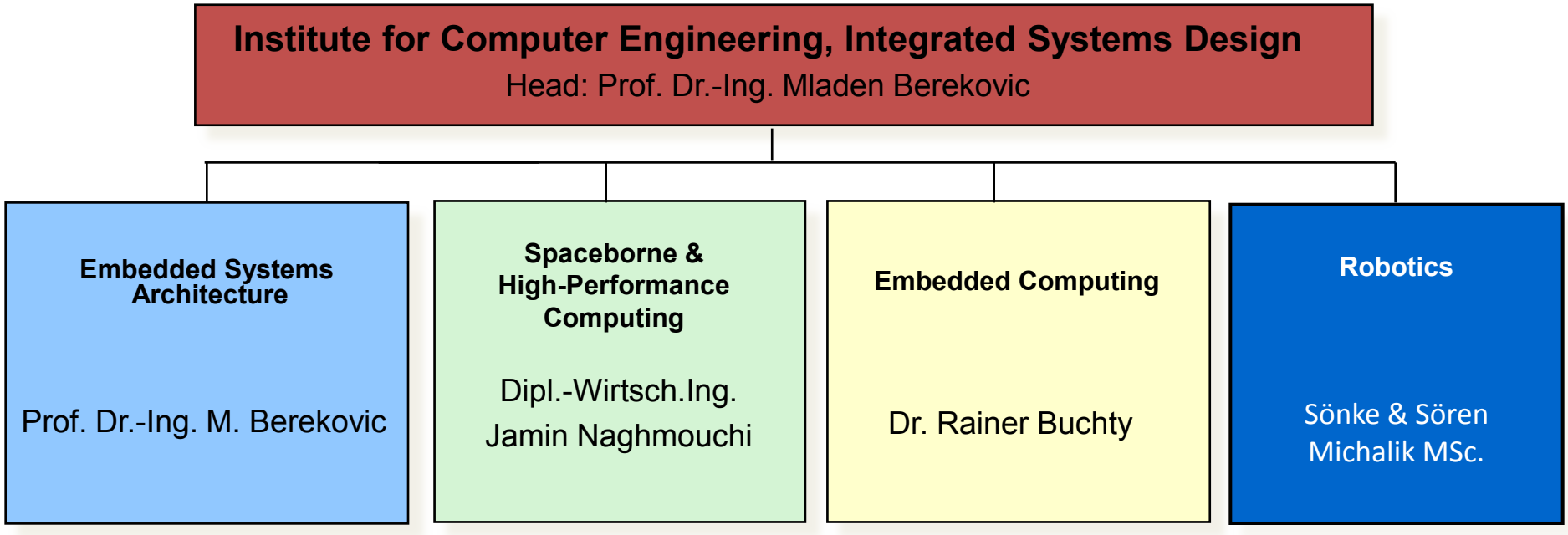
MacSpace
-
High Performance DSP for onboard image processing

Jamin Naghmouchi(1) , Sören Michalik(1), Rolf Scheiber(2), Adreas Reigber(2), Peleg Aviely(3), Ran Ginosar(3), Ole Bischoff(4), Hagay Gellis(5), Mladen Berekovic(1)

(1) TU Braunschweig, Pockelsstrasse 14, Braunschweig, 38106, Germany, naghmouchi@c3e.cs.tu-bs.de
(2) DLR - Microwaves and Radar Institute - SAR Technology Department, 82230 Wessling, Germany
(3) Ramon Chips Ltd, 5 HaCarmel Street, Yokneam, 2069201, Israel
(4) DSI GmbH, Otto-Lilienthal-Strasse 1, Bremen, 28199, Germany
(5) CEVA Inc., 1174 Castro Street, Suite 210, Mountain View, CA 94040, USA

Chair for Chip Design for Embedded Computing @ TU Braunschweig



Motivation 1/2

The evolution of the Earth Observation mission is driven by the development of new processing paradigms to facilitate:

- Data Downlink
- Data Handling
- Storage

Next generation planetary observation satellites will generate a great amount of data at a very high data rate, for both radar based and optical core applications.

Motivation 2/2

Currently employed applications such as e.g. FFT processing and BAQ compression on SAR satellites that usually do not change during the life-time of a satellite and therefore are mostly realized in hardware (e.g. FPGA accelerators).

