

Smart Payload Development for High Data Rate Instrument Systems

Paula J. Pingree, Charles D. Norton
Jet Propulsion Laboratory (JPL)
California Institute of Technology

Outline

- Definition of Terms
- Smart Payload Motivation
- MARVEL/MATMOS Application
 - Overview and On-Board Science Data Processing
 - MATMOS Baseline Design Trade (Historical)
 - Features of the Virtex-4 and the APU-FPU System
 - Results of the Virtex-4FX12 with APU FFT
 - A MATMOS Computer Concept
- SVM Application
 - Task Summary
 - Technical Approach
 - Synthesis Results
- Summary

Definition of Terms

FPGA	Field Programmable Gate Array
SoC	System On a Chip
MARVEL	Mars Volcanic Emission and Life
MATMOS	Mars Atmospheric Trace Molecule Spectroscopy
FTIR	Fourier Transform Infrared (Spectrometer)
FTS	Fourier Transform Spectrometer
APU	Auxiliary Processing Unit
FPU	Floating Point Unit
FFT	Fast Fourier Transform
SVM	Support Vector Machine

Smart Payload Motivation

Assertion: On-board computation has become a bottleneck for advanced science instrument and engineering capabilities.

Smart Payload Definition:

- 1) Localized instrument “smarts” that offload the flight processor of extensive computing cycles, simplify the interfaces, and minimize the dependency of the instrument on the flight system
- 2) Smart development processes that emphasize design flexibility, scalability, and robustness, and significantly increase instrument performance while reducing cost, risk, and preserving schedule

Target Platform:

- Xilinx Virtex-4 FPGA with embedded PowerPC405 processor

Benefits include: better performance, lower cost, reconfigurability

MARVEL/MATMOS Overview

- Proposed 2011 Mars Scout Mission - *not selected*
- MATMOS - Solar occultation FTIR



Spectrometer

- for detection of trace gases: CH_4 , N_2O produced by life or volcanism
- volumes of raw infra-red spectrum data in 3 minute bursts during Martian sunrise & sunset
- remaining 112 minutes of orbit for science data processing and compression

for downlink

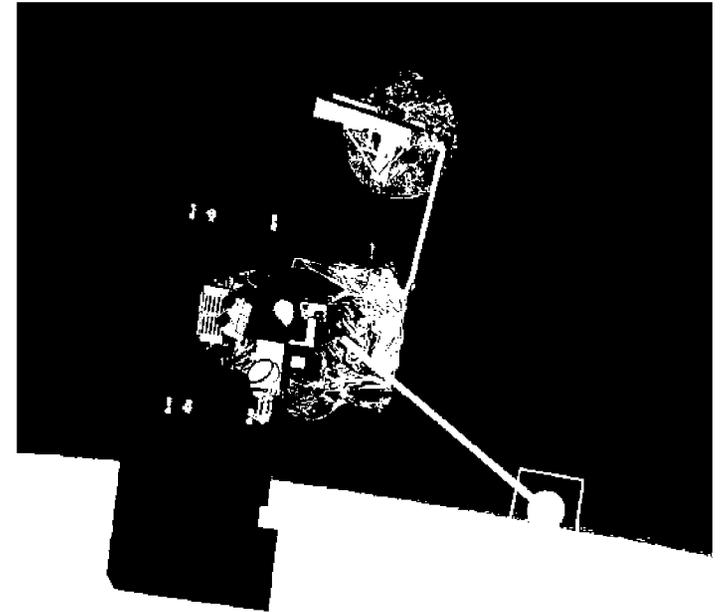


Sun

Earth

Sunrays

Spacecraft



On-Board Science Data Processing

- Two 3-minute occultations produce **691 Mbytes** of data
 - 2 Gbytes on-board storage meet mission requirements
 - Only enough memory for 2 orbits of raw interferogram data
- MATMOS On-Board Data Processing:
 - An observation every 2.5 sec. --> $192 \text{ kHz} * 2.5 \text{ s} = 480,000$ raw samples/channel
 - After re-sampling, $2^{18} = 262,144$ HgCdTe points and $2^{19} = 524,288$ InSb points

