

Space Debris Detection on the HPDP, a Coarse-Grained Reconfigurable Array Architecture for Space

Diego Suárez - Airbus DS GmbH, Germany
Tim Helfers, Airbus DS GmbH, Germany
Dr. Josef Weidendorfer, TU München, Germany.
Daniel Bretz, Airbus DS GmbH, Germany
Dr. Jens Utmann, Airbus DS GmbH, Germany

DSP Day 2016, Gothenburg, Sweden June 2016.

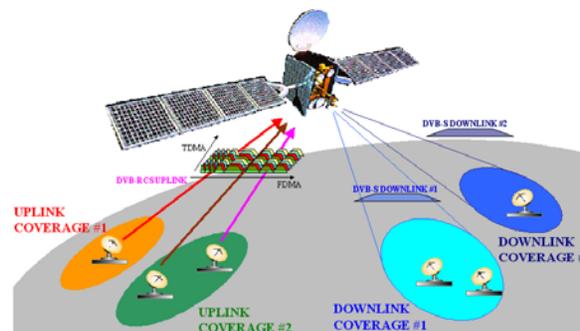
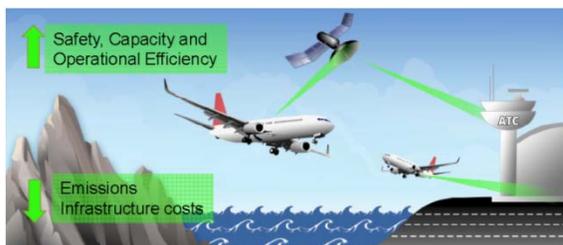
Agenda

1. **Objectives:** algorithm effectiveness, portability and performance in HPDP.
2. **HPDP Architecture:** features and parallelism capabilities.
3. **Feature Detection Algorithm:** boundary tensor and thresholding.
4. **Implementation in the HPDP:** maximize throughput.
5. **Results:** Error, throughput and detection effectiveness. Estimations on planned hardware.
6. **Conclusions.**
7. **HPDP Roadmap and Ongoing Work**

Motivation for HPDP

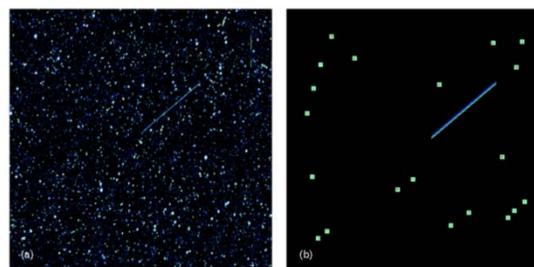
Payload Processing Applications in Telecom and Non-Telecom Missions in Airbus DS

Receiver for Automatic Dependent Surveillance - Broadcast (ADS-B) for Air Traffic Control



Regenerative On-Board Processor for User-to-User communication

Satellite-based Machine-to-Machine (M2M) communication



Feature Extraction for Space Debris Monitoring

→ New Generation Reprogrammable Processor Technologies are required

1. Objective

Space debris detection with the HPDP

SSA System Architecture

NEO Sensors

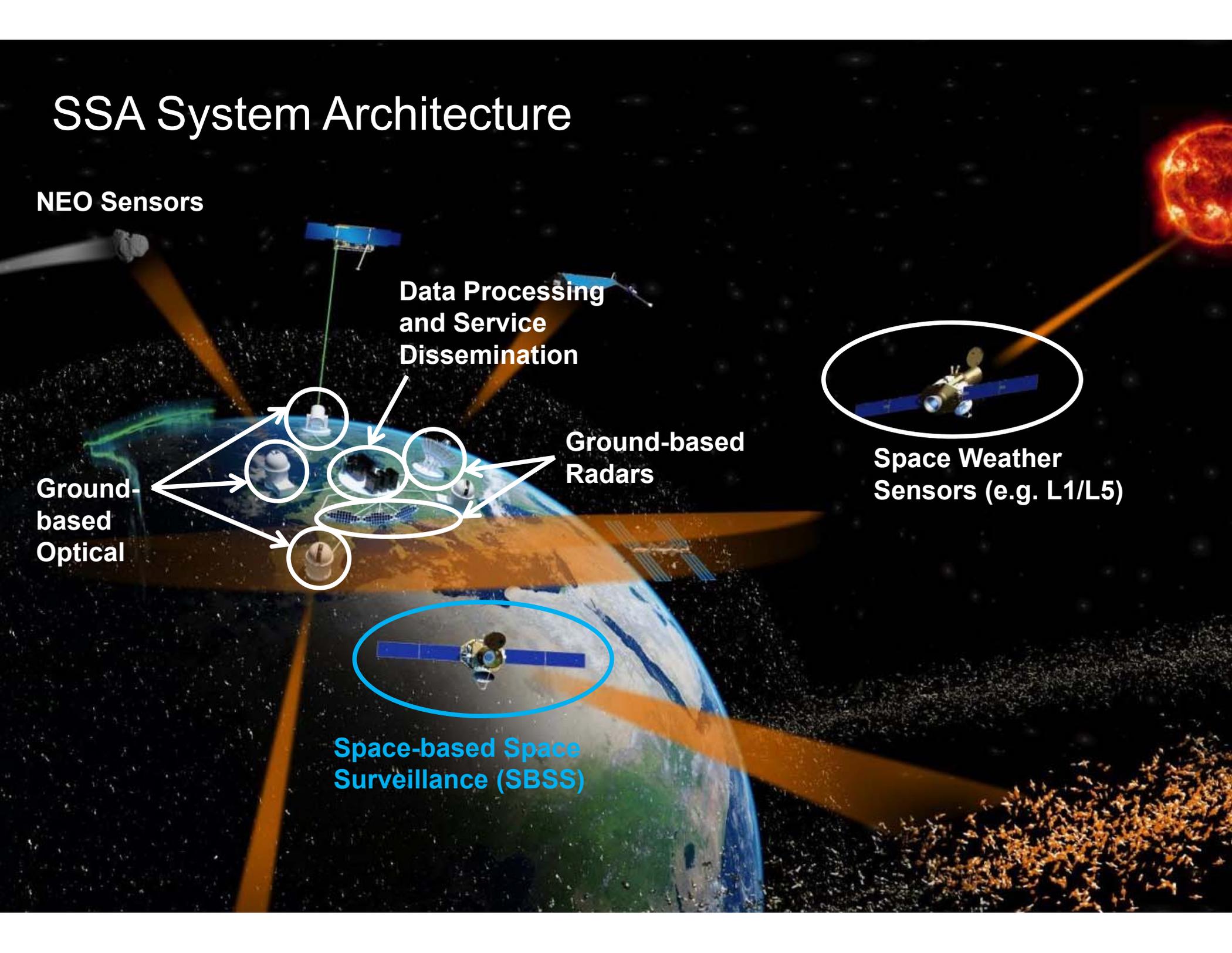
Data Processing
and Service
Dissemination