More modern applications - due to longer development time and relatively high development costs - can't be implemented on special purpose hardware accelerators economically!

We have detected the need for a platform that allows:

- Flexibility for space application developers and mission planners
- Still a very high performance comparable to pure hardware implementations

The role of TU Braunschweig in MacSpace

Task	Benchmarking
Lead	TUBS
Team	DSI, RC, TAS-I

Value Proposition

- Creating a framework for performance tests/measurements
- Measuring performance on FPGA
- Compare actual performance to Simulator based estimates
- Perform “Real Application Benchmarks”

Technical Challenge

- Create software benchmarks
- Integrate and run benchmarks on Simulator
- Integrate Benchmarks for FPGA
- Get the memory footprint
- Measure integer arithmetic performance
- Measure floating point arithmetic performance
- Estimate performance of the real ASIC

Jobs to perform

1) Benchmarks to be developed in Software for Simulator and Emulator

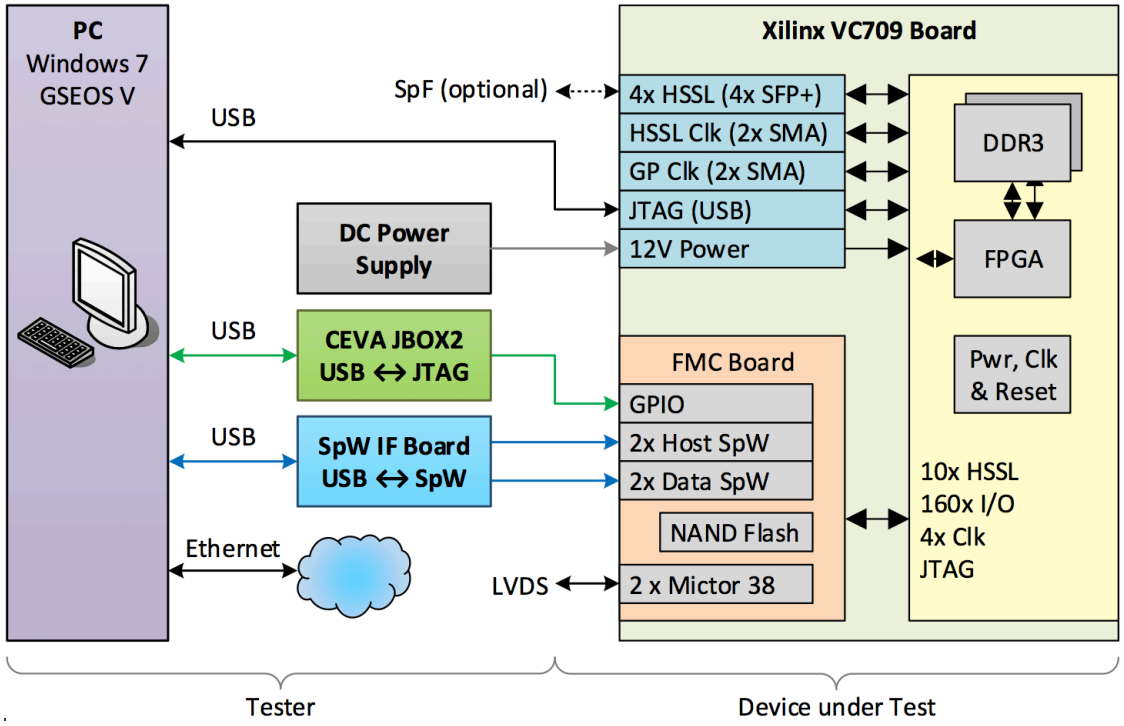
2) Integrate Benchmarks with Simulator

3) Integrate Benchmarks with FPGA

4) Measure Performance on FPGA and estimate ASIC performance

5) Apply Algorithmic Optimizations

System Test Bed

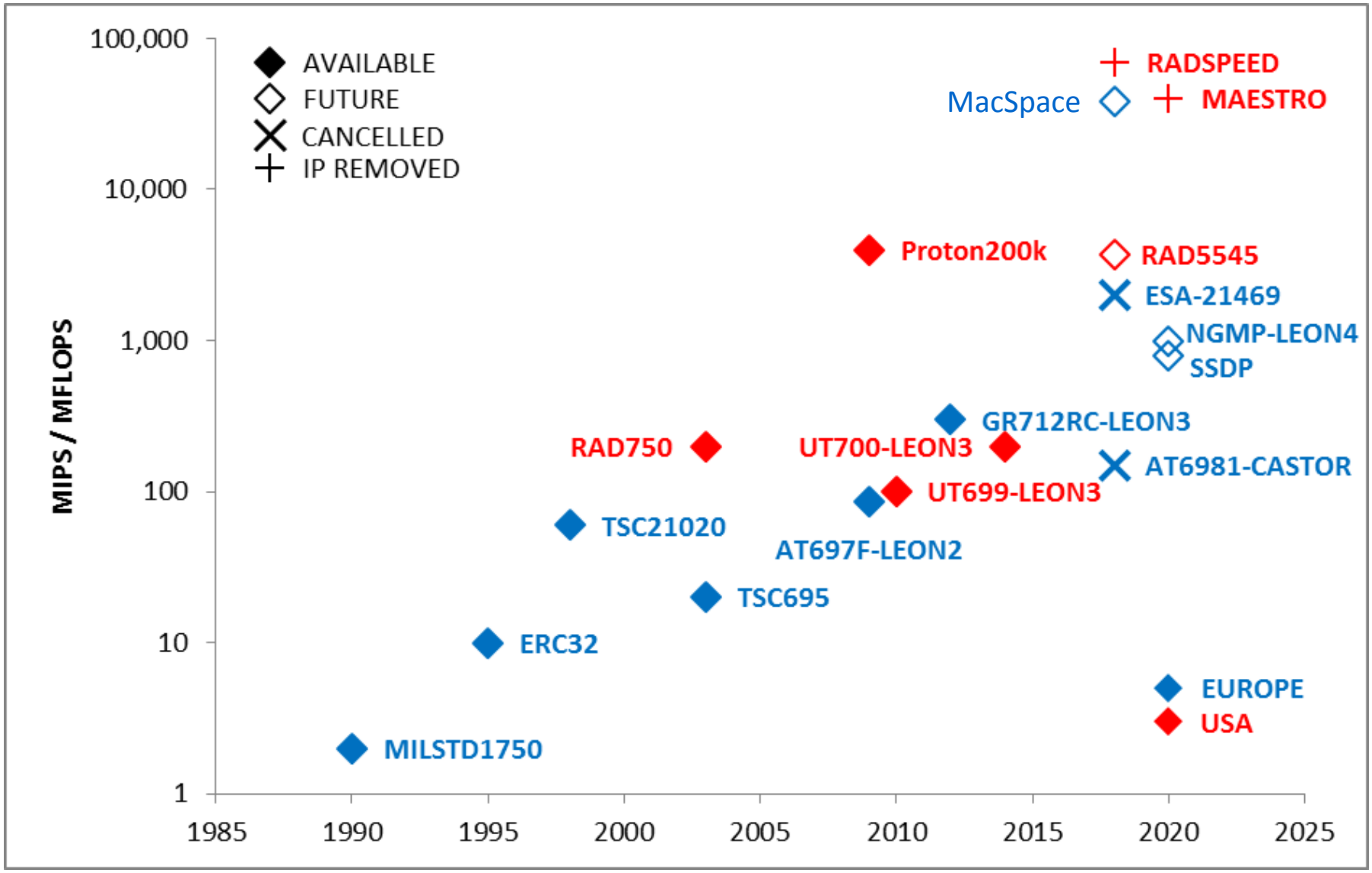


- The development of a MacSpace demonstrator is part of the project to validate the usability and functionality of the system.
- The processor architecture is implemented in a high-performance FPGA (Xilinx Virtex 7) representing the MacSpace RC64 prototype, which executes the image processing.
- A personal computer performs the management and the payload data handling.
- The GSEOS V software package is used to send preprocessed radar data, control and monitor the prototype as well as to analyse the results and qualify the performance.

