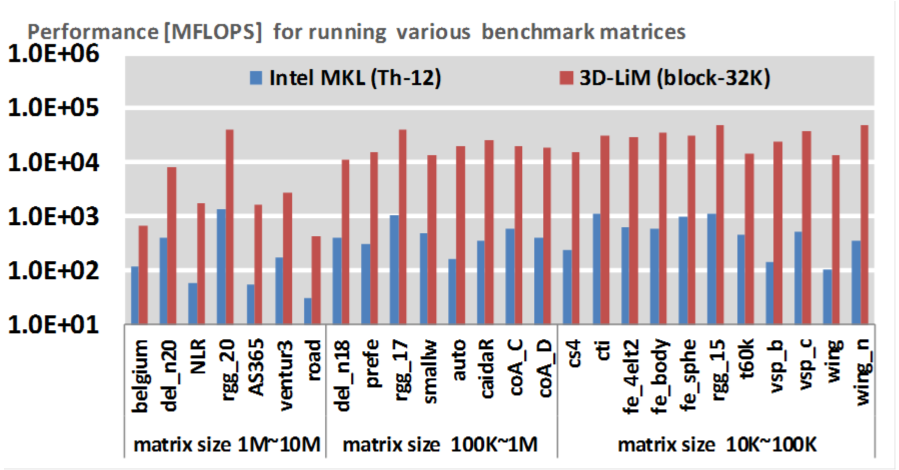
SpGEMM: Experimental Results

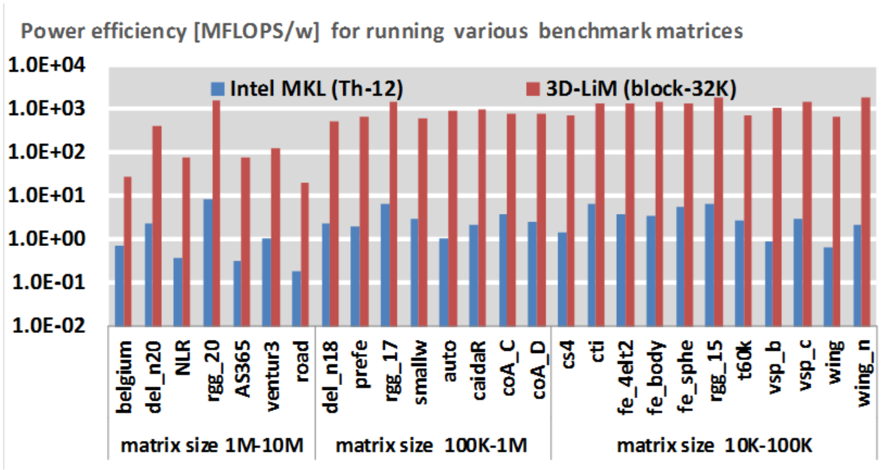
Intel Dual-CPU Intel Xeon E5-2430 system

- about 140W, 6 DRAM channels, 64 GB/s max, 6 DIMMs (48 chips), 210 GFLOPS peak
- Intel MKL 11.0 mkl_dcsrmultdcsr (unsorted):
  100 MFLOPS – 1 GFLOPS, 1 – 10 MFLOPS/W

Our 3DIC System

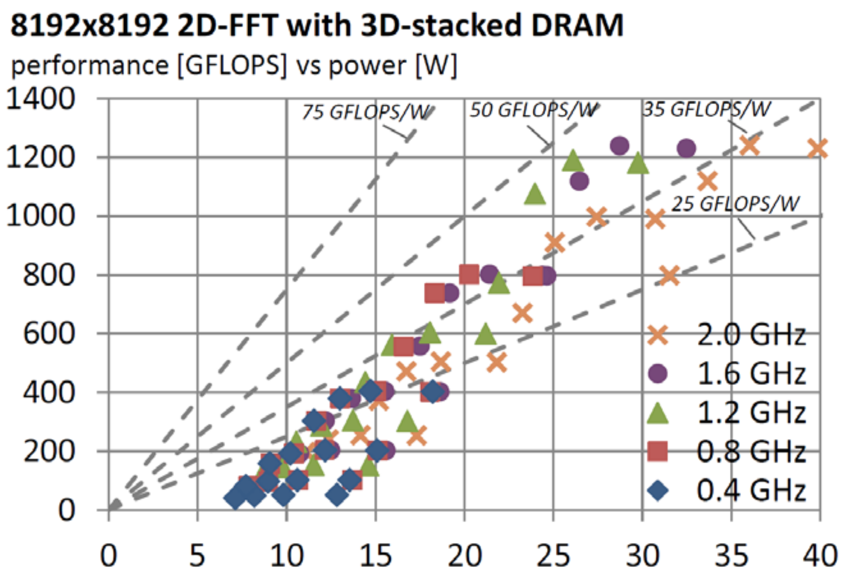
- 4 DRAM layers@ 2GBit, 1 logic layer, 32nm, 30 – 50 cores, 1GHz, 30 – 50 GFLOPS peak
- Performance: 10 – 50 GFLOPS @ 668GB/s with 1024 TSV
  Power efficiency: 1 GFLOPS/W @ 350GB/s with 512TSV or 8GB/s with 1024 TSV

