

Experience and results porting HPEC Benchmarks to MONARCH

Lloyd Lewins & Kenneth Prager

Raytheon Company

2000 E. El Segundo Blvd, El Segundo, CA 90245

llewins@raytheon.com, keprager@raytheon.com

**High Performance Embedded Computing (HPEC)
Workshop**

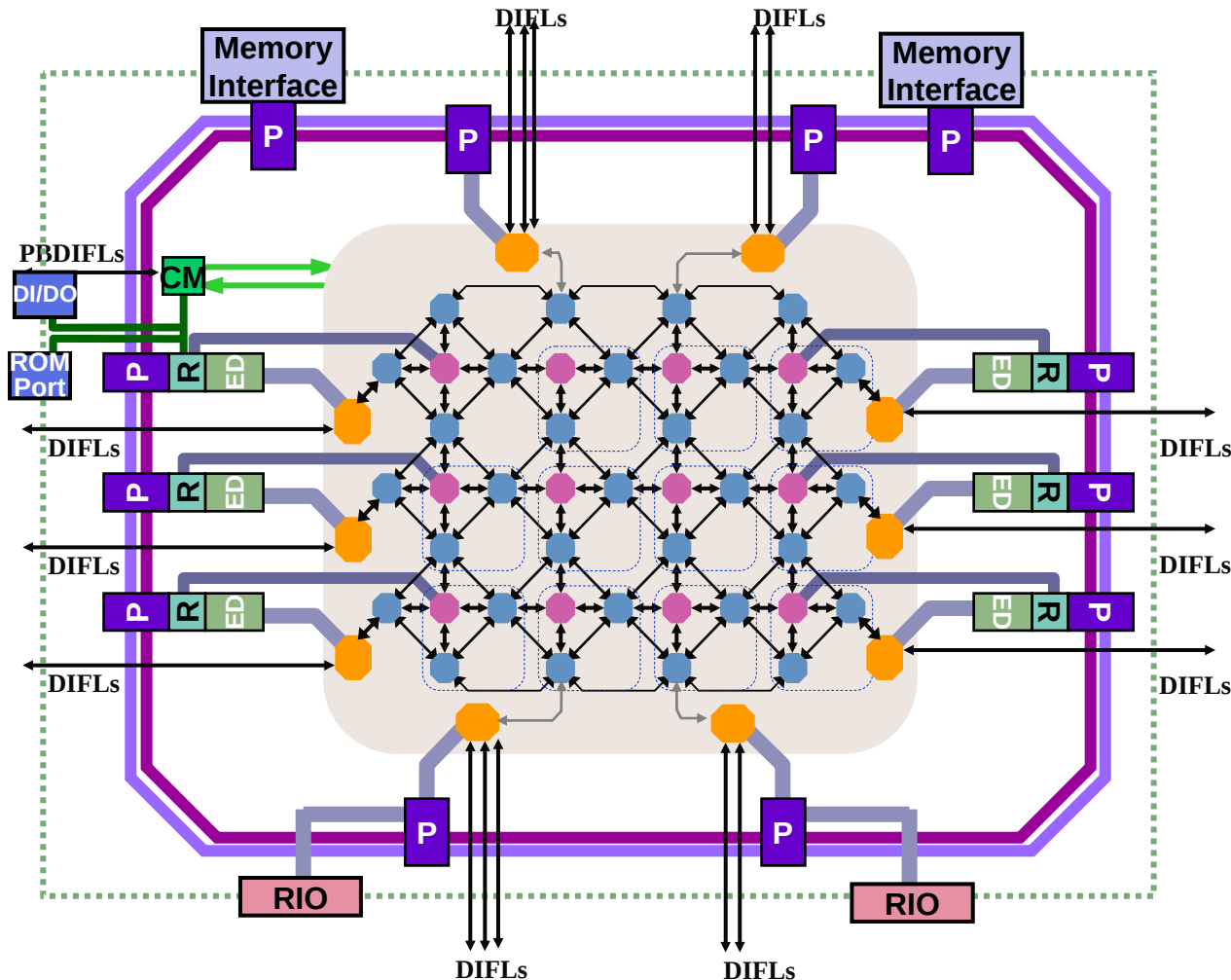
23–25 September 2008

(A) Approved for public release; distribution is unlimited.