Target Performance Table of the MacSpace Processor

*MacSpace Processor: 64 FXD/FP cores, 4 Mbytes**	*MacSpace Processor Functions*	*Performance*	*Performance / Power*
ITAR-Free	Yes		
Functionality as identified by Joint EC-ESA-EDA task force [EU-2012]	Floating point; Radiation hardness; High-speed interfaces; Interfaces to ADC, DAC		
Radiation-Hardness	Rad-hard implementation for total dose hardness, no SEL and high SEU immunity	300 kRad	
High speed interfaces	DDR2/3 interface	25 Gbps	
	High-speed serial interfaces	120 Gbps aggregated	
	Parallel LVDS interface to ADC/DAC	38 Gbps	
Power		10 w	
Clock Freq.		300 MHz	
GFLOPS	**(floating point)**	38 GFLOPS	3.8 GFLOPS/Watt
GMACS	**(16 bit)**	75 GMACs	7.5 GMACs/Watt
GOPS	**(fixed point)**	150 GOPS	15 GOPS/Watt
ESA benchmark B2	**(FFT 4096)**	2 Gsps^	200 Msps/Watt
ESA benchmark B5	**(demodulator and filter)**	500 Msps^	50 Msps/Watt

Target Performance Table of the MacSpace Processor



We do NOT count on 16-bit computations only

- Applications are usually developed on 32-bit and 64-bit processors (mostly Intel x86), before being ported to any kind of space processor
- It's tough to deal with precision issues of new applications
- Datasets are growing and need larger address space
- Experience shows that industrial players and government agencies do not want to change their algorithms to adapt to special hardware

Quote by US DoE: "We do not put your sh*t into our code!"

We do NOT count on 16-bit computations

Against all odds the RC64 seems to perform unexpectedly well also in 32-bit arithmetics

Microbenchmark Measurements

Test No	Name	T1	T2	Total	Iterations	Cycles/s		Cycles/OP	OP/s
B1-1	abs32	599851	681408	81557	100000	300000000		0,81557	367840896,6
B1-2	add32	682244	713799	31555	100000	300000000		0,31555	950720963,4
B1-3	and32	714654	796674	82020	100000	300000000		0,8202	365764447,7
B1-4	cnot32	797565	1148055	350490	100000	300000000		3,5049	85594453,48
B1-5	div32	1888720	2009720	121000	1000	300000000		121	2479338,843
B1-6	mul32	1623576	1786558	162982	100000	300000000		1,62982	184069406,4
B1-7	mad32	1787394	1887732	100338	100000	300000000		1,00338	298989415,8
B1-8	neg32	1148957	1205631	56674	100000	300000000		0,56674	529343261,5
B1-9	not32	1206442	1247344	40902	100000	300000000		0,40902	733460466,5
B1-10	or32	1248219	1317271	69052	100000	300000000		0,69052	434455193,2
B1-11	rem32	2011143	2045492	34349	1000	300000000		34,349	8733878,716
B1-12	shl32	1318395	1418700	100305	100000	300000000		1,00305	299087782,3
B1-13	shr32	1419863	1520324	100461	100000	300000000		1,00461	298623346,4
B1-14	sub32	1521254	1552809	31555	100000	300000000		0,31555	950720963,4
B1-15	xor32	1553683	1622735	69052	100000	300000000		0,69052	434455193,2

All benchmarks were programmed in C, and compiler optimized.

SAR Processing

Modern Synthetic Aperture Radar (SAR) systems are continuously developing:

- higher spatial resolution and new modes of operation
- use of high bandwidths
- combined with wide azimuthal integration intervals

For focusing such data, a high quality SAR processing method is necessary:

- Wavenumber domain (Omega-K) processing is commonly accepted to be an ideal solution of the SAR focusing problem. It is mostly applicable on spaceborne SAR data where a straight sensor trajectory is given.

SAR Benchmark

TU Braunschweig in close connection with the DLR is conducting experimental benchmarks on a representative SAR application excluding preprocessing steps.

The application consists of:

i) Range FFT
ii) Range compression
iii) Azimuth FFT
iv) Modified Stolt Mapping
v) Range IFFT
vi) Azimuth Compression
vii) Azimuth IFFT

SAR Benchmark

SAR Processing is more compute intensive than Hyperspectral Imaging!

It took 47 seconds on an x86 to compute (in double precision) the shown result image (Input data with size of about 500MB).

Computation Type	Number of arithmetic computations
Mixed Radix FFT	~ 50 G
Divisions	~ 1.3 G
Other	~ 4.9 G
Total	~ 56.2 G

SAR Benchmark

SAR Processing is more compute intensive than Hyperspectral Imaging. It took 47 seconds (double precision) on an x86 to compute (Input data with size of about 500MB).