Ground-
based
Optical

Ground-based
Radars

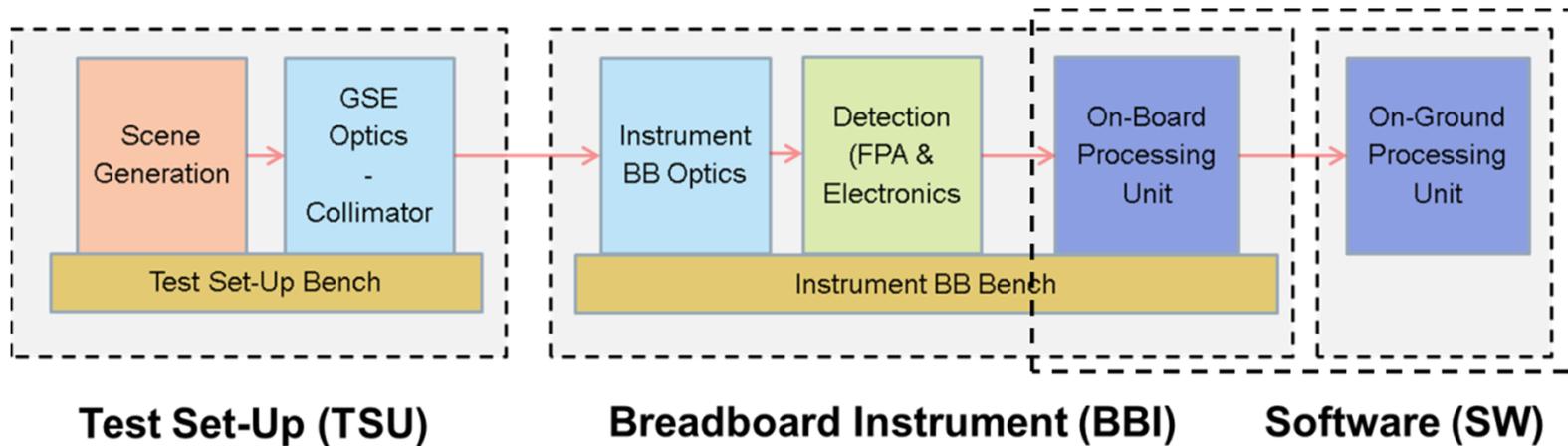
Space Weather
Sensors (e.g. L1/L5)

Space-based Space
Surveillance (SBSS)



Demonstrator Project

- **End-to-end processing pipeline** currently covered in **ESA GSTP “Optical In-Situ Monitor” study**
 - Airbus DS, AIUB, Micos
 - Develop, integrate & test of laboratory breadboard system
 - E2E Test of Data Processing Pipeline



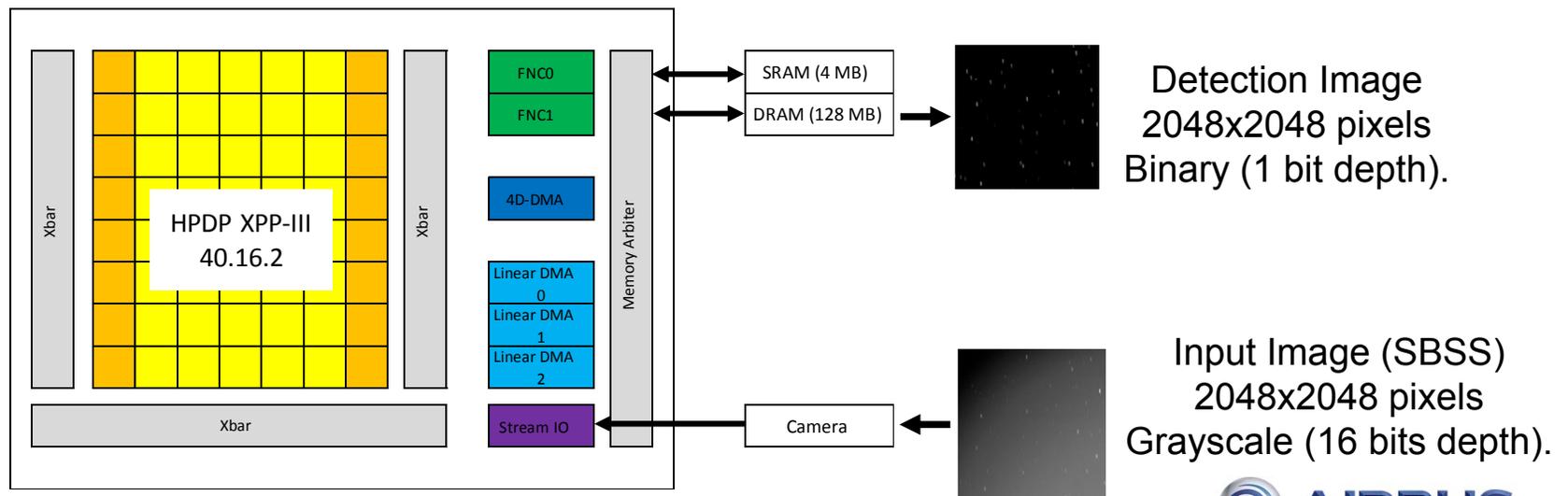
- One possible algorithm to detect objects in an image : **Boundary Tensor**
- Image shall be processed on-board in order to just downlink the relevant data rather than full images