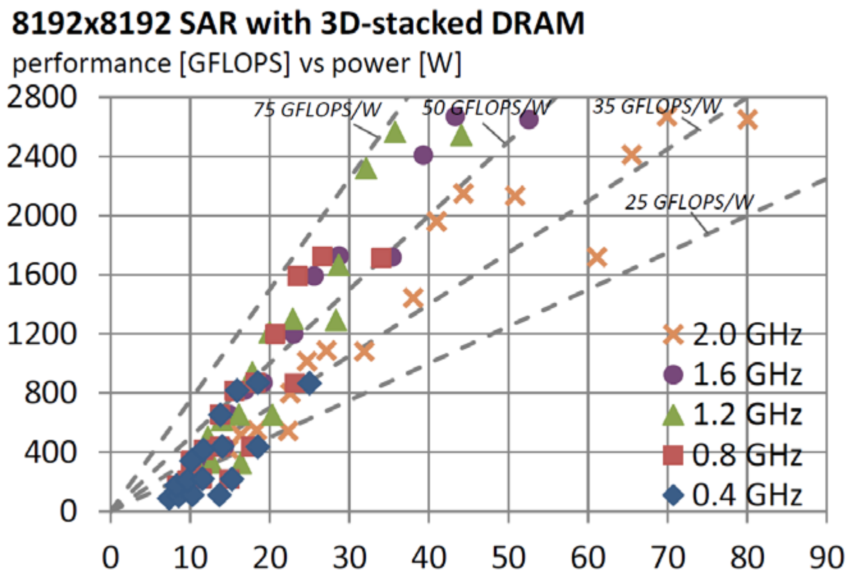




Matrices from University of Florida sparse matrix collection

3DIC for 2DFFT and PFA SAR





- **3DIC:** 8Gbit, 4-layer DRAM + 1-layer logic,
  16 banks/layer, 512 TSV/bank, 1KB pages, 32nm (320GB/s max BW)
- **2DFFT:** tiled FFT, tile size matches DRAM row buffers
- **PFA SAR:** Polar to rectangular local interpolation with logic-in-memory
  More flops for the same # of accesses

Results: DRAM-aware FFT Algorithms

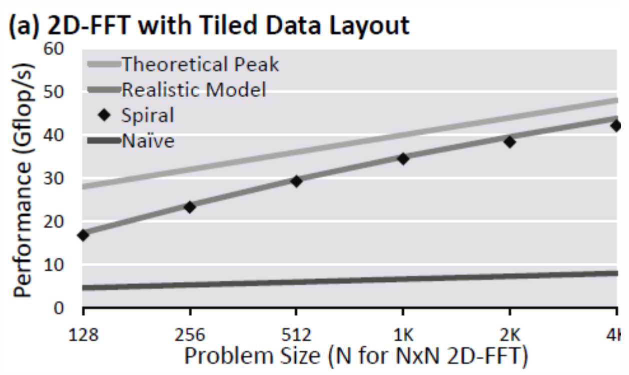
DRAM-aware FFT algorithms optimized by SPIRAL

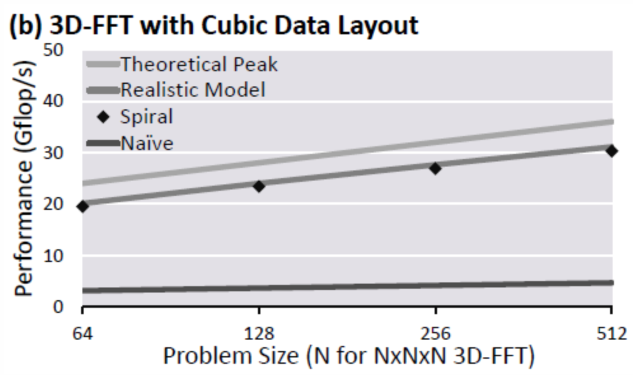
$$\mathrm{DFT}_{n\times n} = \left(\left(R_{t2}^{\mathrm{T}}R_{t3}^{\mathrm{T}}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes\mathrm{DFT}_n})R_{t3}R_{t2} \mid R_{t0}^{\mathrm{T}}R_{t1}^{\mathrm{T}}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes\mathrm{DFT}_n})R_{t1}R_{t0}\right)^{\overleftarrow{Q}}\right)^{\overrightarrow{P}}, \text{ where } P = Q^{-1} = \mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_k .$$

