ESL for Space-Based Image Processing

Lockheed Martin

Space Systems Company, Denver

MAPLD 2008

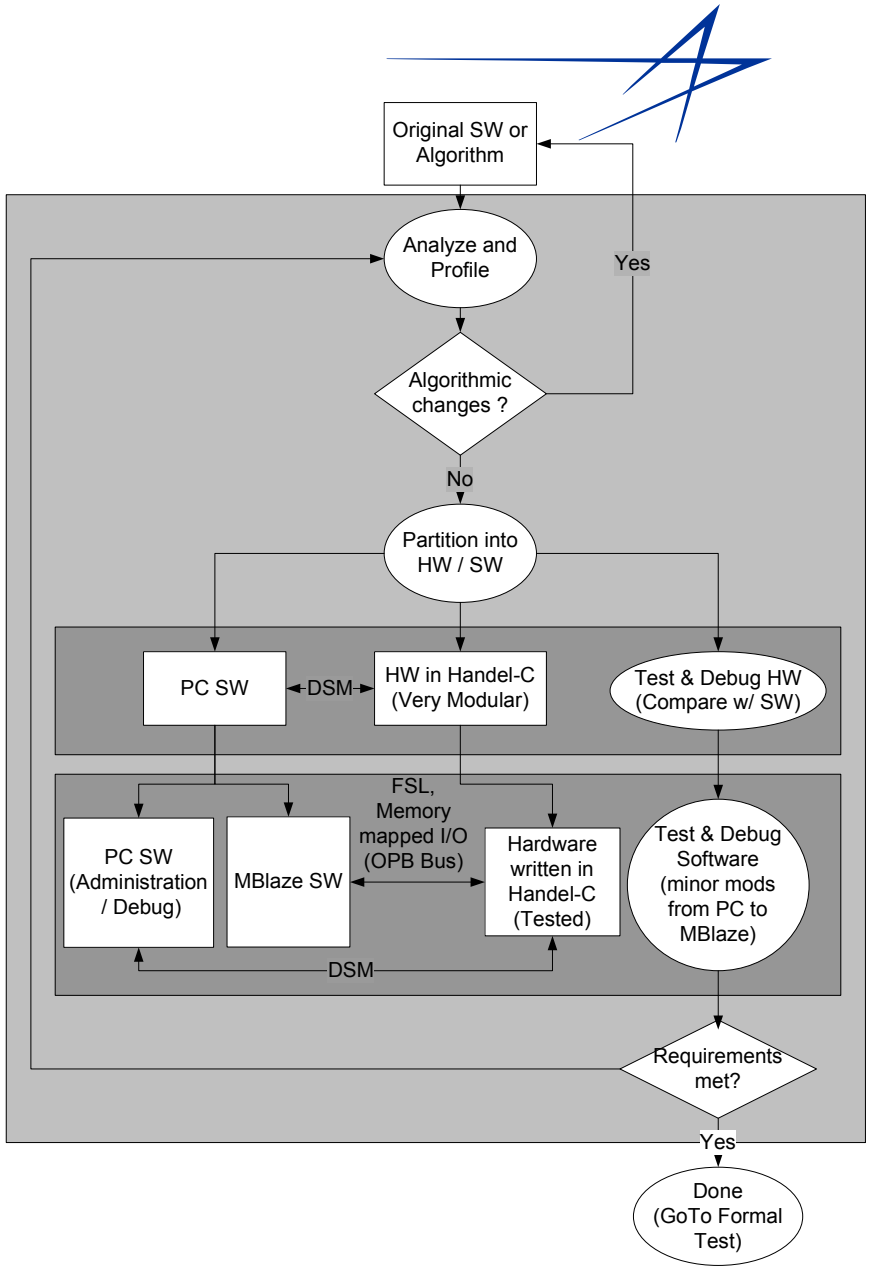
Saul H Weiss

Introduction

- Projects and Experiments
- Results
- Lessons Learned
- Looking Forward

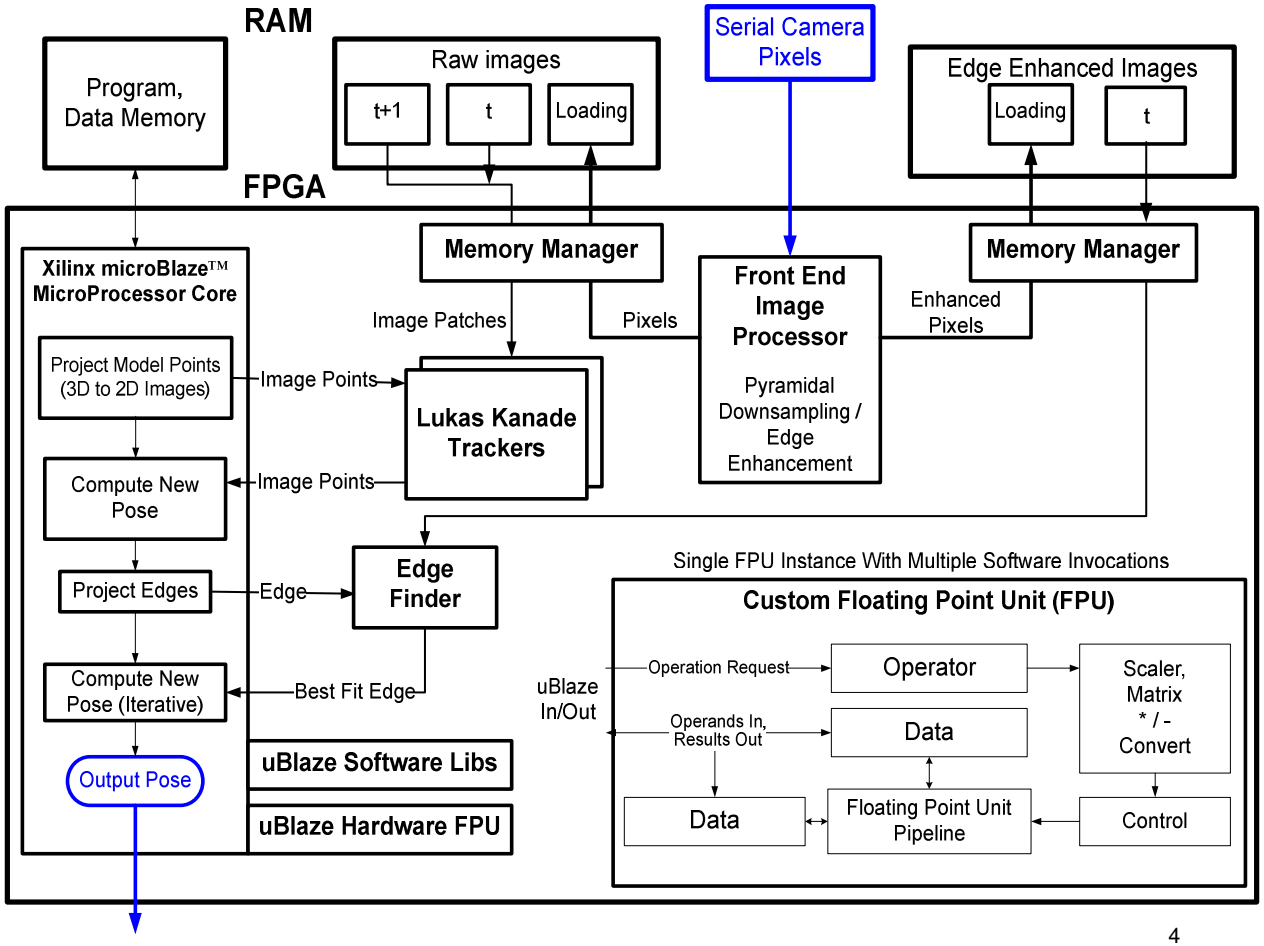
Hubble

- 9 Month Technical Capability Demonstration
  - Image processing on space HW
  - Ongoing research algorithms
  - 2 EE, 1 CS new to Handel-C™
- Natural Feature Image Recognition
  - OpenCV C++ libraries
  - Research status, updated by FPGA team
  - FPGA vs MicroBlaze™ Partitioning
  - Custom Floating Point Unit (Vectors/Matrices)