– Time-domain data stream re-sampling to path difference domain reduces data volume by a factor of 1.5

$(2 * 480000) / (262144 + 524288) * 1.5 = 1.83$, combined re-sampling data reduction

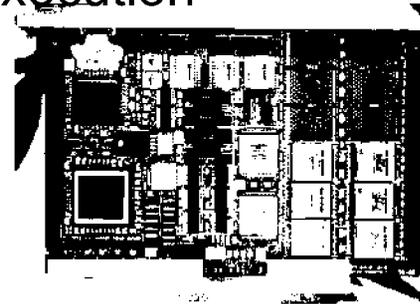
- FFT phase correction by convolution reduces data volume by a factor of 2
- FFT spectrum computation reduces data volume by a factor of 6.1
- Further processing of the spectra (includes averaging scans taken above atmosphere) reduces data volume by a factor of 2

– Standard image compression techniques applied before transmission to S/C computer reduces data volume by a factor of 1.8

MATMOS Baseline Design Trade

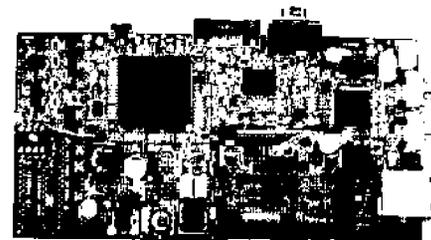
- Simulated raw FTS interferogram data on both Rad750 and Virtex-II Pro platforms
- Virtex-II Pro lagged Rad750 primarily due to smaller cache size and lack of hardware FPU
- A soft-core FPU implemented on the Virtex-II Pro still exhibited high latency associated with floating-point instruction execution

Processor:	RAD750	Xilinx	Xilinx
Operating System:	VxWorks	Linux	Linux
Clock Speed:	133 MHz	300 MHz	300 MHz
Memory:	128 MB	128 MB	128 MB
Software Component			w/Perflib
Reject Dark Interferograms	<1	<1	<1
Interferogram Re-sampling	69	3404	780
Non-Linearity Correction	1	14	4
Phase Correction	42	488	142
Fast Fourier Transformation	15	272	90
Spectra Averaging	2	(10)	(3)
Lossless Compression	1	1	1
Total (112 min available)	130 min	4200 min	1020 min



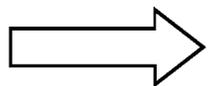
BAE RAD750

- 20 W total power
- 133 MHz CPU
- 16 Gbit addressable memory
- Flight qualified for Deep Impact & MRO



Xilinx (Virtex-II Pro)

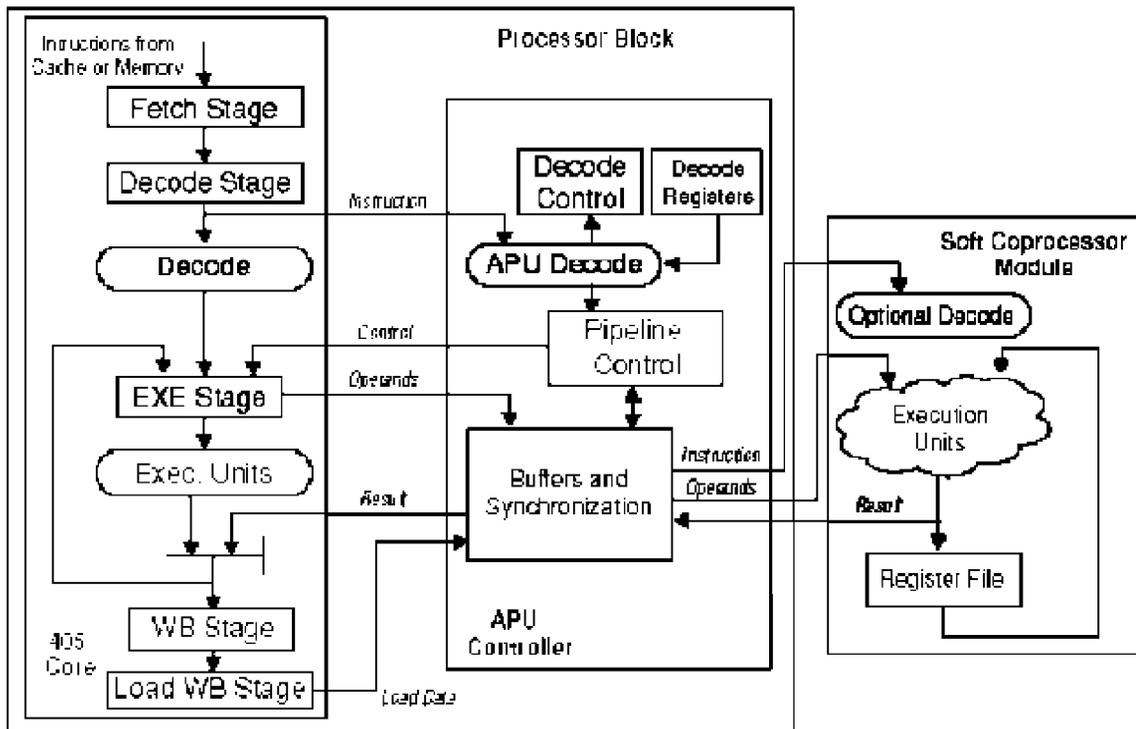
- 5 W total power
- 300 MHz (PPC CPU core)
- No floating-point unit
- Not yet flight qualified