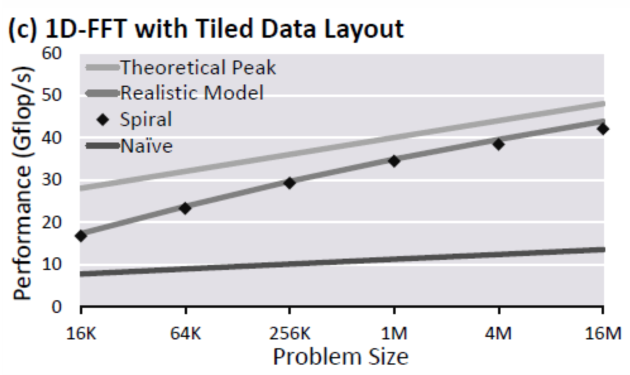
$$\mathrm{DFT}_{n\times n\times n} = \left(\left((\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes\mathrm{DFT}_n})^{R_{c2}} \mid (\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes\mathrm{DFT}_n})^{R_{c1}} \mid (\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes\mathrm{DFT}_n})^{R_{c0}}\right)^{\overleftarrow{Q}}\right)^{\overrightarrow{P}}, \; P = Q^{-1} = (\mathrm{I}_{n^2/k^2}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_{k^2})(\mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_k)$$

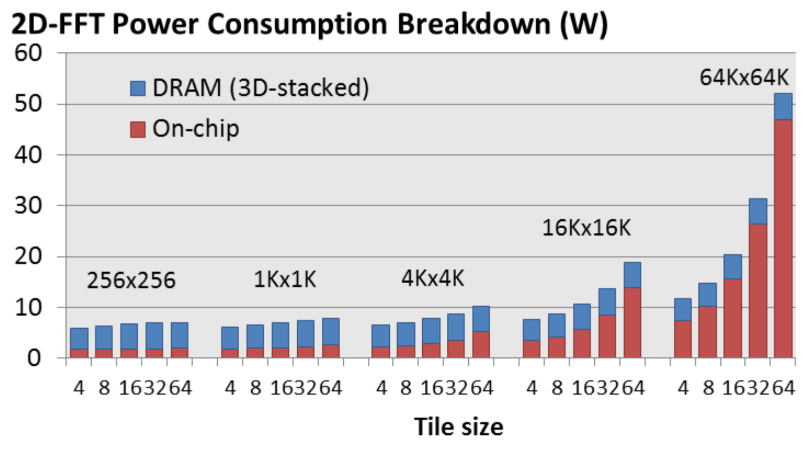
$$\mathrm{DFT}_{n^2} = \left(\left(R_{t2}^{\mathrm{T}}R_{t3}^{\mathrm{T}}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes\mathrm{DFT}_n})R_{t3}R_{t2} \mid R_{t0}^{\mathrm{T}}R_{t1}^{\mathrm{T}}\underline{\mathrm{D}_n^{n^2}}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes\mathrm{DFT}_n})R_{t3}R_{t2}\right)^{\overleftarrow{Q}}\right)^{\overrightarrow{P}}, \text{ where } P = Q^{-1} = \mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_k .$$

Performance model and power/performance simulation results

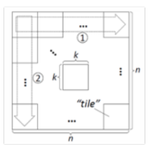






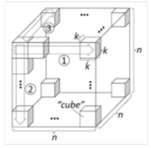


2D tiles



$$\begin{aligned}
A &\rightarrow (A^{\overleftarrow{Q}})^{\overrightarrow{P}}, \quad \text{where } Q = P^{-1}\\
AB &\rightarrow A|B\\
\mathrm{I}_n\otimes A_n &\rightarrow \mathrm{I}_{n^2}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes A_n})\mathrm{I}_{n^2}\\
A_n\otimes \mathrm{I}_n &\rightarrow \mathrm{L}_n^{n^2}(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{I}_k\otimes A_n})\mathrm{L}_n^{n^2}\\
\mathrm{I}_{n^2} &\rightarrow \underbrace{(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{L}_k^{n}\otimes\mathrm{I}_k})}_{R_{t1}}\underbrace{\overline{(\mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_k)}}_{R_{t0}}\\
\mathrm{L}_n^{n^2} &\rightarrow \underbrace{(\mathrm{I}_{n/k}\tilde{\otimes}\underline{\mathrm{L}_k^{nk}})}_{R_{t3}}\underbrace{\overline{(\mathrm{L}_{n/k}^{n^2/k}\otimes\mathrm{I}_k)}}_{R_{t2}}
\end{aligned}$$

