



LeonSVF Characterisation Case Study #2

Federico Cordero, Eduard Baumstark, Johan Marx
09th June 2016





Content

- ✦ Project objectives
- ✦ Background on reference SDVF and OBC
- ✦ Integration of LeonSvf on reference SDVF
- ✦ Performance comparison
- ✦ Conclusions



Project Objective

- ✦ Characterize the Leon Emulator Board (LEB) of the LeonSVF as a building block in the context of a flight-representative system simulator (SDVF for MASCOT)

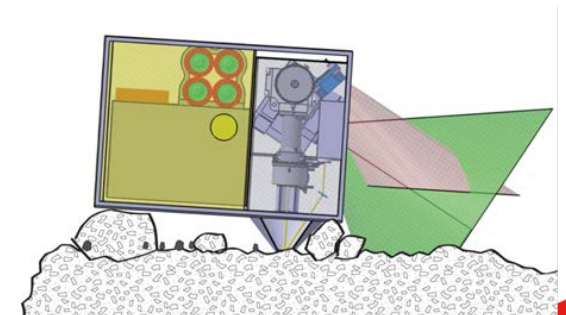
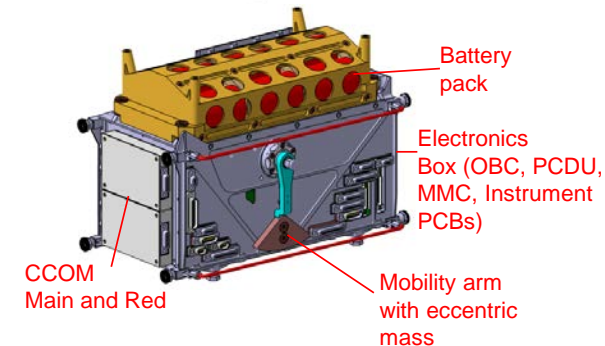
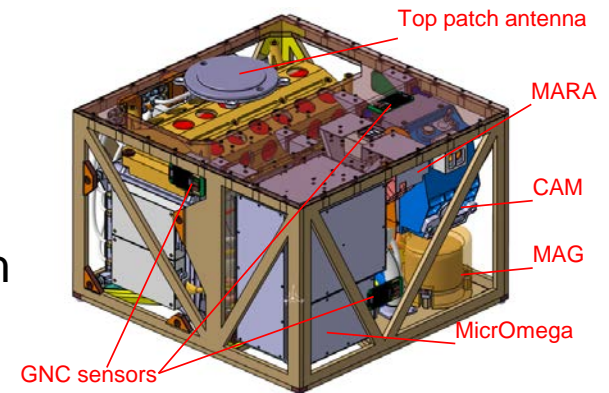
- ✦ Propose potential enhancements for the LeonSVF LEB and its software layers

- ✦ Main steps of the activity:
 - Integrate LeonSVF LEB in the MASCOT SDVF infrastructure