MARVEL baselined 2 Rad750s in their 2011 Mars Scout Proposal

Features of the Virtex-4

- Three families: 1. LX (logic), 2. SX (DSP apps), 3. FX (embedded processing)
- Single or dual embedded PowerPC405 cores, clocked up to 450 MHz
- Auxiliary Processor Unit (APU)
 - integrates directly into the processor pipeline
 - decodes soft co-processor supported instructions
 - ex. APU I/F to soft-core FPU eliminates need for software floating point



Xilinx's APU Floating Point Unit v2.1

Core Specifics			
Supported Device Family	Virtex-4 FX		
Resources Used	Slices	Xtreme-DSP Blocks	Block RAMs
Lite (no div/sqrt)	1100	4	2
Full (with div/sqrt)	1250	4	2
Clock Speeds	-10 part	137.5 MHz	
	-11 part	150 MHz	
	-12 part	170 MHz	

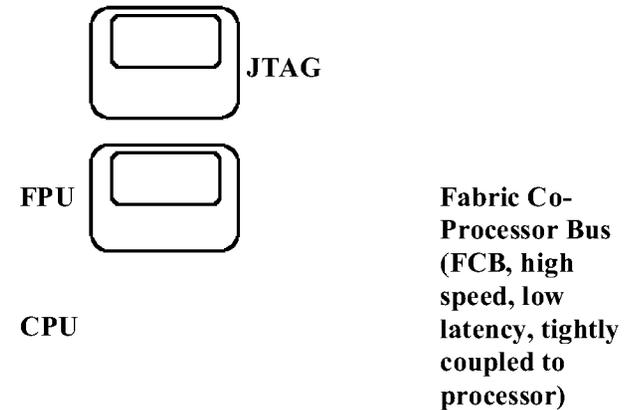
Graphics Credit to Xilinx

The APU-FPU System

- APU-FPU (single precision)
- RS232 UART (terminal I/O)
- 64 MB DDR RAM (for heap + stack space)
- 64 KB BRAM (for all other linker sections, i.e. text)
- L1 cache support for all memories
 - 16 KB data
 - 16 KB instruction
- Compact Flash Interface (for reading large data files)

RESOURCE UTILIZATION OF THE APU-FPU SYSTEM

Resource	Usage	ML403 (V4FX12)	ML410 (V4FX60)
Slices	3,455	63%	14%
BRAM	34	94%	15%
XtremeDSP	4	12%	3%

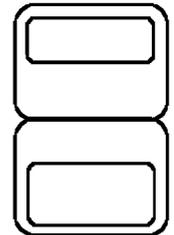


Processor Local Bus (PLB, high speed)

QuickTime™ and a Graphics decompressor are needed to see this picture.