3D bricks



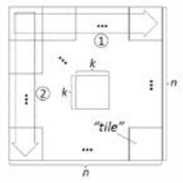
$$\begin{aligned}
\mathrm{I}_n\otimes\mathrm{I}_n\otimes A_n &\rightarrow \mathrm{I}_{n^3}(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes A_n})\mathrm{I}_{n^3}\\
\mathrm{I}_n\otimes A_n\otimes\mathrm{I}_n &\rightarrow (\mathrm{I}_n\otimes\mathrm{L}_n^{n^2})(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes A_n})(\mathrm{I}_n\otimes\mathrm{L}_n^{n^2})\\
A_n\otimes\mathrm{I}_n\otimes\mathrm{I}_n &\rightarrow \mathrm{L}_n^{n^3}(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{I}_{k^2}\otimes A_n})\mathrm{L}_{n^2}^{n^3}\\
\mathrm{I}_{n^3} &\rightarrow (\mathrm{I}_{n/k}\otimes\mathrm{L}_k^{n}\otimes\mathrm{I}_{nk})(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{L}_{k^2}^{nk}\otimes\mathrm{I}_k})\\
&\quad \overline{(\mathrm{I}_{n^2/k^2}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_{k^2})(\mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_k)} \triangleq R_{c0}\\
\mathrm{I}_n\otimes\mathrm{L}_n^{n^2} &\rightarrow (\mathrm{I}_{n/k}\otimes\mathrm{L}_k^{n}\otimes\mathrm{I}_{nk})(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{L}_{k^2}^{nk^2}(\mathrm{I}_{n/k}\otimes\mathrm{L}_k^{k^2}\otimes\mathrm{I}_k)})\\
&\quad \overline{(\mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n^2}\otimes\mathrm{I}_k)(\mathrm{I}_{n/k}\otimes\mathrm{L}_{n/k}^{n}\otimes\mathrm{I}_{nk})} \triangleq R_{c2}\\
\mathrm{L}_{n^2}^{n^3} &\rightarrow (\mathrm{I}_{n/k}\otimes\mathrm{L}_k^{n}\otimes\mathrm{I}_{nk})(\mathrm{I}_{n^2/k^2}\tilde{\otimes}\underline{\mathrm{L}_{k^2}^{nk^2}})\\
&\quad \overline{(\mathrm{L}_{n^2/k^2}^{n^3/k}\otimes\mathrm{I}_k)(\mathrm{I}_n\otimes\mathrm{L}_k^{n}\otimes\mathrm{I}_n)} \triangleq R_{c1}
\end{aligned}$$