Design Requirements

✓ **Objective:** determine...

- ❑ *Performance:* process one image in one second or less.
- ❑ *Portability:* suitability to implement data-flow.

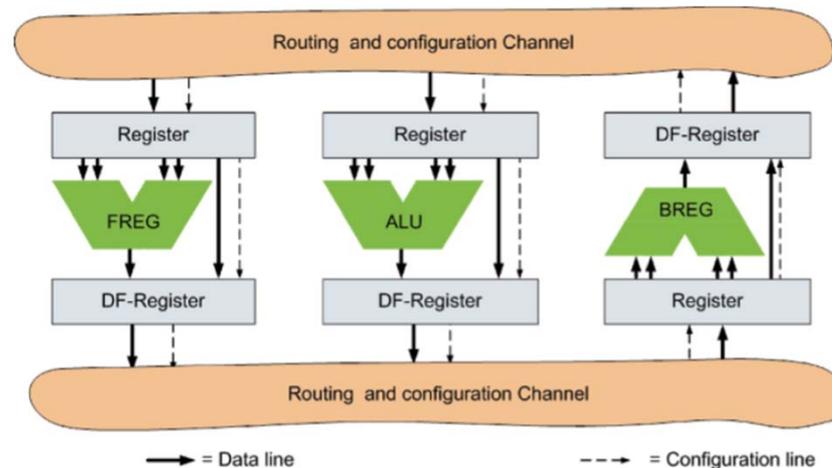
...of a space debris detection algorithm in the HPDP architecture.



2. HPDP architecture

HPDP Architecture Features

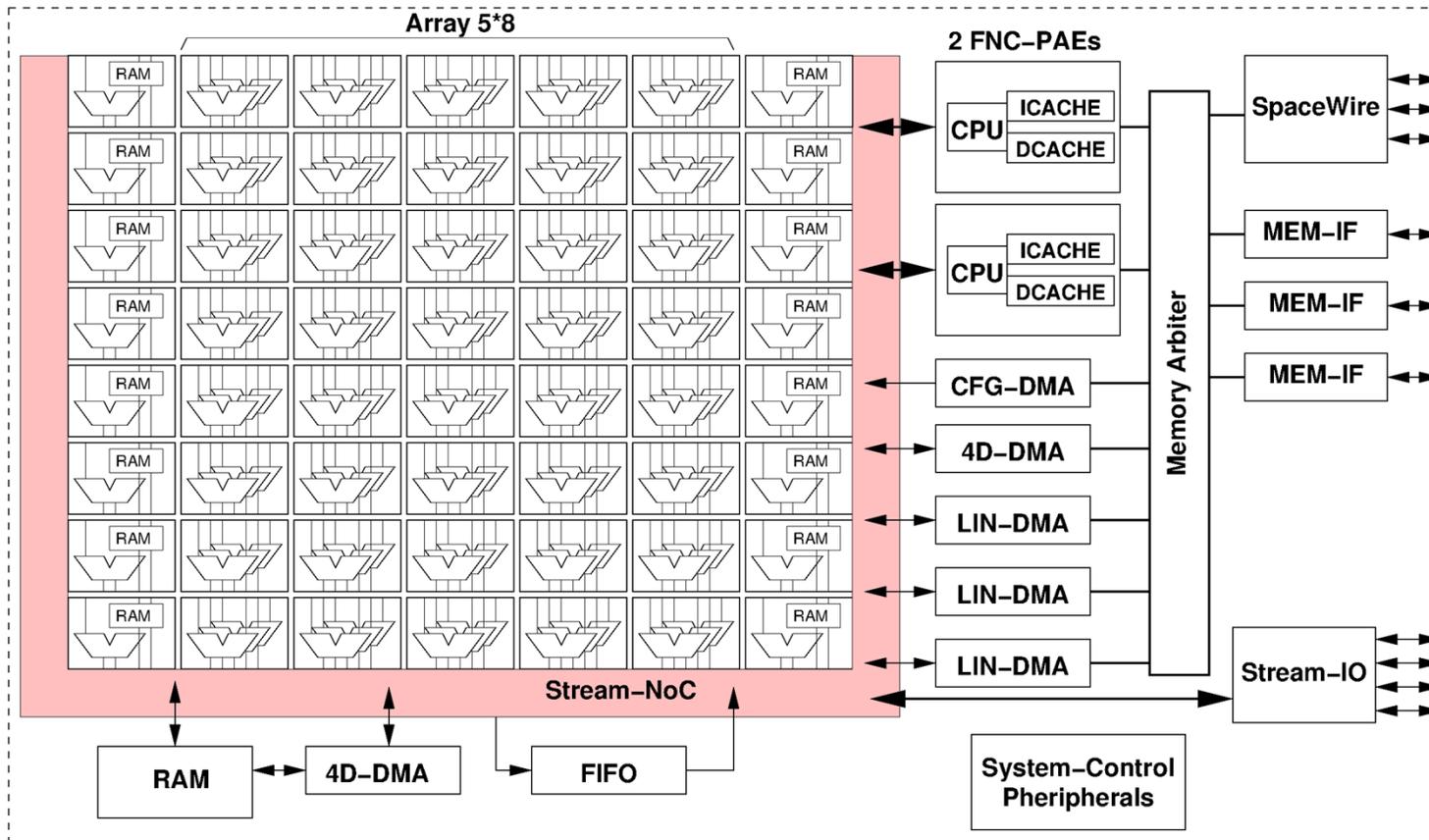
- The eXtreme Processing Platform (XPP) is the core of the HPDP architecture:
 - Coarse-grain reconfigurable of Processing Array Elements (PAE).
 - Automatically synchronizing, packet-oriented communication network.
 - Runtime and self-reconfigurable.