On-Chip Peripheral Bus (OPB, low speed)

DDR RAM



RS232 UART

CF Interface

Results of Virtex-4FX12 with APU FFT

FPGA embedded processing system evaluated in 3 configurations

1. No FPU, software FP emulation through GCC
2. No FPU, software FP emulation through IBM Perflib
3. Hardware acceleration via APU-FPU

Virtex-4FX12 RESULTS

Xilinx Virtex-4FX12 PowerPC405 No OS (GCC 3.4.1, patched) CPU Freq: 200 MHz FPU Freq: 100 MHz	FFT Time / Iteration (SEC)	Speedup (vs. base case)
No FPU	40	base case
No FPU, Perflib	16	2.5
APU-FPU	3.56	11.24

MCP750 RESULTS

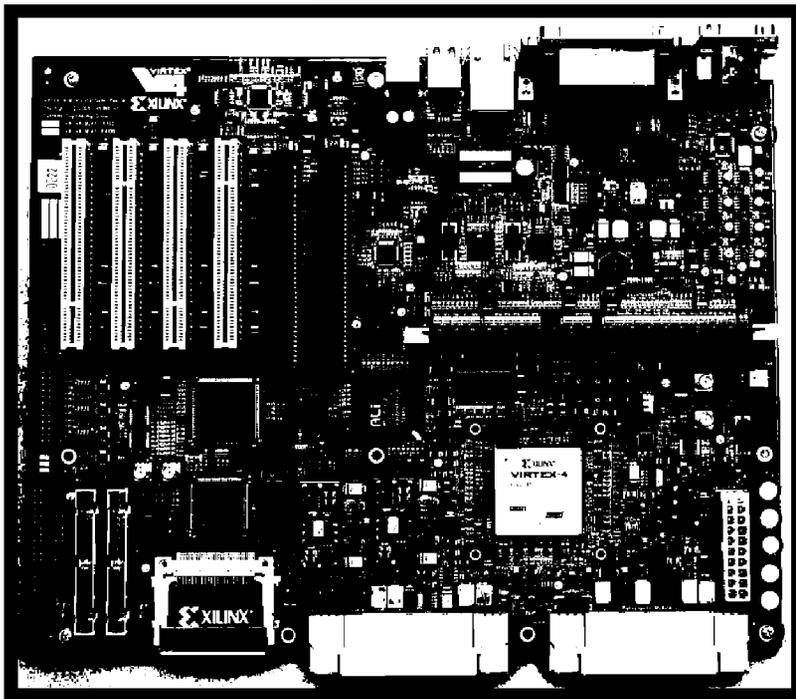
Motorola MCP750 PowerPC750 VxWorks OS / Compiler Freq: 233 MHz	FFT Time / Iteration (SEC)	Speedup (vs. base case)
Hardware FPU	0.637	62.79

Other optimizations to be evaluated include:

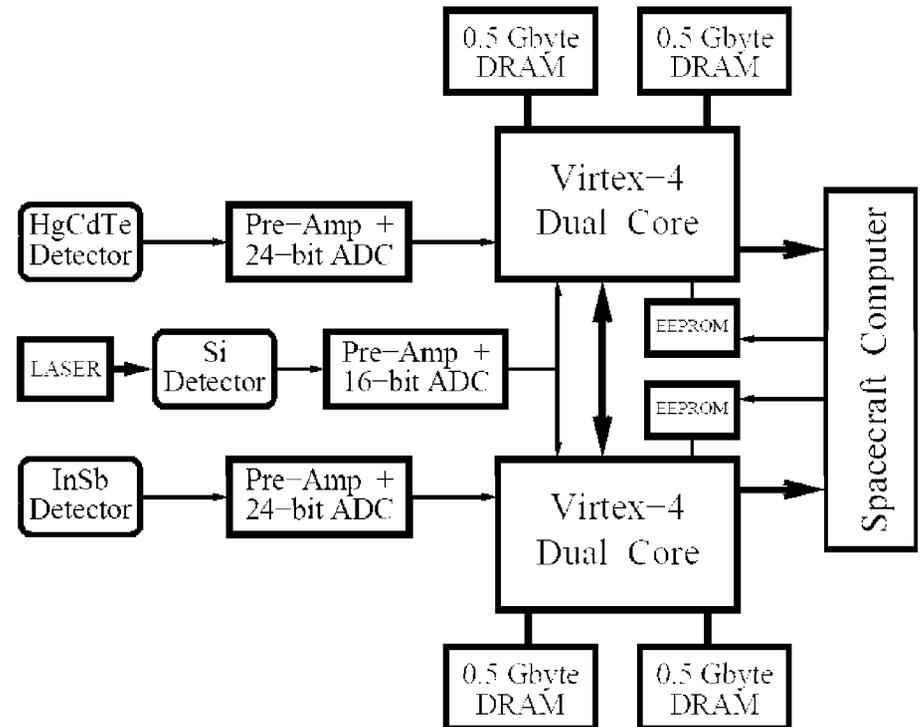
- Support for vector math operations with hardware acceleration
- OS (Linux, VxWorks, XMK) with access to APU co-processors
- Additional co-processors targeted at specific algorithms (example: 5.25 ms 1,000,000 point FFT core, on the market today)
- Optimize APU-FPU utilization
 - Parallelize code to take advantage of pipelined multiply and addition/subtraction
 - Remove data dependencies
 - Evaluate the use of optimized fused multiply-addition/subtraction operation
- Further optimizations in memory
 - Different memory configurations
 - Different cache options