Near Memory Computing in DARPA PERFECT

HW/SW Formalization

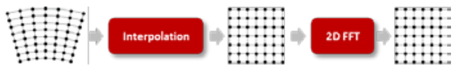
SPIRAL System

Hardware Synthesis

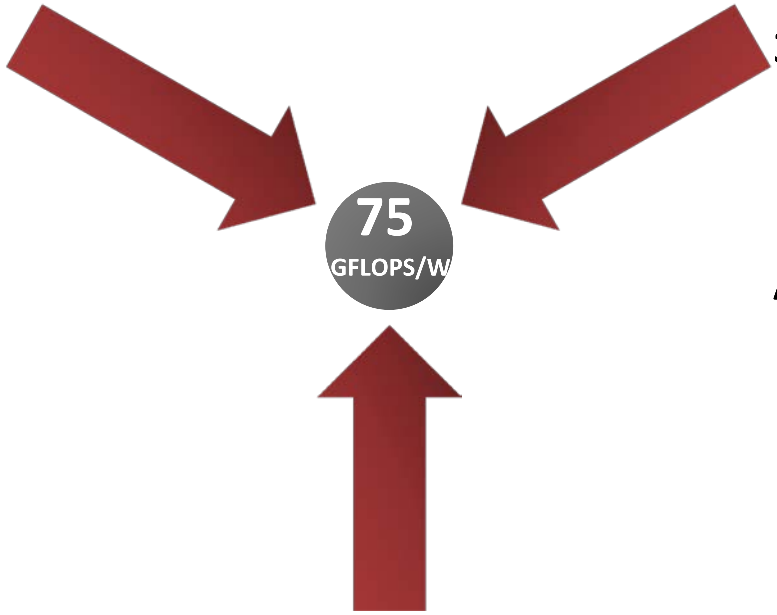


$$\begin{aligned}
A &\to (A^{\overleftarrow{Q}})^{\overrightarrow{P}}, \quad \text{where } Q = P^{-1}\\
AB &\to A|B\\
I_n \otimes A_n &\to I_{n^2}(I_{n/k} \tilde{\otimes} \underline{I_k \otimes A_n}) I_{n^2}\\
A_n \otimes I_n &\to L^{n^2}_n (I_{n/k} \tilde{\otimes} \underline{I_k \otimes A_n}) L^{n^2}_n\\
I_{n^2} &\to \underbrace{(I_{n/k} \tilde{\otimes} \underline{L^n_k \otimes I_k})}_{R_{i1}} \underbrace{\overline{(I_{n/k} \otimes L^n_{n/k} \otimes I_k)}}_{R_{i0}}\\
L^{n^2}_n &\to \underbrace{(I_{n/k} \tilde{\otimes} \underline{L^{nk}_k})}_{R_{i3}} \underbrace{\overline{(L^{n^2/k}_{n/k} \otimes I_k)}}_{R_{i2}}
\end{aligned}$$

Software Synthesis



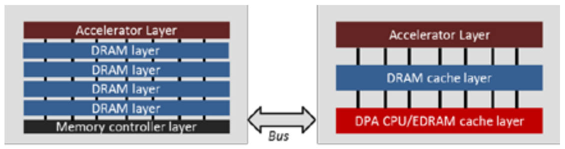
$$\begin{aligned}
\mathsf{SAR}_{k\times m\to n\times n} &\to \mathsf{DFT}_{n\times n} \circ \mathsf{Interp}_{k\times m\to n\times n}\\
\mathsf{DFT}_{n\times n} &\to (\mathsf{DFT}_n \otimes \mathrm{I}_n) \circ (\mathrm{I}_n \otimes \mathsf{DFT}_n)\\
\mathsf{Interp}_{k\times m\to n\times n} &\to (\mathsf{Interp}_{k\to n} \otimes_i \mathrm{I}_n) \circ (\mathrm{I}_k \otimes_i \mathsf{Interp}_{m\to n})\\
\mathsf{Interp}_{r\to s} &\to \left(\bigoplus_{i=0}^{n-2} \mathsf{InterpSeg}_k\right) \oplus \mathsf{InterpSegPruned}_{k,\ell}\\
\mathsf{InterpSeg}_k &\to \mathrm{G}_f^{u\cdot n\to k} \circ \mathsf{iPrunedDFT}_{n\to u\cdot n} \circ \left(\frac{1}{n}\right) \circ \mathsf{DFT}_n
\end{aligned}$$



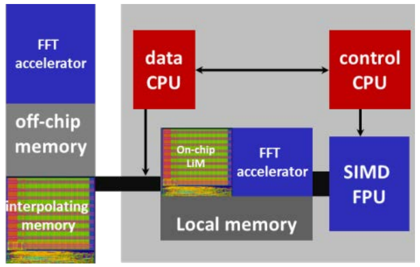
Algorithm+HW Design

Memory-Side Accelerators

3DIC System



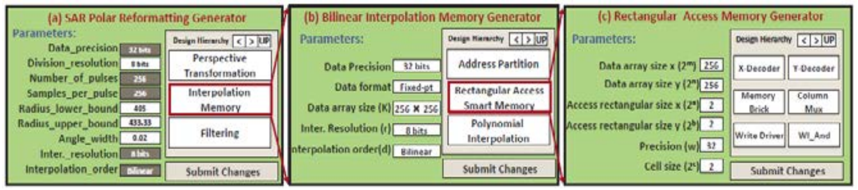
Accelerator Architecture



Low Power Accelerator Cores

Logic-in-memory for sub-22nm CMOS

Design Tools and Simulation



(a) SAR Polar Reformatting Generator

Parameters:

Data_precision	32 bits
Division_resolution	8 bits
Number_of_pulses	256
Samples_per_pulse	256
Radius_lower_bound	405
Radius_upper_bound	433.33
Angle_width	0.02
Inter._resolution	8 bits
Interpolation_order	Bilinear

(b) Bilinear Interpolation Memory Generator

Parameters:

Data Precision	32 bits
Data format	Fixed-pt
Data array size (K)	256 ✖ 256
Inter. Resolution (r)	8 bits
nterpolation order(d)	Bilinear

(c) Rectangular Access Memory Generator

Parameters:

Data array size x (2^m)	256
Data array size y (2^n)	256
Access rectangular size x (2^a)	2
Access rectangular size y (2^b)	2
Precision (w)	32
Cell size (2^c)	2