Overview of HPEC Benchmarks

- Provides a means to quantitatively evaluate high performance embedded computing (HPEC) systems
- Addresses important operations across a broad range of DoD signal and image processing applications
 - Finite Impulse Response (FIR) Filter
 - QR Factorization
 - Singular Value Decomposition
 - Pattern Matching
 - Corner turn etc
- Documentation, Uniprocessor C-code, MATLAB, Sizes
- <http://www.ll.mit.edu/HPECchallenge/index.html>

Overview of MONARCH



- ◆ 6 RISC Processors
- ◆ 12 MBytes on-chip DRAM
- ◆ 2 DDR2 External Memory Interfaces (8 GB/s BW)
- ◆ Flash Port (32 MB)
- ◆ 2 Serial RapidIO Ports (1.25 GB/s each)
- ◆ 16 IFL ports (2.6 GB/s each)
- ◆ On-chip Ring 40 GB/s
- ◆ Reconfigurable Array: FPCA (64 GFLOPS)

Benchmark Selection

- Transpose (corner-turn)
 - 50x5000 and 750x5000
 - Transpose to/from External DRAM
- Constant False Alarm Detection (CFAR)
 - 16x64x24, 48x3500x128, 48x1909x64 and 16x9900x16
 - Few ops – bandwidth limited.
 - Larger datasets in External DRAM – smaller in EDRAM
- QR Factorization
 - 500x100, 180x60, and 150x150
 - Givens Rotation (more complex)
 - Many 2x2 matrix multiplies (but simple)
- Note: results for FIR and FFT previously reported

MONARCH Mapping Issues

■ Bandwidth Limitations

– External DRAM (DDR2)

- 4.7 Gbyte/s peak per port (64 bits @ 333MHz DDR + overhead)
- Only one port populated on test board

– Implementation Issues

- EDRAM bank conflict bug – no simultaneous read/write
- PBuf to Node-Bus arbitration – unload one word every 3 clocks (cuts 10.6 Gbyte/s PIRX bandwidth down to 3.6 Gbyte/s).
- DDR2 latency versus MMBT pipeline depth – limits reads to 3.8 Gbyte/s.