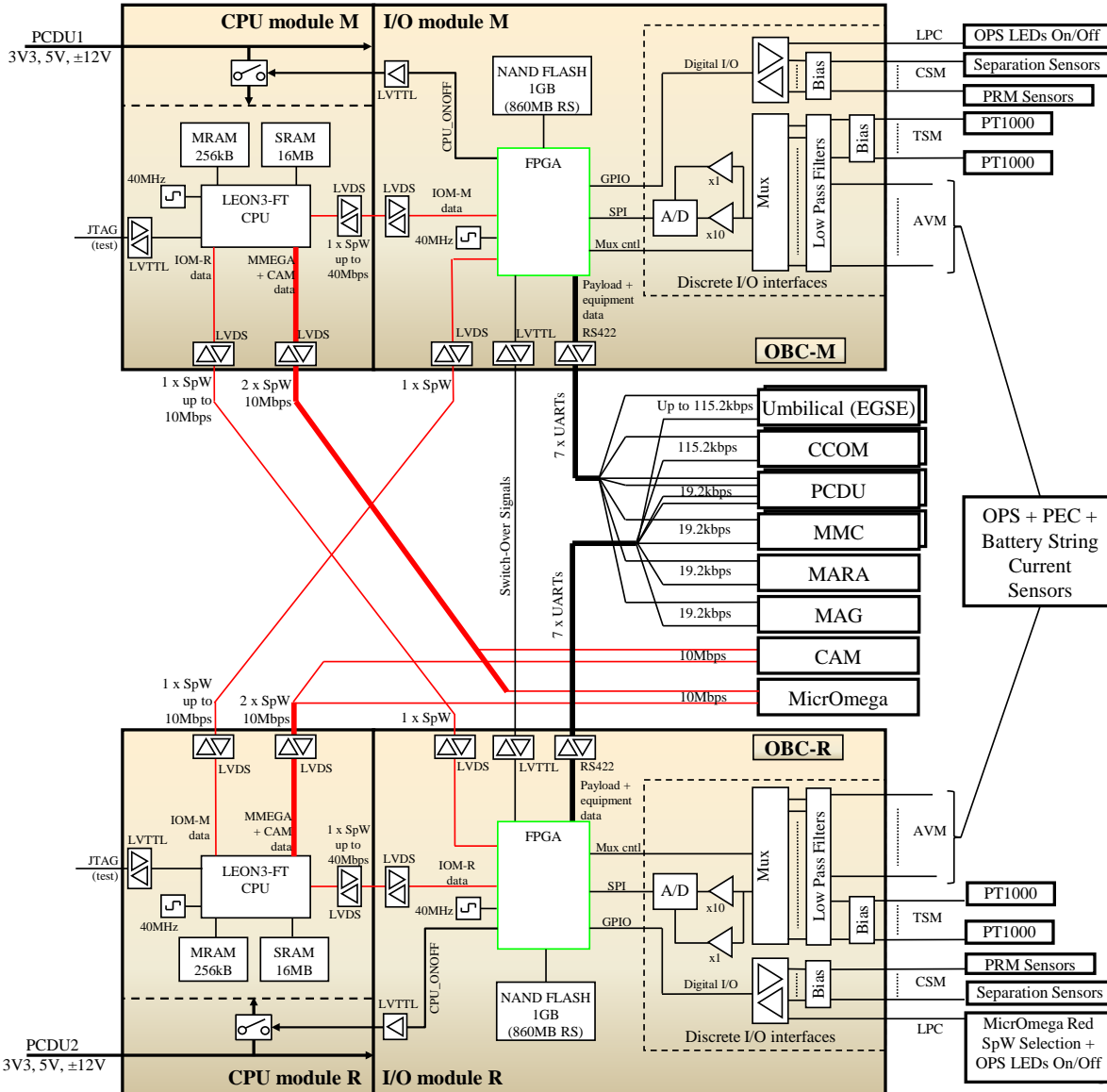
 - Run test scenarios representative of high CPU load and high I/O throughput on SDVF with both LeonSVF LEB and original emulator (TSIM based) and compare results

Background: MASCOT

- ✦ **Mobile Asteroid Surface Scout:** Light and compact lander for in-situ asteroid research: ~10kg, 19.5x29.5x27.5cm³
- ✦ Design and development by DLR in collaboration with CNES as contribution to JAXA Hayabusa-2 mission
- ✦ Launched on 03/12/2014; Arrival on 06/2018; Deployment to asteroid end 2018/ begin 2019
- ✦ Telespazio-VEGA role:
 - ✦ Technical management of the OBC hardware procurement (consortium with Aeroflex/Gaisler for EM and DSI Bremen for QM & FM)
 - ✦ OBC flight software development
 - ✦ Software Development and Verification Facility (SDVF) procurement



Background: MASCOT OBC



Dual redundant, single fault tolerant

- 2 CPU boards in cold redundancy
- 2 IO boards in hot redundancy

CPU board

- Leon3-FT Aeroflex GR712RC at 40MHz (OSW uses only one CPU core)
- MRAM(as PROM): 3DPlus 256kB
- SRAM Aeroflex MCM UT8R4M39 16Mbytes, EDAC protected
- 4 SpW interfaces (2 for IOM cross-strapping + 2 for payload)
- LVTTTLs for inter-board interfaces
- JTAG accessible from E-box