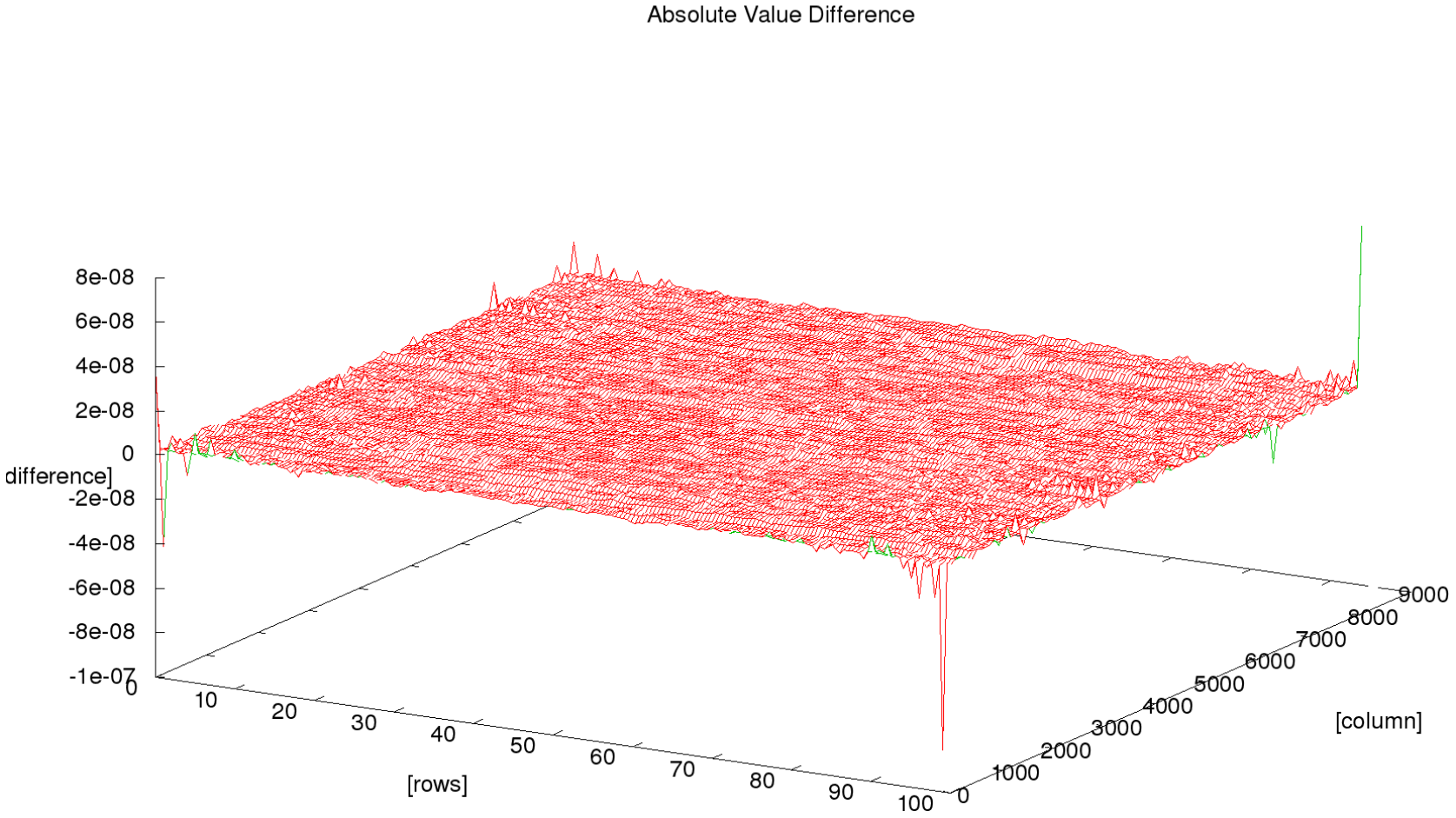
Computation Type	Number of arithmetic computations
Mixed Radix FFT	~ 50 G
Divisions	~ 1.3 G
Other	~ 4.9 G
Total	~ 56.2 G

Reduction of compute time achievable through:

- **Loss in precision (approximated computing)**
- **Replacing computations through use of Look-Up Tables (LUT)**
- **Optimized pipelining**
- **Higher degree of parallelism**