■ Partitioning

■ Algorithm Selection

- “Fast” Givens versus “regular” Givens

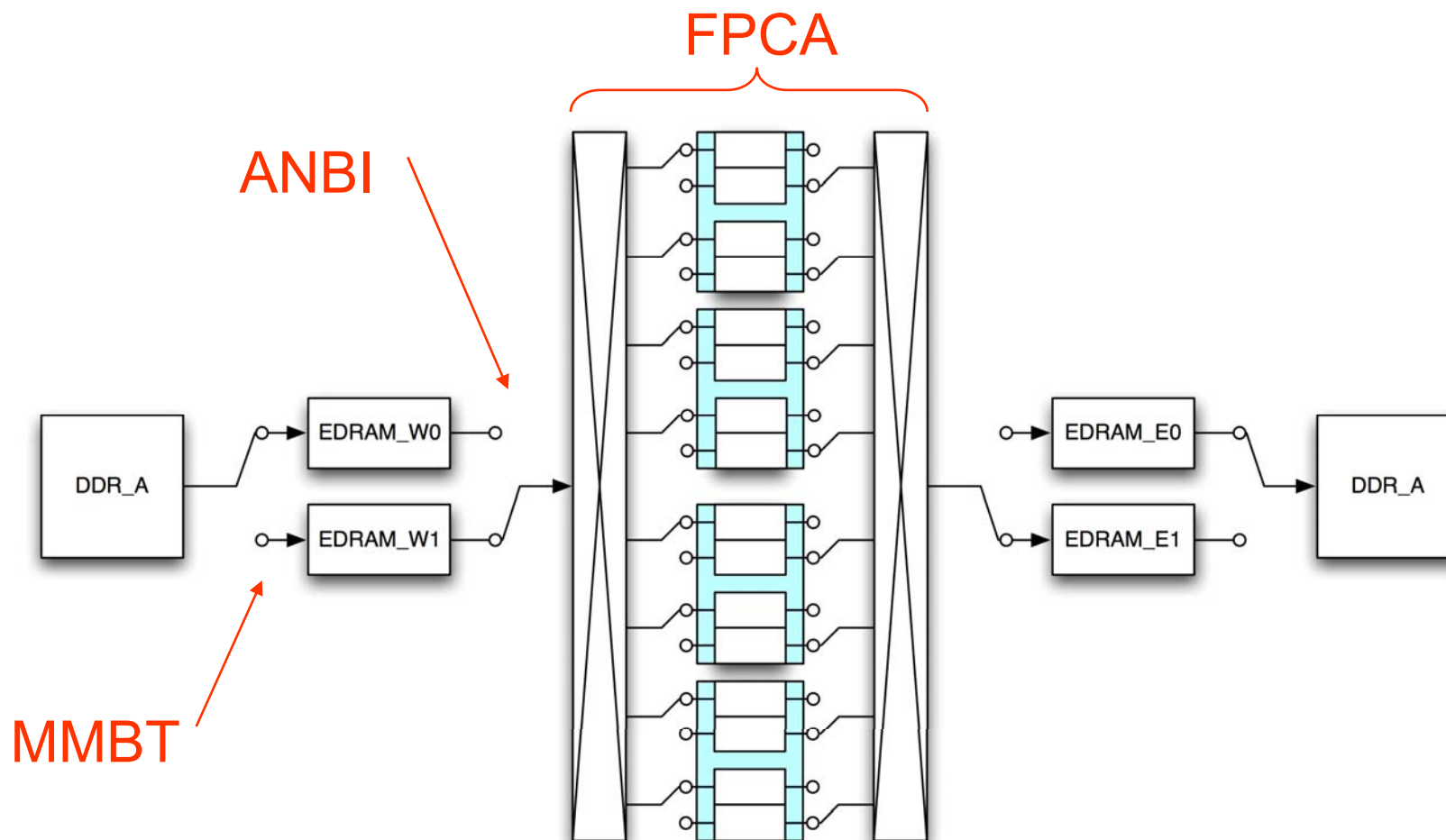
■ Reciprocal/Square Root

- Synthesize using Newton-Raphson

Corner Turn Benchmark

- Hierarchical Block Transpose
 - FPCA handles 32x8 inner block (uses 16 MEM elements)
 - EDRAM contains 32x2528 blocks – ANBI streams into 32x8 blocks
 - MMBT transfers 32x2528 blocks to/from DDR2
- Alignment Issues
 - MMBT/DDR2 interactions require transferring 32 words for peak performance
 - Total transpose was 768x5056 (3.5% larger)
- Performance Issues
 - Single FPCA Transpose engine limits bandwidth to 1.3 Gbyte/s
 - Elimination of bank conflict bug and two DDR2 ports would allow three transpose engines (3.6 Gbyte/s) – limited by PBuf/Node-Bus arbitration

Corner Turn Implementation



FPCA – Field Programmable Computer Array; ANBI – Array Node Bus Interface; MMBT – Memory Block Transfer;
DDR_A – Double Data Rate DRAM interface A; EDRAM – Embedded DRAM

Corner Turn Results

- Measured performance and predicted performance if second DDR2 bank available:

M	N	Setup Time	Current Chip: 1 DDR2			Current Chip: 2 DDR2		
			Predicted Bandwidth	Measured Bandwidth	% Error	Predicted Bandwidth	Derated Bandwidth	% Error
50	5000	29.0E-6	1.3E+9	640.7E+6	-51.9%	1.5E+9	732.2E+6	-51.9%
750	5000	28.8E-6	1.3E+9	1.1E+9	-17.7%	1.5E+9	1.3E+9	-17.7%

- Predicted performance in the absence of the bank conflict bug

M	N	Setup Time	Updated Chip (No BCB): 1 DDR2			Updated Chip (No BCB): 2 DDR2		
			Predicted Bandwidth	Derated Bandwidth	% Error	Predicted Bandwidth	Derated Bandwidth	% Error
50	5000	29.0E-6	1.8E+9	854.3E+6	-51.9%	3.6E+9	1.7E+9	-51.9%
750	5000	28.8E-6	1.8E+9	1.5E+9	-17.7%	3.6E+9	2.9E+9	-17.7%