IO board

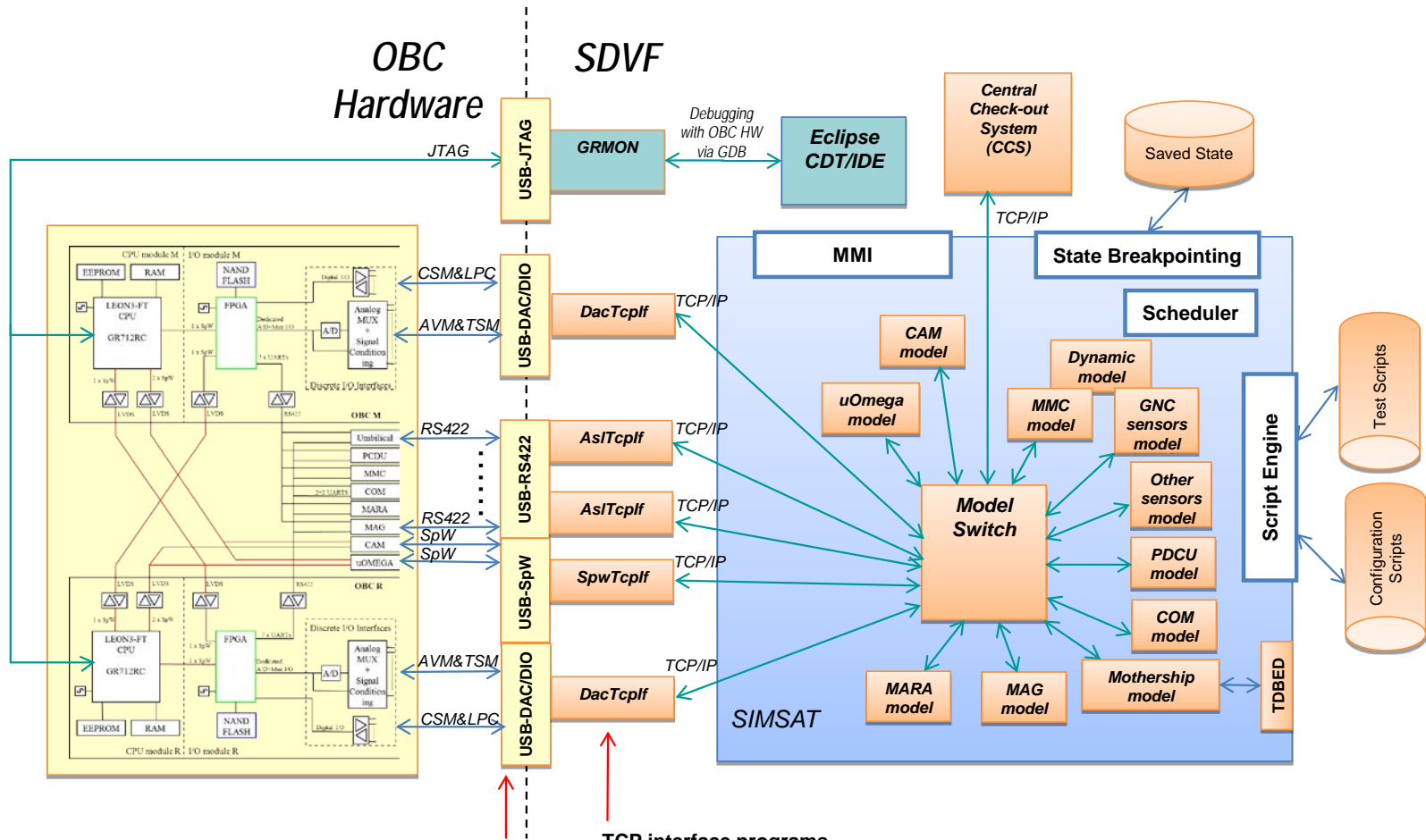
- Mass Memory: 3DPlus 1GB Nand Flash
- FPGA Actel ProAsic RT3PE3000L
- FPGA cores: NF controller with R/S enc/dec, Timer, on-chip RAM, UARTs, SPI and MUX, GPIO, SpWs with router
- 2 SpW interfaces allow access from active CPU to:
 - all IO board resources as RMAP target
 - opposite CPU (in standby) as RMAP target
- 7 Bidirectional RS422 UARTS
- 17 AVM (0..5V or 0..0.5V 12bits)
- 15 TSM (PT1000,-100C/+150C 12bits)
- 4 CSM (Contact sensor)
- LVTTTLs for inter-board interfaces
- JTAG accessible from board only