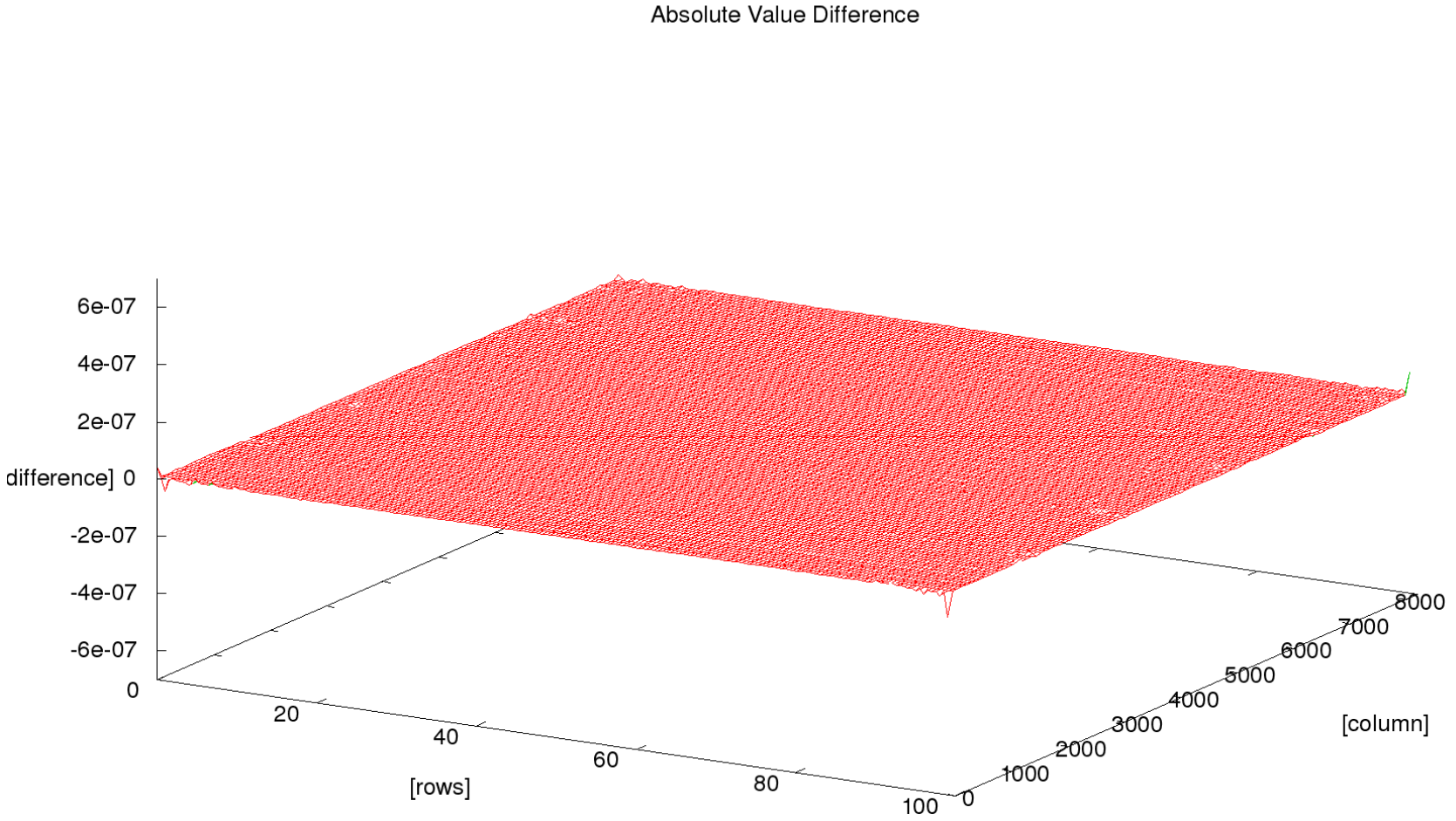
Comparison to original results

Shown is the difference of absolute values of complex number results from SAR computation before and after applying approximated computing.