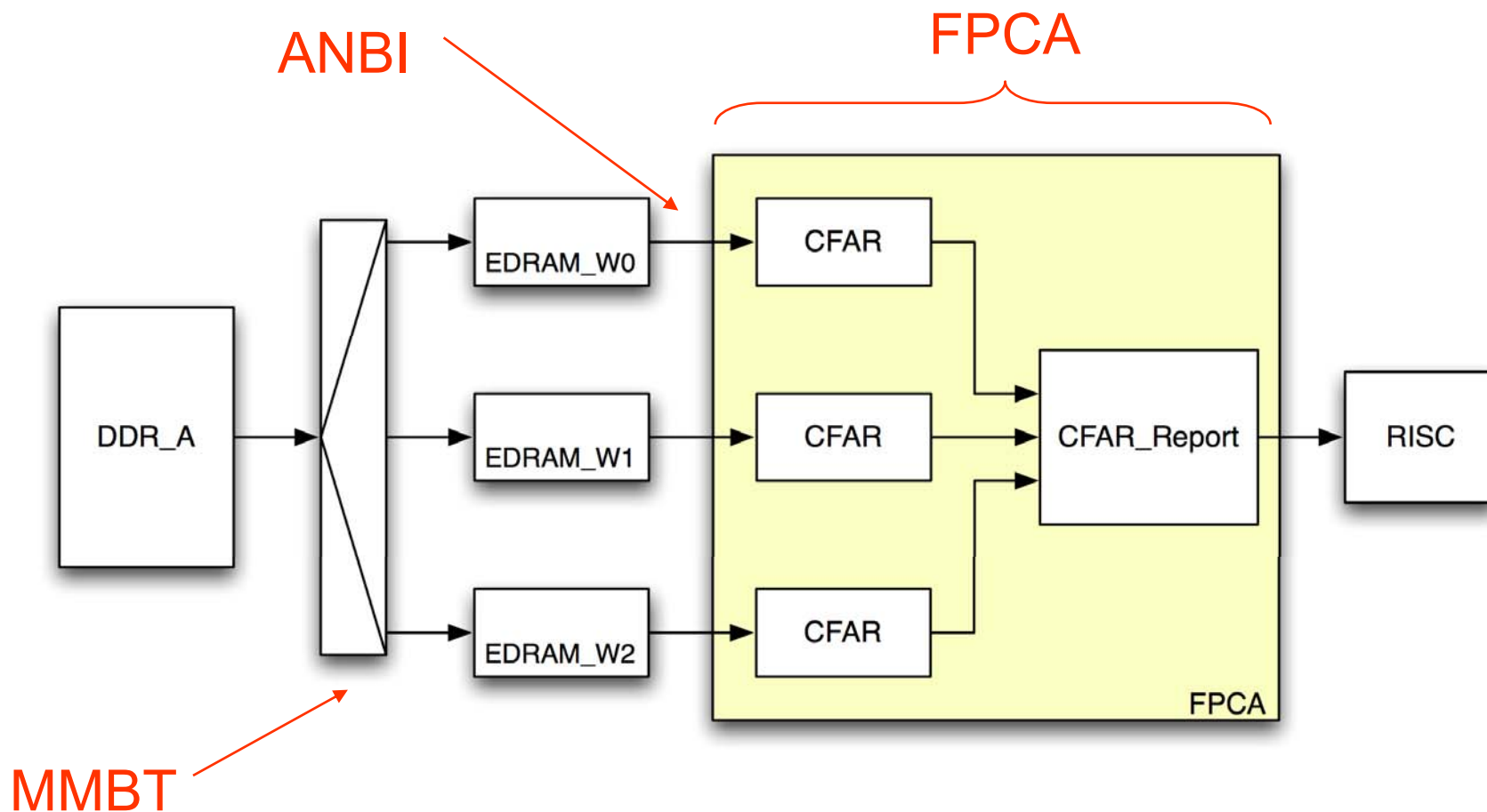
Note: this is end-to-end bandwidth – achieved memory bandwidth of 2X
Bandwidth is in Bytes per Second

DDR2 – Double Data Rate DRAM interface
BCB – Band Conflict Bug (EDRAM)

Constant False Alarm Rate Benchmark

- Multiple CFAR engines implemented in FPCA
 - Limited by number of EDRAMs to feed them
- Smaller datasets stored in EDRAM
 - Six CFAR engines – 14 GFLOPS
- Larger datasets stored in DDR2
 - Three CFAR engines (because of bank conflict bug)
 - Further limited by DDR2 bandwidth – 6.2 GFLOPS (12.4 GFLOPS with two DDR ports)

CFAR Implementation



FPCA – Field Programmable Computer Array; ANBI – Array Node Bus Interface; MMBT – Memory Block Transfer;
DDR_A – Double Data Rate DRAM interface A; EDAM – Embedded DRAM

Constant False Alarm Rate Results

- Measured performance and predicted performance if second DDR2 bank available:

N_bm	N_rg	N_dop	Setup Time	Current Chip: 1 DDR2			Current Chip: 2 DDR2		
				Predicted FLOP/S	Measured FLOP/S	% Error	Predicted FLOP/S	Derated FLOP/S	% Error
16	64	24	58.3E-6	14.0E+9	6.4E+9	-54.6%	14.0E+9	6.4E+9	-54.6%
48	3500	128	239.0E-6	5.1E+9	4.8E+9	-5.2%	10.2E+9	9.6E+9	-5.2%
48	1909	64	238.7E-6	5.1E+9	4.6E+9	-8.9%	10.2E+9	9.3E+9	-8.9%
16	9900	16	63.7E-6	14.0E+9	12.3E+9	-12.2%	14.0E+9	12.3E+9	-12.2%

- Predicted performance in the absence of the bank conflict bug:

N_bm	N_rg	N_dop	Setup Time	Updated Chip (No BCB): 1 DDR2			Updated Chip (No BCB): 2 DDR2		
				Predicted FLOP/S	Derated FLOP/S	% Error	Predicted FLOP/S	Derated FLOP/S	% Error
16	64	24	58.3E-6	14.0E+9	6.4E+9	-54.6%	14.0E+9	6.4E+9	-54.6%
48	3500	128	239.0E-6	6.2E+9	5.9E+9	-5.2%	12.4E+9	11.8E+9	-5.2%
48	1909	64	238.7E-6	6.2E+9	5.7E+9	-8.9%	12.4E+9	11.3E+9	-8.9%
16	9900	16	63.7E-6	14.0E+9	12.3E+9	-12.2%	14.0E+9	12.3E+9	-12.2%