The XPP ALU Processing Array Element [1]

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

XPP Architecture



The XPP-III 40.24.12 Reference Design [11]

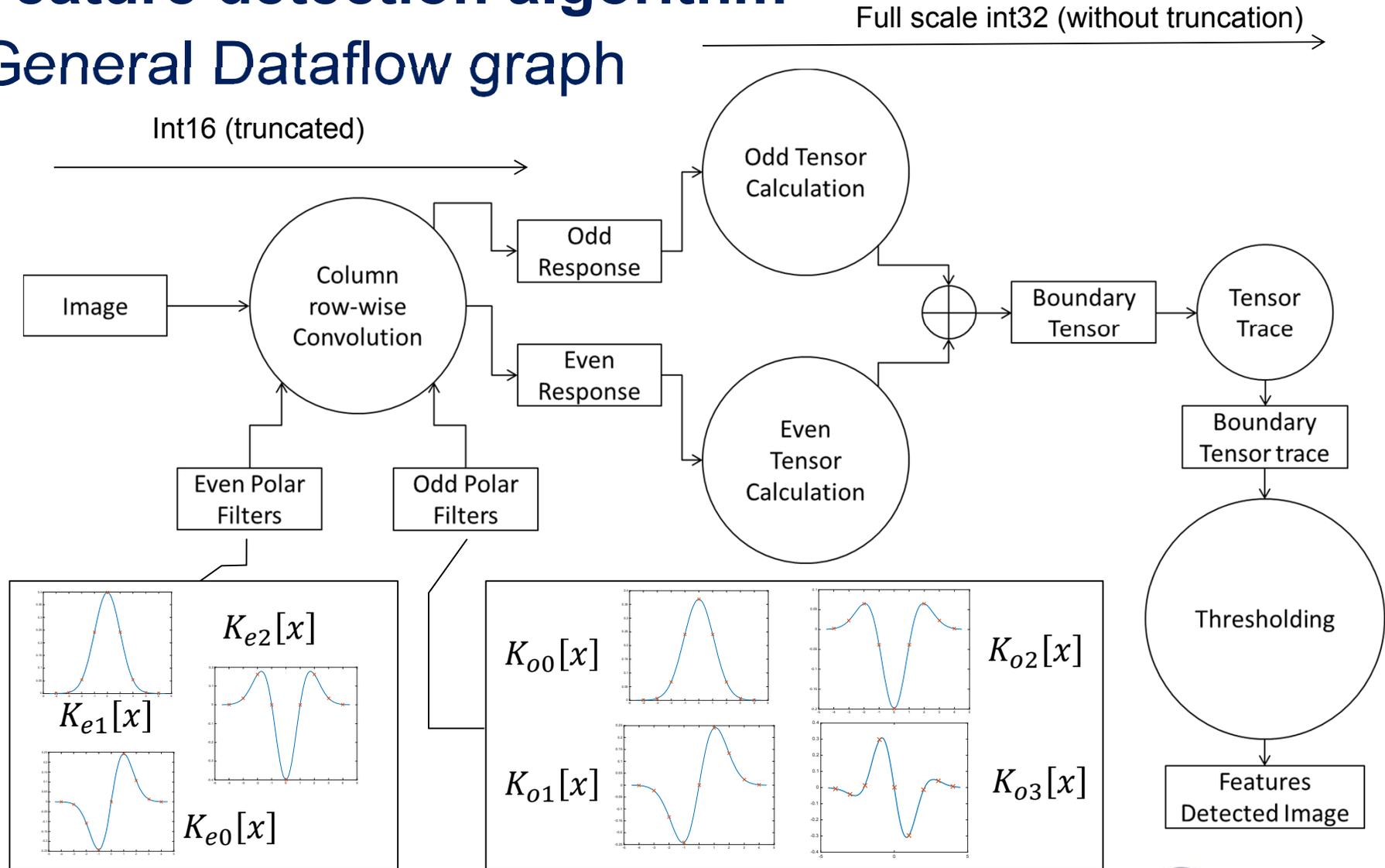
This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

3. Feature Detection Algorithm: Boundary Tensor and Thresholding

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential.
It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Feature detection algorithm