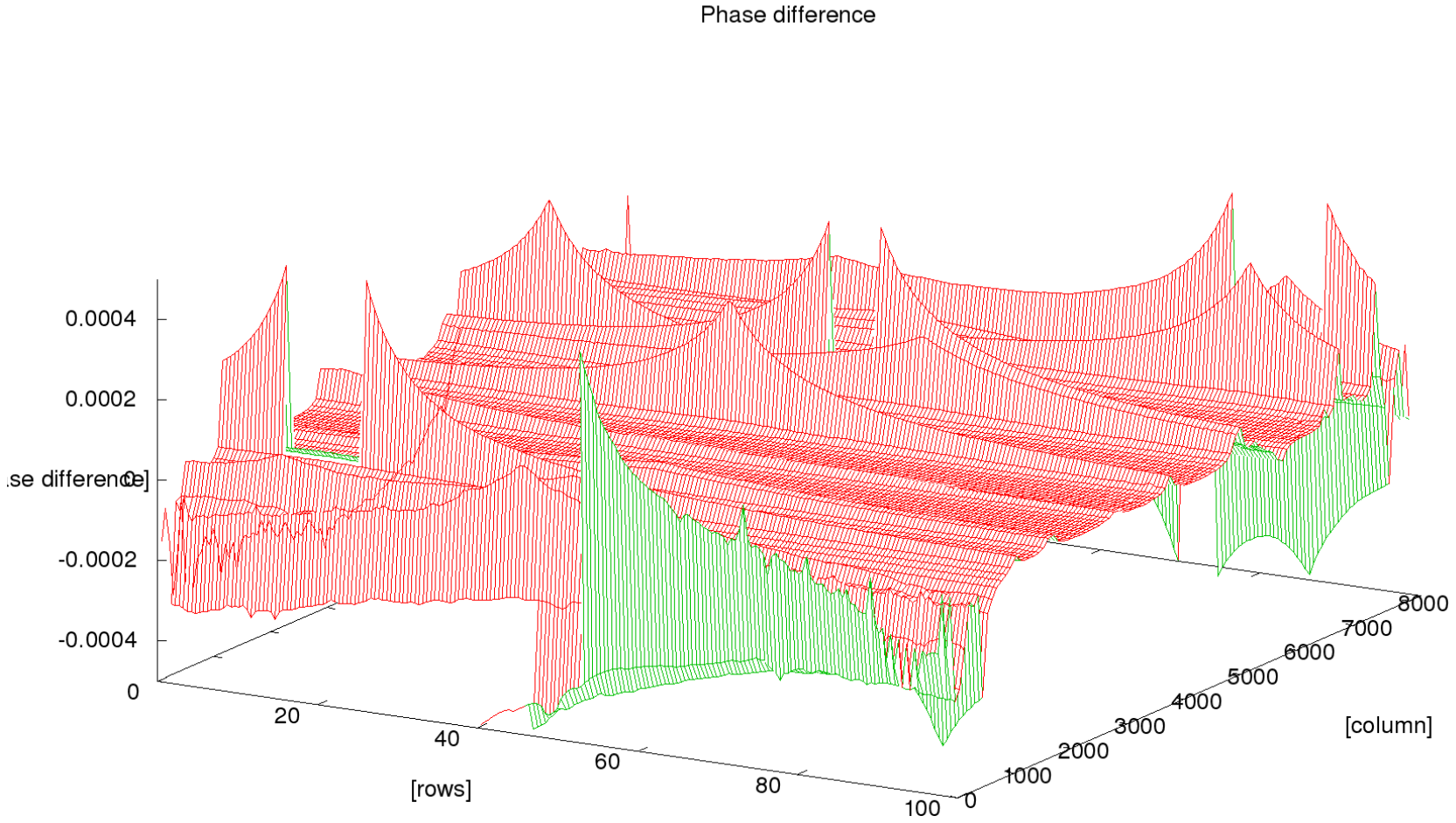
Comparison to original results

Shown in adjusted scale is the difference of absolute values of complex number results from SAR computation before and after applying approximated computing.