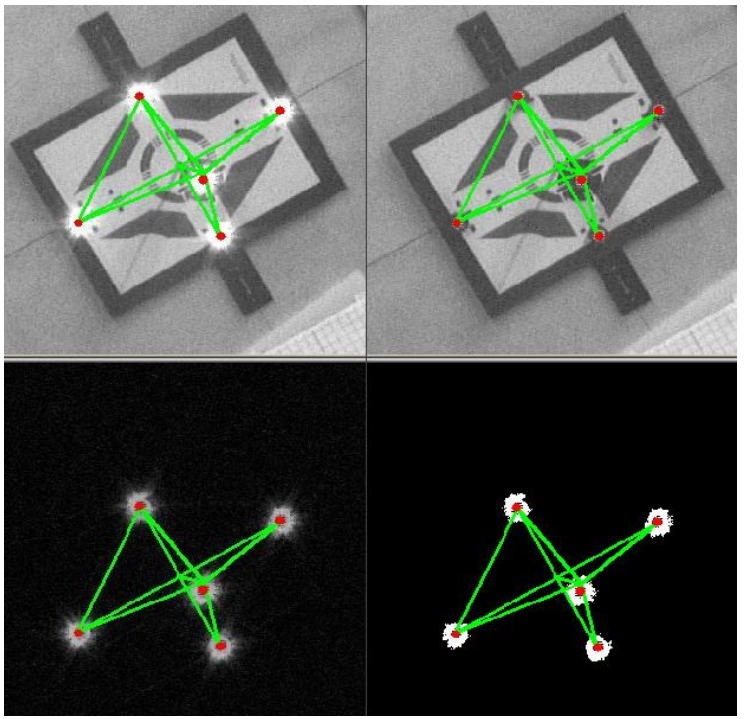
Hubble Results and Lessons Learned

- **Capability Verified on Alpha-Data Xilinx Virtex-II FPGA Board**
- **C-style language faster than straight to HDL**
- **Hardware acceleration key for image processing**
  - **Dual P4 Xeon 3.2Ghz w/ hyperthreading. ~10fps**
  - **FPGA 40Mhz ~5fps**



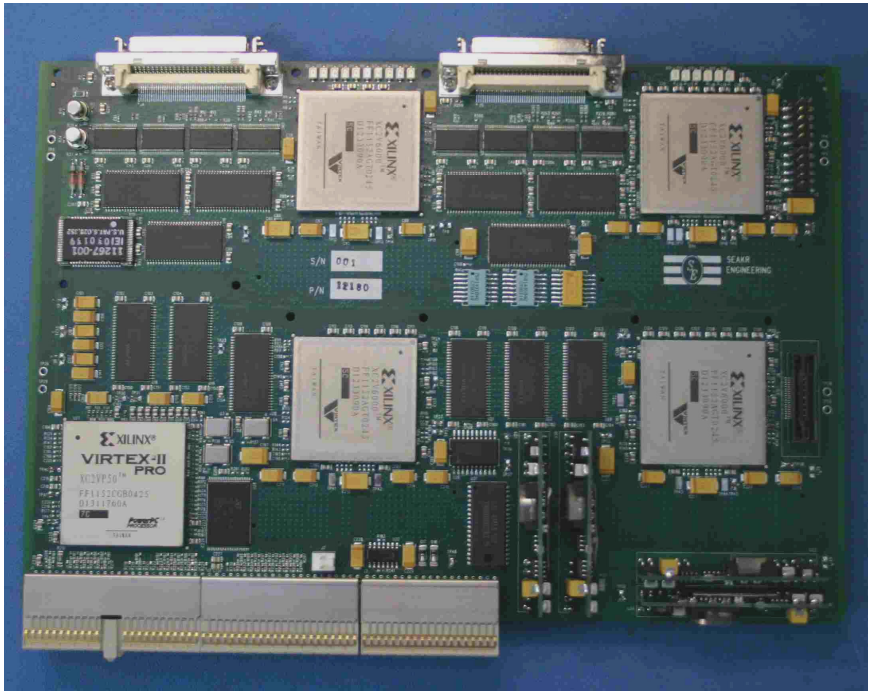
ARPO

- Autonomous Rendezvous and Proximity Operations
  - 5 Month Tech Demo
  - 2 FPGA Designers
- Refined partitioning and implementation approach
- Camera, RS422 interfaces
- Created generic Handel-C, Microblaze base design