A MATMOS Computer Concept

- MATMOS needs 1.2 x RAD750, 133 MHz
 - 112 minutes available for science data processing
 - RAD750 needs 130 minutes = 1.16×112 min
- Scaled down MCP750 is 3.2x faster than V4FX12
 - MCP750 (233 MHz) is 5.59x faster than V4FX12 (200 MHz)
 - MCP750 is 233 MHz, scale down to 133 MHz to compare ($0.57x$)
 - Real speed factor: $3.19 = 5.59 \times 0.57$
- Use dual core FPGA: MCP750 = 1.6 x dual core V4FX12 ($3.2 / 2$)
- For flight: $1.2 \times 1.6 =$ 1.9 x dual core V4FX12



Xilinx ML410 Board -- a good candidate



Task Summary - SVM Application

Objectives:

- Demonstrate new/smart on-board data processing capabilities, driven by data actively collected by instrument sensors and enabled by Field Programmable Gate Array (FPGA) co-processor implementations, which reduce the most significant bottlenecks limiting new measurement concepts including:
 1. Decision-making
 2. Event detection
 3. Reconfiguration
- Prototype how advanced data fusion algorithms, such as support vector machines (SVM) and others, in signal and image processing can be realized within this architecture to demonstrate improved data processing capabilities and performance

Approach:

- Target Xilinx ML403/410 Development Platforms including Virtex-4FX FPGA with Auxiliary Processing Unit (APU)
- Use CoDeveloper™ C-to-HDL tool set by ImpulseC
- Implement SVM algorithm (legacy C-code) in FPGA and demonstrate performance improvement

Support Vector Machine (SVM) Algorithms

For Phenomena Detection

Classifier
cloud
ice
land
snow
water
unclassified

Co-designed in an FPGA with

PowerPC405 processor and auxiliary controller

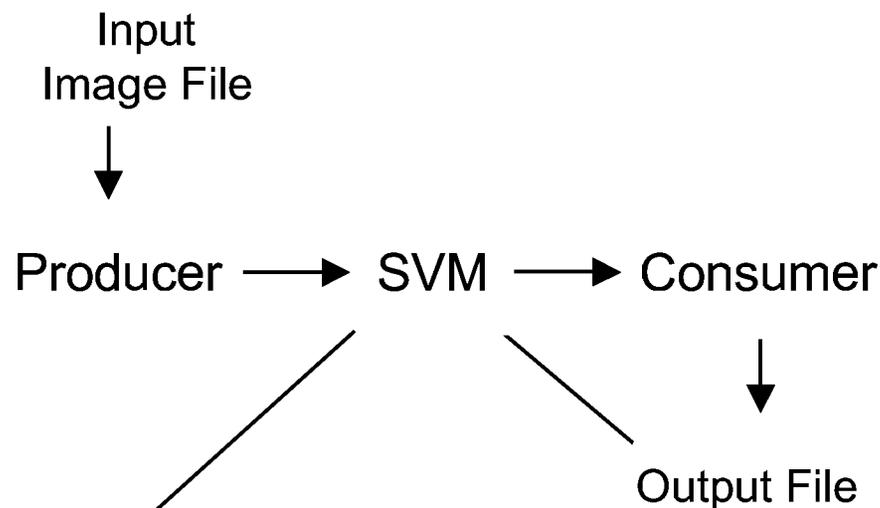


Graphic Credit: Xilinx, Inc.