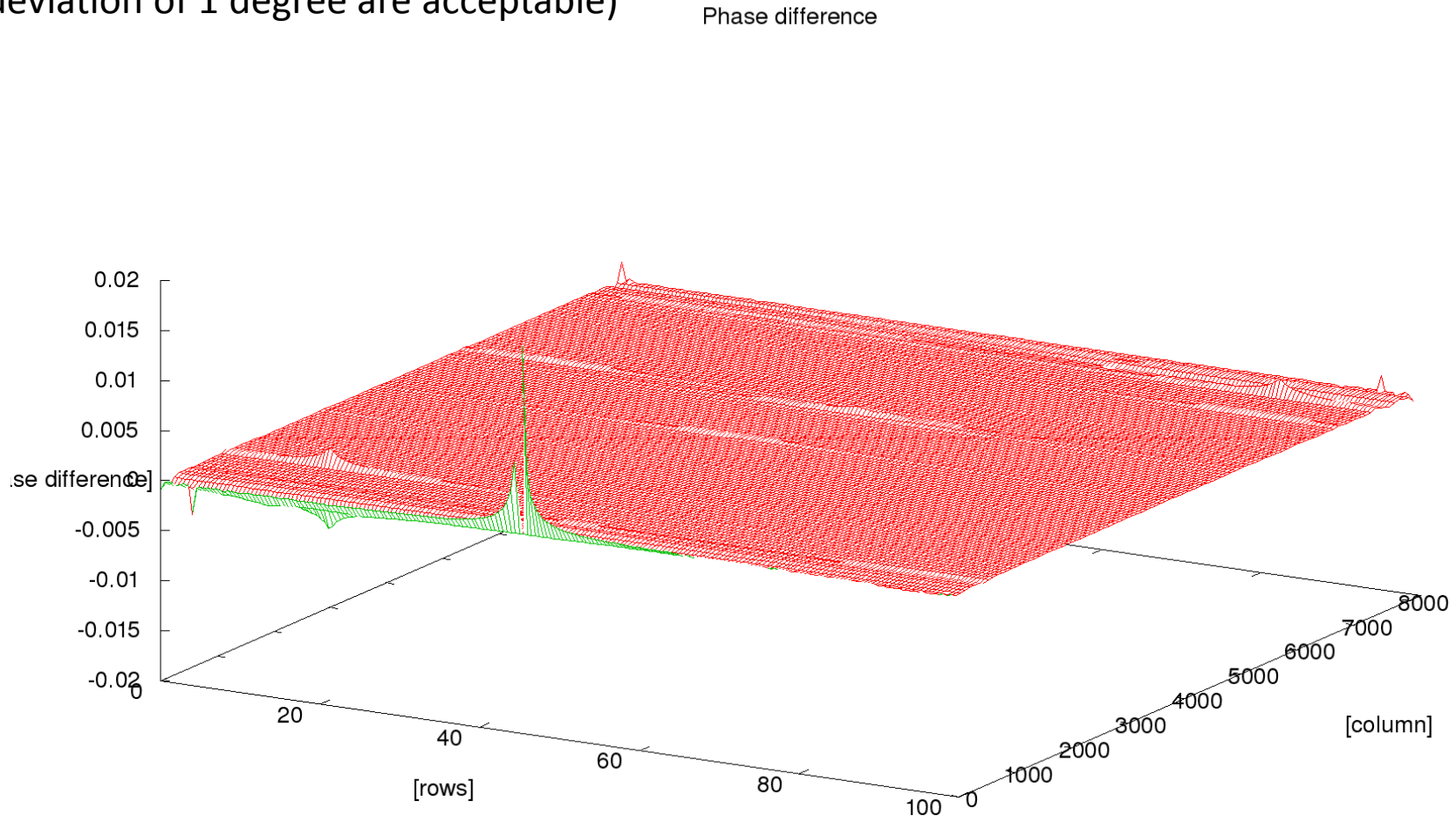
Comparison to original results

Shown is the phase angel between of complex number results from SAR computation before and after applying approximated computing



Comparison to original results

Shown in adjusted scale is the phase angel between of complex number results from SAR computation before and after applying approximated computing
(deviation of 1 degree are acceptable)



Estimated Performance of SAR benchmark

- Computation-wise one single **RC64 chip could be capable of processing SAR data of 8192x8192 complex values** (single precision floating point, i.e. in total 512MB) **in under 2 seconds @ 300MHz**

- 100% compute utilization
- based on a computation count: 62G Floating Point Operations @ 38 GFLOPS)
- Since the onboard data bandwidth (per core: L1 data – peak 128bit read/write per cycle per core from/to registers, L1 from/to shared memory ('L2') 128bit @~50% utilization read and 32bit write) potentially can sustain the demand by computations, reaching the best-case performance will be a matter of latency hiding.
- In the worst-case scenario, we still expect the application to finish processing the above described data in less than 1 minute.

Thanks for your attention!

ANY QUESTIONS?