General Dataflow graph

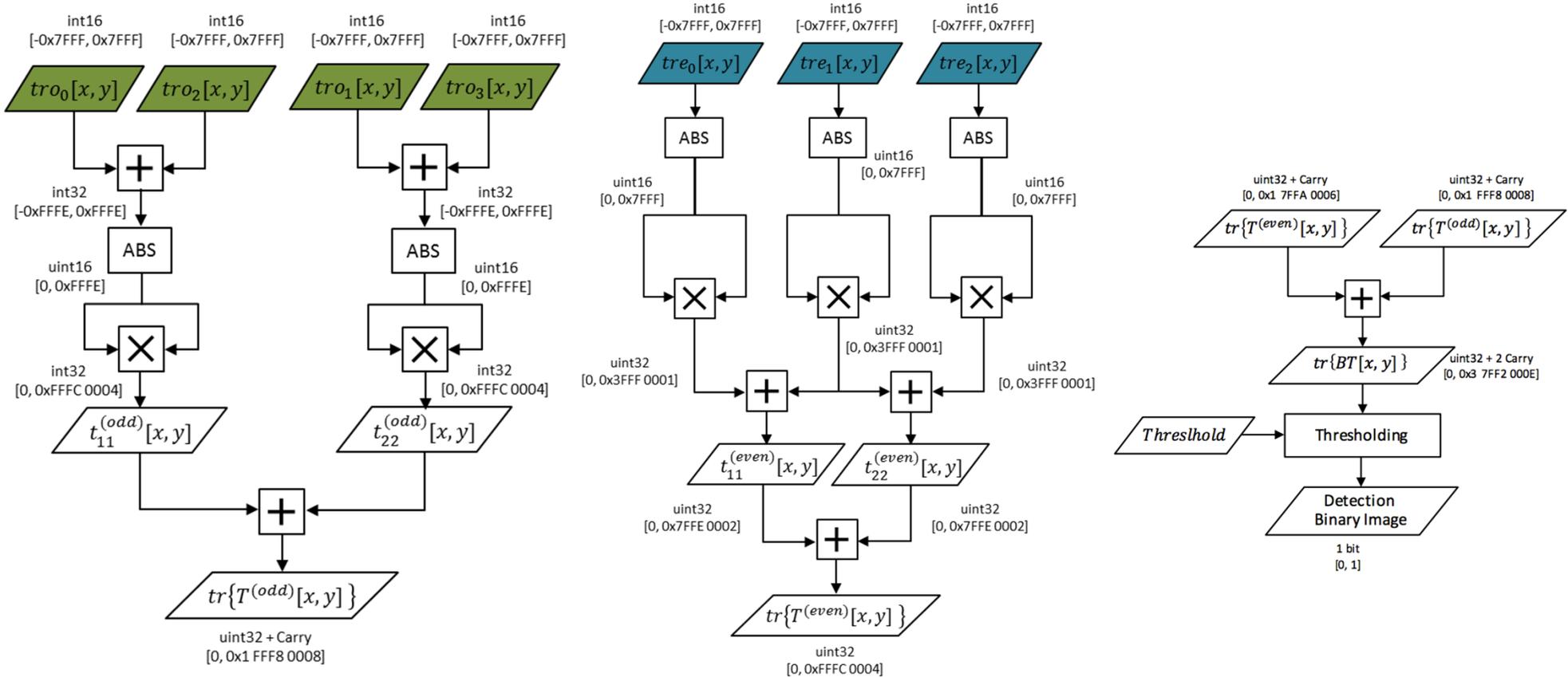


U. Köthe, VIGRA Library [5]

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Implementation in HPDP

Boundary Tensor and threshold calculation

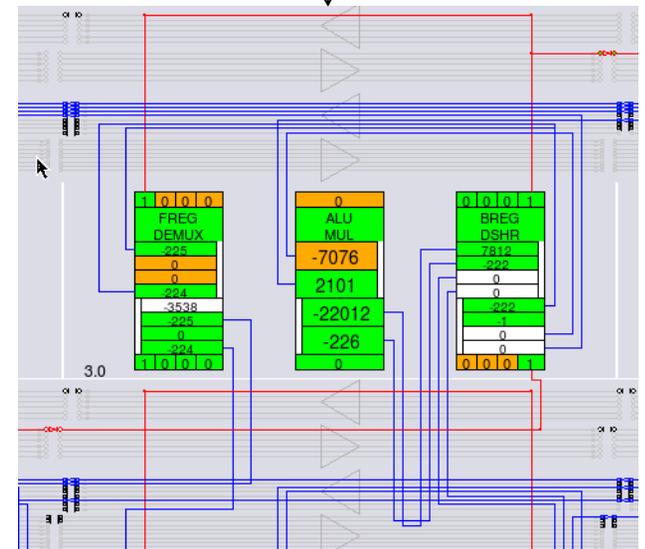


This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

4. Results

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential .
It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

XPP Array Simulator View (XDBG)



XPP Array Simulation of the row convolution with odd symmetry kernel Configuration.

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Feature detection algorithm

Throughput in HPDP Simulation

Simulated throughput of each of the XPP array configurations used in the feature detection algorithm:

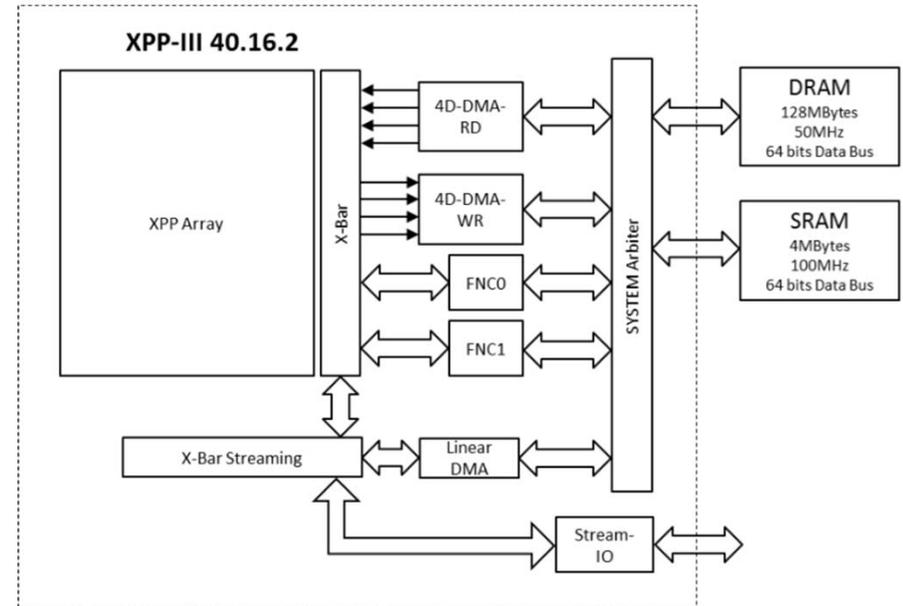
		Throughput		Bit Rate with XPP Array Clock of 200MHz
		[Pixels/cycle]	[Bytes/cycle]	[MBytes/s]
Row Convolution Even Symmetry		1.63	3.27	653.6
Row Convolution Odd Symmetry		1.99	3.98	796.6
Column Convolution Even Symmetry		1.63	3.27	653.7
Column Convolution Odd Symmetry		1.99	3.98	796.6
Transpose 2D Array		1.60	3.20	639.4
Boundary Tensor and Thresholding	In Stream	1.90	3.81	761.6
	Out Stream	0.27	0.54	108.8

⇒ Memory Bandwidth requirement (XPP array @ 200 MHz) for single port transactions: 1594 MBytes/s.

Feature detection algorithm

Expected HPDP Hardware

- XPP array @ 200 MHz.
- FNC-PAE @ 100 MHz.
- Two Memory ports 64 bits wide
 - 100 MHz single port SRAM (4 MBytes).
 - 50 MHz single port DRAM (128 MBytes).



⇒The theoretical maximum memory bandwidth is 800 MBytes/s for the SRAM and 400 MBytes/s for the DRAM.

⇒Algorithm in XPP array achieves nearly twice the data rate of the memory.

⇒Memory is the performance bottleneck in the expected HPDP hardware.

Feature detection algorithm

Sub-Image processing

- Algorithm makes heavy use of memory.
- 64 read/write access of a 2048x2048 int16 image buffers (8 MBytes):

Algorithm Memory Transaction Requirements		
Module	Number of Read RAM Operations [One Image]	Number of Write RAM Operations [One Image]
Row Convolution	7	7
Column Convolution	7	7
Transpose	14	14
Boundary Tensor	7	1
Total	64	

- ⇒ SRAM should be used for processing.
- ⇒ Sub-image processing is done.

Feature detection algorithm

Performance

- Input Image is split in 16 sub-images that are processed each at a time using image buffers in SRAM.
- DRAM is used to store input and result image.

Total memory access Time for the complete algorithm			
Task	Time [sec]	Origin	Destination
transfImg_StreamIOtoDRAM	0,02097	Stream IO	DRAM
transfSubImg_DRAMtoSRAM	0,02097	DRAM	SRAM
boundaryTensor	0,67109	SRAM	SRAM
transfRes_SRAMtoDRAM	0,02097	SRAM	DRAM
Total [sec]	0,734		

⇒ Real-time requirement is met.

Feature detection algorithm

Reconfiguration Time

Array Reconfiguration Time	
Sub-Image Number	16
Array Clock	200 MHz
Required Cycles for single Reconfiguration	30.000
Number of Configurations (per image processing)	6
Total required Reconfigurations (for 16 Sub-Images processing)	96
Total Cycles in Reconfiguration	2.880.000
Reconfiguration Time [sec]	0,01440

⇒ 1.4ms out of 73.4ms for Reconfiguration

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Feature detection algorithm

Detection Results in HPDP Simulation



Input Image

Geosynchronous equatorial orbit survey. Ground base capture.

Source: Astronomische Institut der Universität Bern (AIUB)