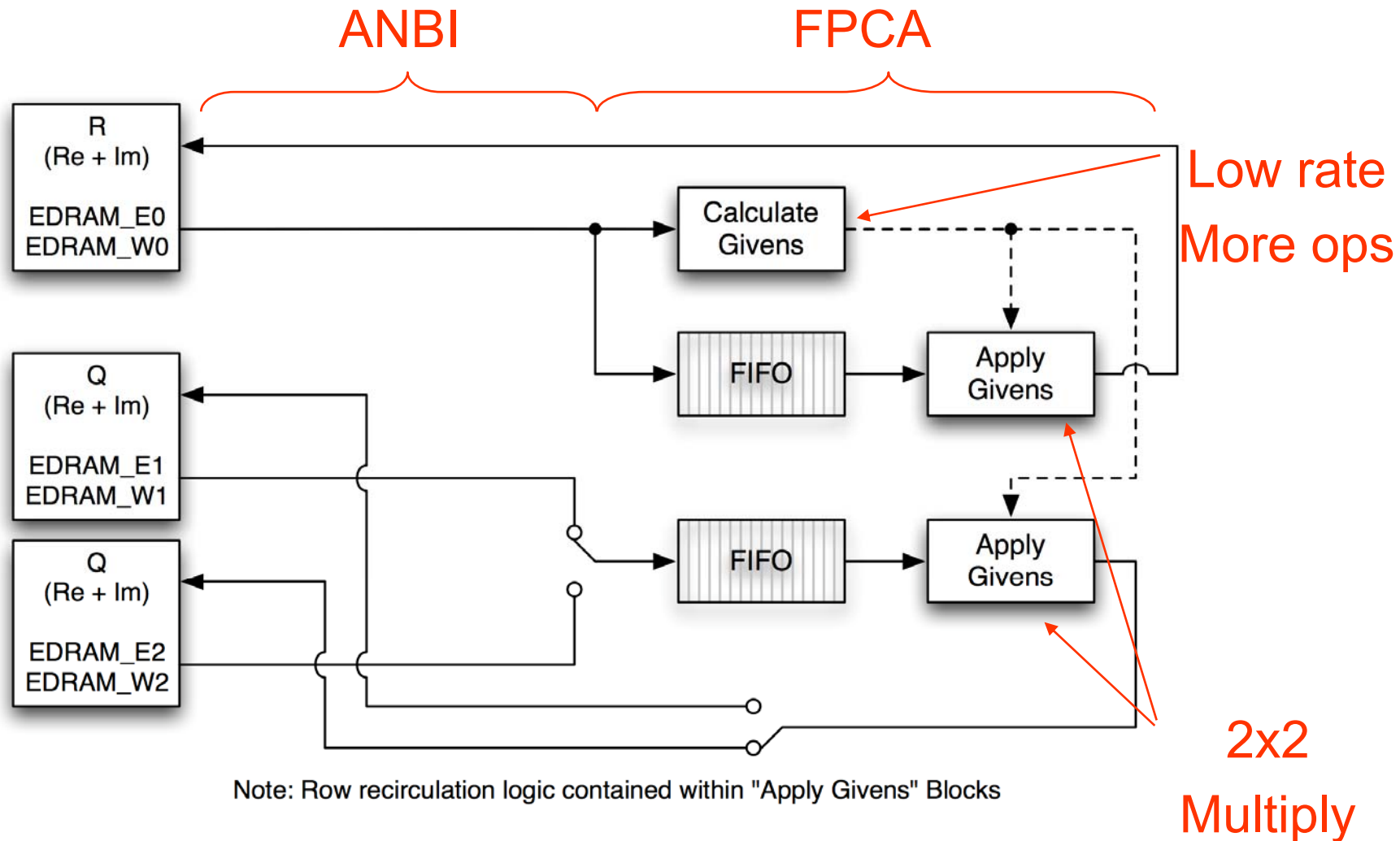
DDR2 – Double Data Rate DRAM interface

BCB – Band Conflict Bug (EDRAM)

QR Factorization Benchmark

- Single QR Engine implemented in FPCA
 - Uses high percentage of resources
 - Multiple streams to/from memory
- Performance limited by bandwidth to EDRAM
- Classic “Fast Givens” requires even more streams to/from EDRAM
 - Issue is not FLOPS, but Bandwidth
- Calculating Givens rotation requires square-root and reciprocal.
 - Implemented in FPCA using Newton-Raphson.

QR Factorization Implementation



FPCA – Field Programmable Computer Array; ANBI – Array Node Bus Interface; EDRAM – Embedded DRAM

QR Factorization Results

■ Measured performance:

M	N	Setup Time	Current Chip		
			Predicted FLOP/S	Measured FLOP/S	% Error
500	100	65.5E-6	5.9E+9	5.7E+9	-3.8%
180	60	65.0E-6	6.3E+9	5.8E+9	-7.1%
150	150	65.0E-6	6.6E+9	4.9E+9	-25.8%

■ Predicted performance in the absence of the bank conflict bug:

M	N	Setup Time	Updated Chip: No BCB		
			Predicted FLOP/S	Derated FLOP/S	% Error
500	100	65.5E-6	11.8E+9	11.3E+9	-3.8%
180	60	65.0E-6	12.5E+9	11.6E+9	-7.1%
150	150	65.0E-6	13.2E+9	9.8E+9	-25.8%

BCB – Band Conflict Bug (EDRAM)

Reciprocal/Square Root

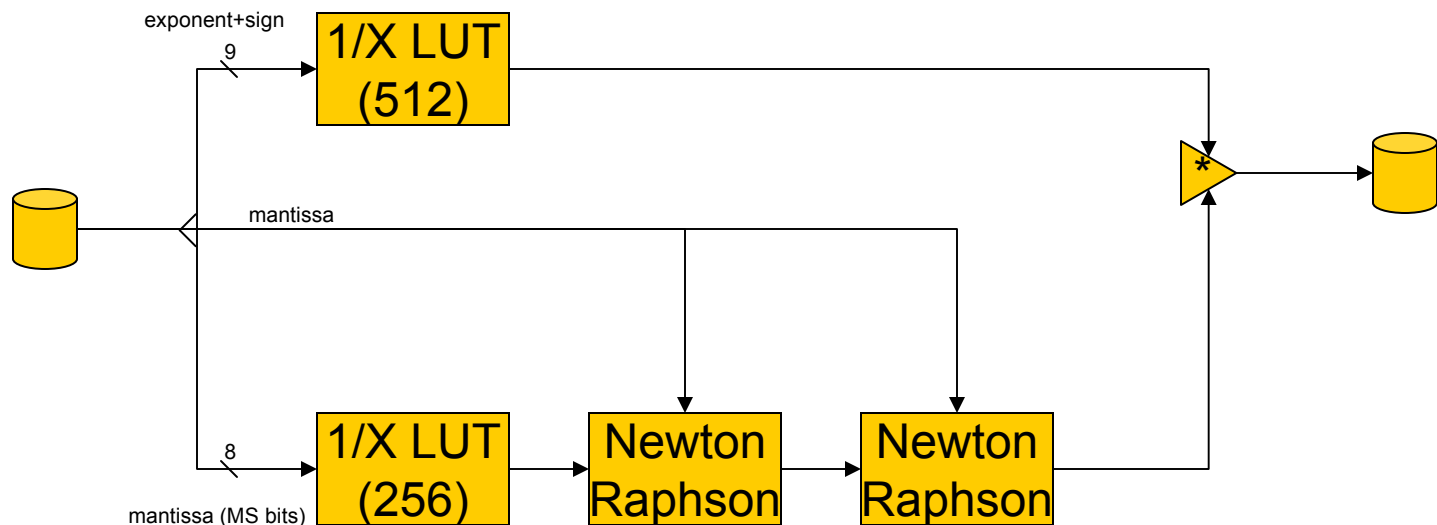
- FPCA doesn't support division or square-root directly
- Number of approaches considered, including CORDIC
- Newton-Raphson works surprisingly well, even for floating point numbers
 - Use a few small lookup tables
 - Integer arithmetic to extract exponent and mantissa
 - Floating point arithmetic to iterate estimate
 - Fully pipelined