Background: MASCOT SDVF

- ✦ SDVF includes:
 - ✦ OBC virtual model (TSIM+IO models) fully replaceable by real OBC HW for HIL tests
 - ✦ Full MASCOT simulator with IO models for all payload units and equipment
 - ✦ ESA SimSat kernel + reuse of standard models
 - ✦ Test Script Engine
 - ✦ Central Checkout System (CCS) + TM/TC DB, based on ESA SCOS2000
 - ✦ Development environment: Eclipse + LEON toolchain
 - ✦ I/O cards: UART RS422, Spacewire, JTAG DAC IO
 - ✦ Host computer (HP Z400)
- ✦ SW validation based on Automated Regression Testing scripts executed on two SDVF configurations:
 - ✦ Full Virtual
 - ✦ Processor in the loop
- ✦ Dual use as EGSE

Background: SDVF with Processor In The Loop

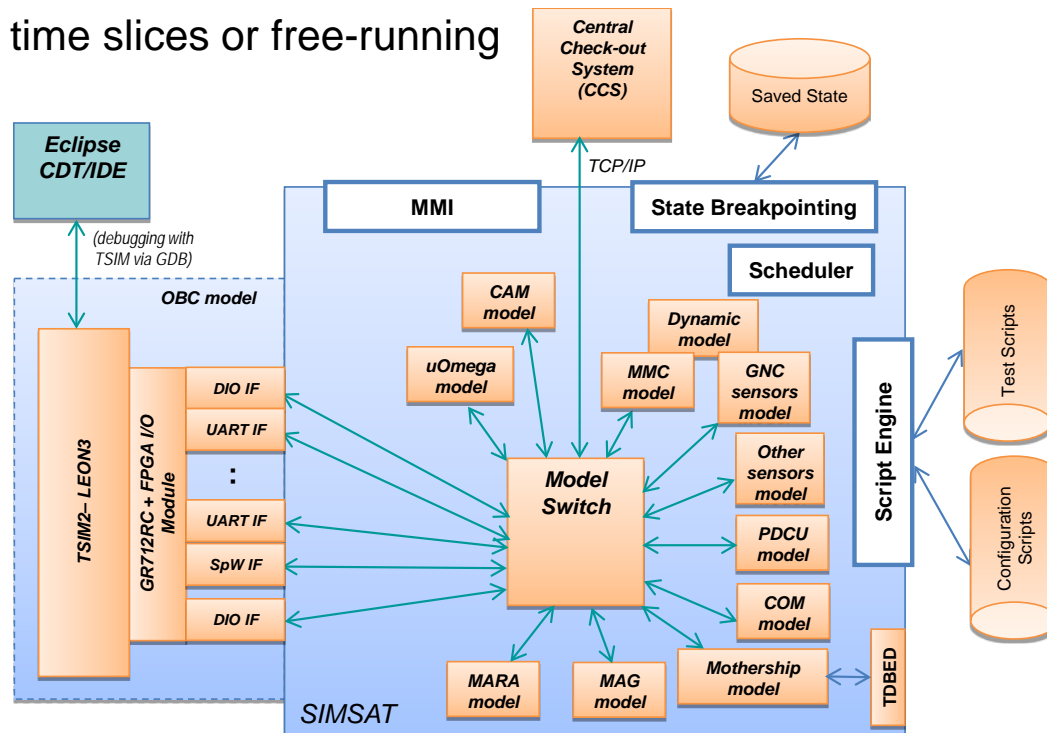


I/O Interface cards:
 USB-RS422
 USB-Spacewire
 USB-DAC IO
 USB-JTAG

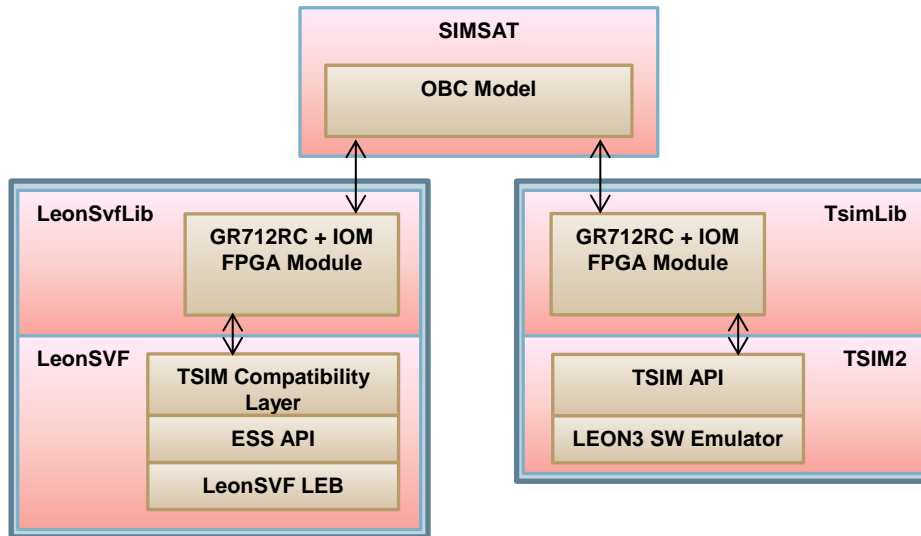
TCP interface programs
 for HW cards I/O with
 simulation models

Background: SDVF Full Virtual Configuration

- ✦ **Reference environment for comparison with LeonSVF LEB**
- ✦ Virtual OBC: TSIM LEON3 instruction emulator + models of CPU & FPGA I/O devices developed by TPZV (IP cores for GRSPW2, UART, Timer, Reconfiguration Logic, SPI/MUX, NF controller etc)
- ✦ Virtual OBC interfaces connected to models of MASCOT equipment/payloads via SW callbacks
- ✦ Simulation execution in time slices or free-running



LeonSvf integration on MASCOT SDVF



Approach:

- ✦ TsimLib: Original library embedding GR712RC SOC + IOM FPGA models
- ✦ LeonSvfLib: New library derived from TsimLib, replacing TSIM2 and selected I/O core models with LEB cores + adaptations
 - ✦ LEON3 IP core with simulated frequency set to 40MHz: Includes GRFPU as GR712RC, cache configuration different
- ✦ MASCOT OBSW modified to account for the changes
- ✦ Most of LeonSvfLib adaptations integrated back into TsimLib, to enable running the same OBSW, for comparison purposes