Technical Approach

■ SVM Application

- Producer in sw.c
 - Reads input image file
 - Streams data to SVM
 - To run in embedded PowerPC405 processor
- Consumer in sw.c
 - Streams data from SVM
 - Writes pixel classifications to output file
 - To run in embedded PowerPC405 processor
- SVM in hw.c
 - Original algorithm + added I/O
 - C-to-HDL auto-translation and optimization via CoDeveloper™
 - Synthesized in Xilinx ISE for execution in FPGA fabric



Original SVM main loop:

```
/* If all reflectances are reasonable, apply the SVM. */  
for(class_index=0; class_index<NUM_CLASSES; class_index++) {  
    PIXEL output =  
        (-svm_biases[class_index]) +  
        refl8 * svm_coefs[class_index][0] +  
        refl21 * svm_coefs[class_index][1] +  
        refl31 * svm_coefs[class_index][2] +  
        refl34 * svm_coefs[class_index][3] +  
        refl41 * svm_coefs[class_index][4] +  
        refl51 * svm_coefs[class_index][5] +  
        refl85 * svm_coefs[class_index][6] +  
        refl110 * svm_coefs[class_index][7] +  
        refl150 * svm_coefs[class_index][8] +  
        refl210 * svm_coefs[class_index][9] +  
        refl213 * svm_coefs[class_index][10];  
    if (class_index==0 || output > best_value) {  
        best_class = class_index;  
        best_value = output;  
    }  
}
```

Synthesis Results

- Simulated in CoDeveloper™
 - I/O read/write
 - Producer, SVM, and Consumer execution
- Pixel classification results:
 - **Total pixels = 857856** ✓
 - (S)now = 146138
 - (W)ater = 356773
 - (I)ce = 99119
 - (L)and = 197374
 - (C)loud = 40971
 - (U)nclassified = 17481
- ImpulseC resource output:
 - 10 Adders/Subtractors (6 bit)
 - 1 Adder/Subtractor (32 bit)
 - 11 Multipliers (3 bit)
 - 1 Comparator (2 bit)
 - 2 Comparators (32 bit)
 - 12 Floating Point Adders/Subtractors (32 bit)
 - 11 Floating Point Multipliers (32 bit)
- HDL auto-generated
- Synthesized in Xilinx ISE for V4FX12 and V4FX60 FPGAs
- **Design fits on V4FX60**

FPGA RESOURCES	V4FX60 (on ML410)
Number of Slices:	8495 out of 25280 (33%)
Number of Slice Flip Flops:	9640 out of 50,560 (19%)
Number of 4 input LUTs:	10495 out of 50560 (20%)
Number of DSP48s:	44 out of 128 (34%)
Maximum frequency:	129.95 MHz

Summary

- Hybrid FPGA technology can deliver breakthrough performance by tightly coupling hardware and software
- Interest in the Xilinx Virtex-4FX FPGA for space flight & instrument applications is growing
- Smart Payload designs for instruments such as MATMOS can meet science data return requirements with more competitive use of available on-board resources
- Smart Payload designs can provide algorithm acceleration in hardware leading to implementation of better (more advanced) algorithms in on-board systems for improved science data return

Acknowledgements

Dmitriy Bekker (Rochester Institute of Technology, graduate student), Jean-Francois Blavier (JPL), and Geoff Toon (JPL) for their contributions to the MATMOS FPGA development effort.

Lucas Scharenbroich (JPL) for his contribution to the SVM FPGA development effort.

Abdullah Aljabri (JPL) for his vision and support of Smart Payload technology development.

Jim Cutts (JPL) for his support of Smart Payload initiatives.

End of File