Reciprocal Calculation (Math)

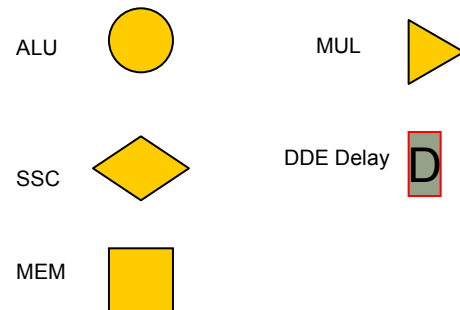
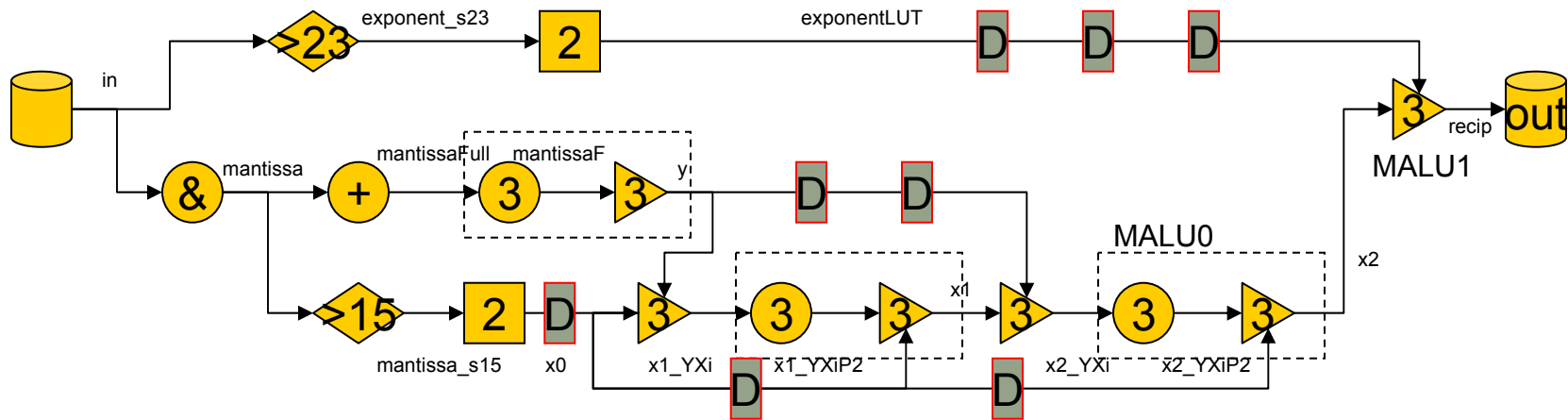
- Newton Raphson: to solve $1/y$, given an estimate of $1/y$ (x_i), a better estimate of $1/y$ (x_{i+1}) is given by:

$$x_{i+1} = x_i \cdot (2 - y \cdot x_i)$$

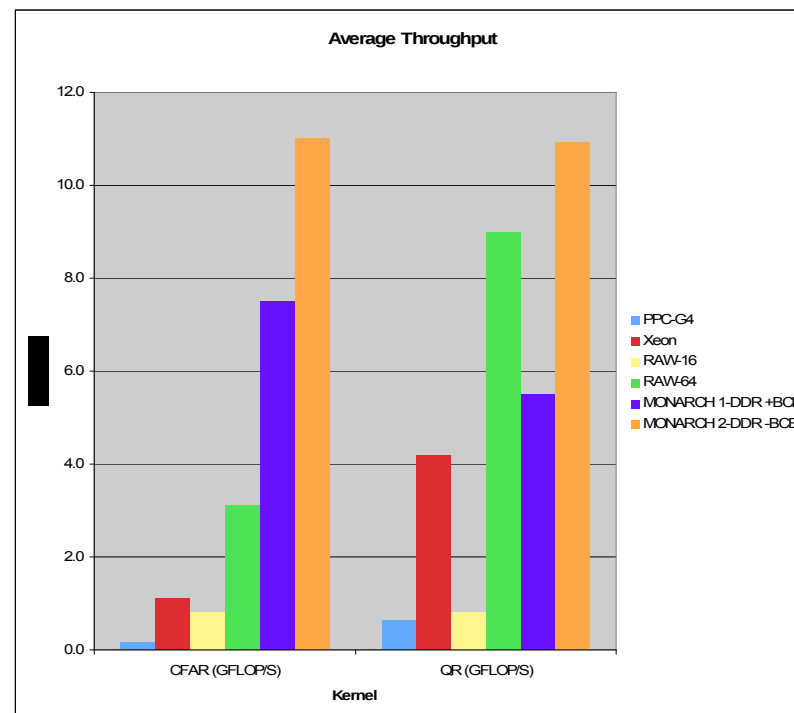
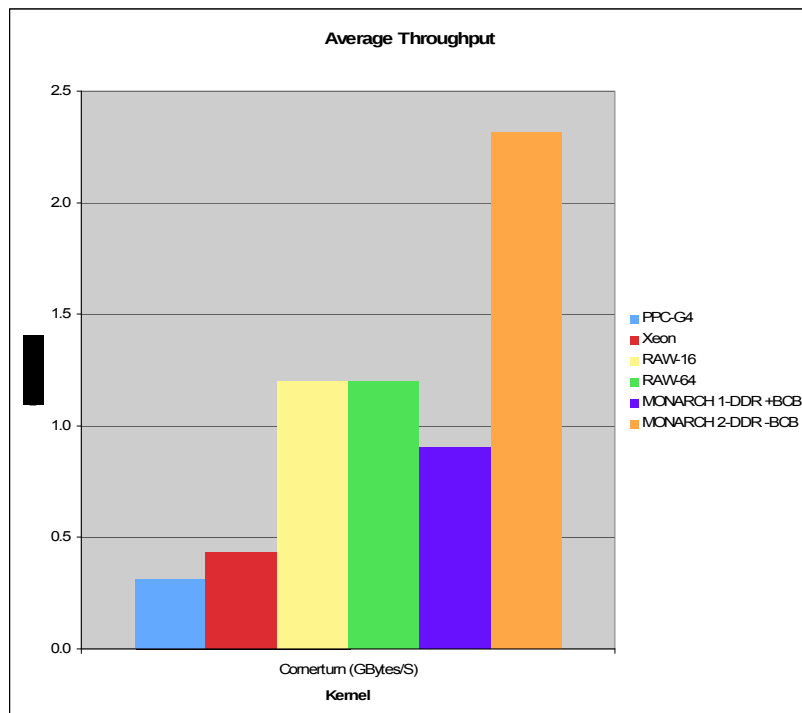
- Split the number into exponent (plus sign), and mantissa. Use LUT to calculate reciprocal of exponent, and a second LUT to estimate the reciprocal of the mantissa. Use Newton Raphson twice to refine the reciprocal of the mantissa (getting more than 23 bits) and finally multiply the resulting mantissa and exponent.



Reciprocal Calculation (Implementation)



Comparison to other Architectures



Benchmark (Units)	PPC-G4	Xeon	RAW-16	RAW-64	MONARCH 1-DDR +BCB	MONARCH 2-DDR -BCB
Cornertum (GBytes/S)	0.3	0.4	1.2	1.2	0.9	2.3

Benchmark (Units)	PPC-G4	Xeon	RAW-16	RAW-64	MONARCH 1-DDR +BCB	MONARCH 2-DDR -BCB
CFAR (GFLOP/S)	0.2	1.1	0.8	3.1	7.5	11.0
QR (GFLOP/S)	0.6	4.2	0.8	9.0	5.5	10.9

RAW-64 performance projected

Conclusion

- Several interesting HPEC benchmarks successfully implemented on MONARCH
- MONARCH performance very competitive with other published HPEC results
- Benchmarks all bandwidth limited
 - Partitioning focuses on optimizing data movement
 - Buffer data in EDRAM to ensure sequential DDR accesses
 - Select algorithm which is “bandwidth friendly”
 - “It’s the data movement stupid!”
- Reciprocal/square root readily synthesized from existing FPCA resources
 - Demonstrates flexibility of FPCA
- Working around errata of current chip added challenge!

This work was supported by the NRO