ARPO Results and Lessons Learned

- Re-use of Hubble Image Processing Libraries
- Tool flow verified through repeated use
  - Time from concept to demo nearly halved
- Verified in Hubble EDU (4 Xilinx V2 FPGAs)
  - 30 fps
  - closed loop life-sized ISS docking mockup



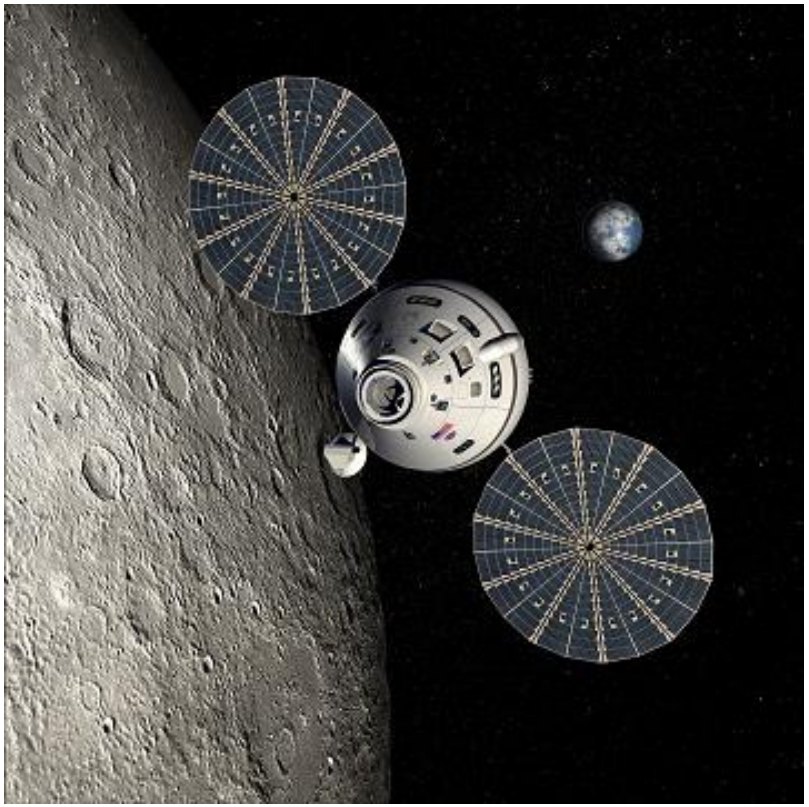
Used with permission of SEAKR Engineering, Inc.

DCU NCC-TTP IRAD

- Non-Conventional Computing Transformational Technology Program led by Tim Gallagher
  - 8 Month Research Program
  - 2 Full-Time FPGA Designers
- Algorithms designed in Matlab/Simulink
- Implementation verified using
  - Matlab/Simulink™ (Mathworks®) for baseline comparison
  - Link for Modelsim (Mentor Graphics®) for co-simulation

Orion Vision Processing Unit

- Relative Navigation Image Processing and Compression of
  - Star Trackers
  - VNS (LIDAR)
  - Docking Camera
  - Situational Awareness Cameras
- Multiple FPGAs configured in-flight based on Mission phase requirements