Survey01210820130809.fts

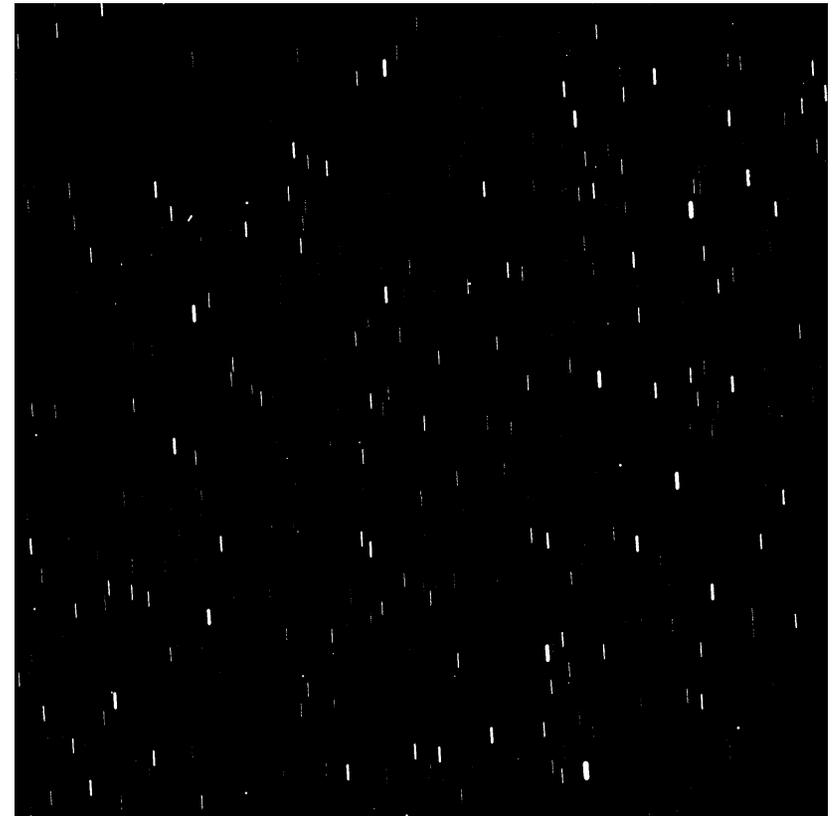
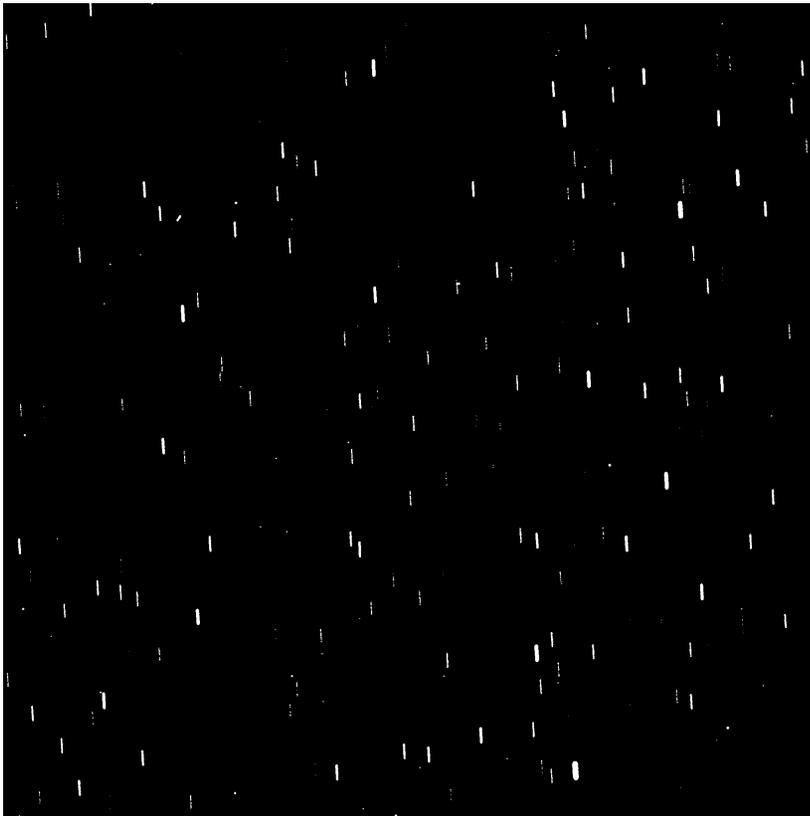
This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Feature detection algorithm

Detection Results in HPDP Simulation

Floating-point using 16 bits input image.
Threshold = 6400

HPDP Simulation Results
Threshold = 1600



Results: Thresholded boundary tensor trace

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Feature detection algorithm

Detection Results in HPDP Simulation

Zoom section in (350,441):

- Around 10% of high values (detection) are lost in the HPDP results.

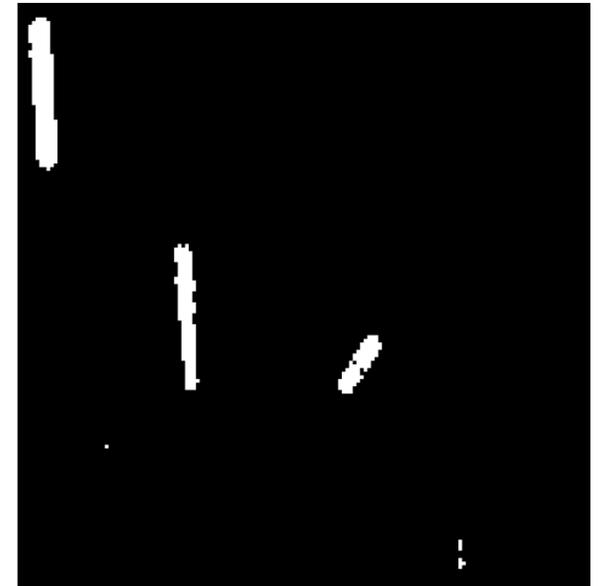
Zoom in region Result
Floating-point Reference Implementation



Zoom in region Input Image



Zoom in region Result
HPDP Simulation



Conclusions

1. Algorithm effectiveness:

- The HPDP implementation result loses 10% of detected pixels in the periphery. Centroid is conserved, then streak location capabilities are conserved.

2. Algorithm Portability:

- Convolution data-flow and boundary tensor trace calculation are suitable for implementation in XPP array.
- Algorithm takes advantage of pipelining and task parallelism of the XPP array.