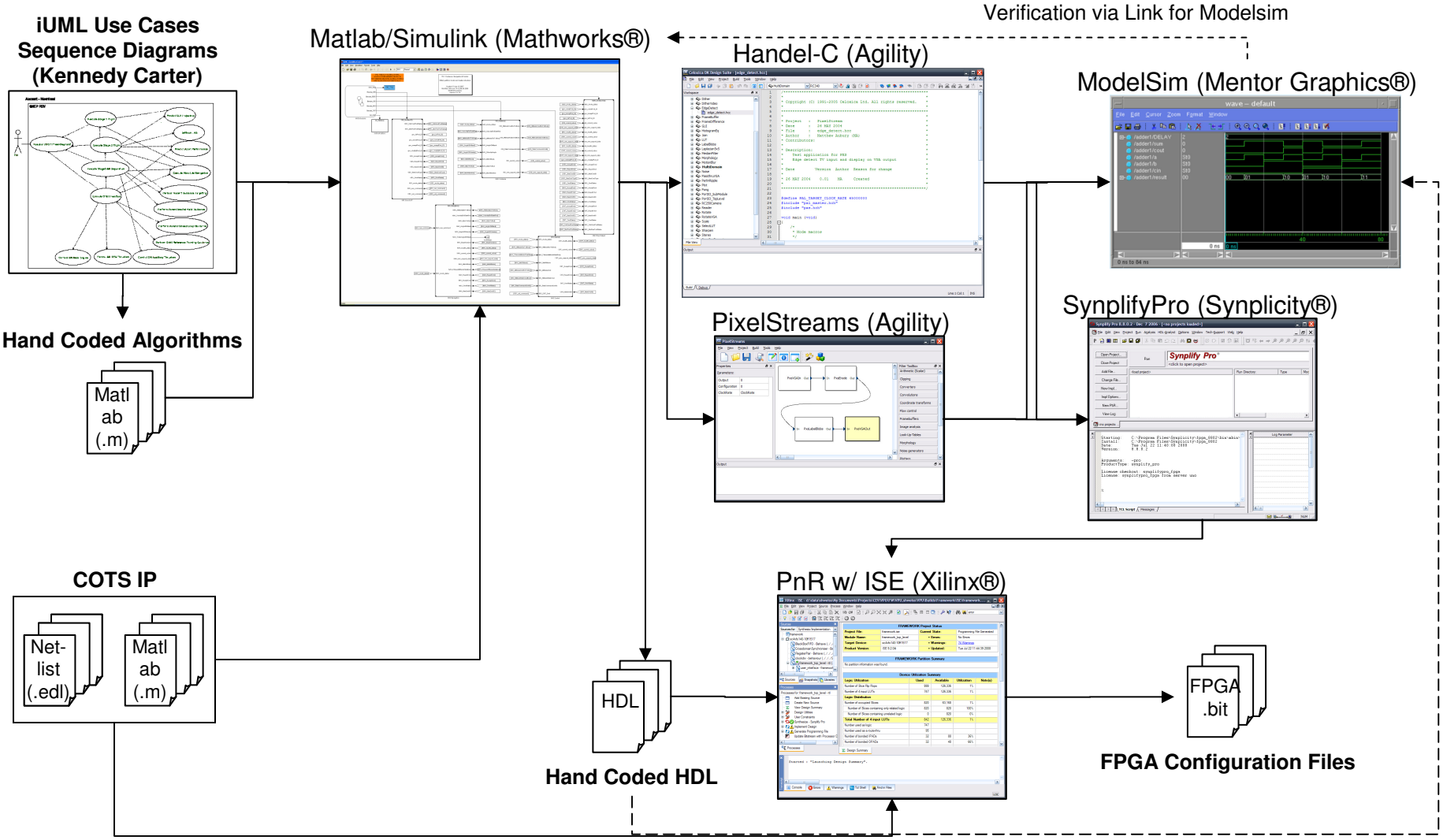


Orion VPU Analysis and Design

- "Firmware" (ASIC/FPGA) Development
  - Under Software Management
  - Meet the intent/rigor of Software Development Plan
- UML for Requirements and Interface Analysis
- The Mathworks Matlab/Simulink models for Development and Verification
- Agility™ Handel-C and PixelStreams for FPGA logic
- Link for Modelsim Co-simulation for Verification

Agility™

Orion VPU ESL Tool Flow



Lessons Learned

- All Tools have Strengths and Weaknesses
  - Know their Limitations
  - Work with their Strengths
- Do Not Underestimate Benefits of Tool Integration
- Good Partitioning is Key but can be Challenging
  - Early enough for head-start SW
  - Late enough to avoid big impacts from changes
- Automated generation is faster than direct implementation

Acknowledgements: Preston Faiks, Tim Gallagher

Looking Forward

- C-style Languages Provide Proven, Efficient Balance of Abstraction, Visibility, Performance
  - Handel-C Impulse C
- Matlab Tools Direct to HDL from Algo Enviro.
  - AccelDSP SynplifyDSP
  - Simulink HDL Coder
- Automated translation from requirements and system modeling (UML/SML) to Matlab/Simulink
- Tools greatly matured in recent years
- Continued tool use and feedback leads to further improvements

Continuous learning, innovation and improvement