Conclusions

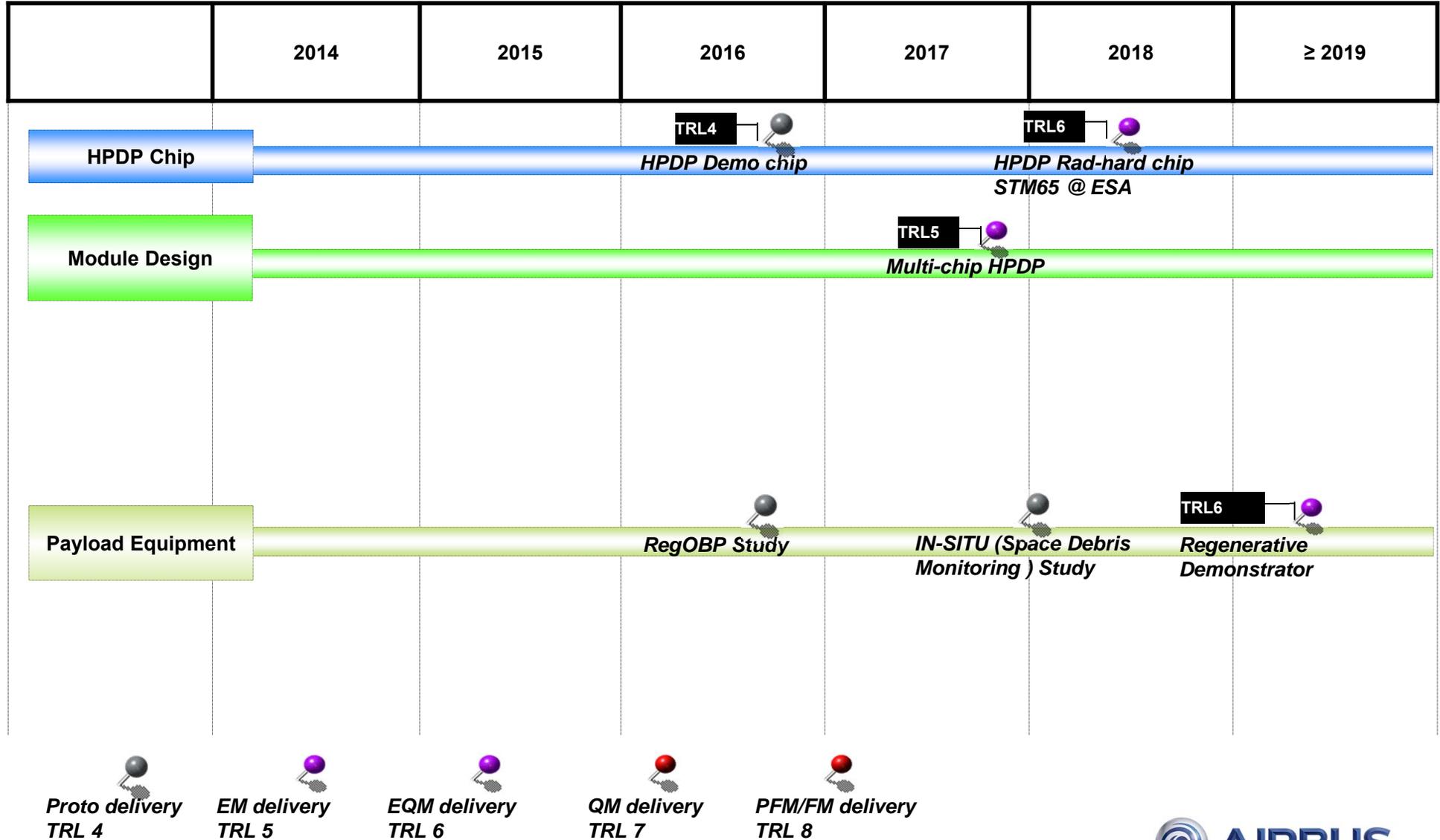
3. Algorithm Performance in the HPDP:

- Algorithm performance is limited by the memory bandwidth.
- One 2048x2048 pixels 16-bit image in 0.737 seconds. With 200MHz @XPP and 100MHz @FNC according to backend results from mapping to STM C65SPACE technology
- Average of 4.7 GOps/s for 16-bits fixed-point Multiplication-Addition operations in the XPP Array (measured with simulator for this algorithm)
- Multi-chip implementation boost performance: a HPDP chip is assigned to a set of sub-images.

4. HPDP Roadmap and Ongoing Work

This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential .
It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

Development Roadmap



This document and its content is the property of Astrium [SAS/GmbH], and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [SAS/GmbH].

References

- [1] Christophe Bobda. Introduction to Reconfigurable Computing. Springer Netherlands. 2007
- [2] PACT XPP Technologies AG. XPP-III Programming Tutorial. 2009.
- [3] PACT XPP Technologies AG. XPP-III Reference Manual - XPP Dataflow Array. 2009.
- [4] PACT XPP Technologies AG. XPP-III Processor Overview White Paper. 2006.
- [5] U. Köthe, Integrated Edge and Junction Detection with the Boundary Tensor, ICCV 03, Proc. of 9th Intl. Conf. on Computer Vision, Nice 2003, vol. 1.
- [6] Goubault, E. Static Analysis of Finite Precision Computations. Verification, Model Checking, and Abstract Interpretation. Lecture Notes in Computer Science. Volume 6538, 2011, pp 232-247.
- [7] Richard L. Wheeden. Measure and Integral: An Introduction to Real Analysis.
- [8] Jähne, Bernd, Digital image processing, 6th ed, 2005, Springer.
- [9] Ahmadreza Baghaie, Zeyun Yu, Structure tensor based image interpolation method, AEU - International Journal of Electronics and Communications, Volume 69, Issue 2, February 2015, Pages 515-522, ISSN 1434-8411.
- [10] Test Images from the Astronomische Institut der Universität Bern (AIUB).
- [11] Syed, M., Acher, G., Helfers, T., A High Performance Reliable Dataflow Based Processor for Space Applications. In: Proceedings of the ACM International Conference on Computing Frontiers, 1–4, ACM, New York, USA (2013).