- ✦ **Additional study objective:** ESA SPW IP core integration as replacement of GRSPW2 software model



SDVF Changes

MASCOT SDVF changes to integrate LeonSvf (TsimLib → LeonSvfLib):

- ✦ Remapping of the IP core control/status registers to the APBSLAVE addresses available on LEB

- ✦ Adaptation of the event scheduling:
 - ✦ Some GR712RC core models works based on periodically scheduled events
 - ✦ With TSIM, SDVF uses high frequency events to increase their fidelity
 - ✦ If left un-changed considerable performance degradation due to high frequency suspension of LeonSvf
 - ✦ Solution: Reduce frequency where possible + Use SIMSAT events scheduling for events with $T > 20\text{ms}$ (called outside the emulator time steps)

Event	MASCOT SDVF Freq	LeonSVF Freq	Type	Scheduler
PM SpW Link State	10us	5ms	Periodic	LeonSVF
PM GPIO IRQ (new event)	N/A	10us after the IRQ is raised	Event	LeonSVF
IOM SpW Link State	10us	5ms	Periodic	LeonSVF
GRSPW2	End of Packet TX	End of Packet TX	Event	LeonSVF
MUXCTRL	100ms	100ms	Periodic	LeonSVF
AHB TCP IF to SDVF	10ms	10ms	Periodic	SIMSAT
FIFOUART	UART TX event	UART TX event	Event	SIMSAT
GPTIMER	Prescaler setting (1 us)	Next underflow (IOM only)	Event	SIMSAT
SWITCHOVER Prescaler	104.8576 ms	104.8576 ms	Periodic	SIMSAT
NANDCTRL	End of operation	End of operation	Event	LeonSVF



SDVF Changes

MASCOT SDVF changes to integrate LeonSvf (TsimLib → LeonSvfLib) - *continued:*

✦ LEB GPIO integration:

- ✦ Replacement of the GR712RC simulated GPIO cores with the LEB embedded GPIO core
- ✦ GRGPIO1 and GRGPIO2 ports remapped to the single LEB GPIO (0x80000600)
- ✦ IRQ lines required additional scheduled event to reset the line

✦ Optimization of the size of the PCI data transfers

- ✦ With TSIM, AMBA bus data transfers for GRSPW2 DMA was done via 4 bytes words
- ✦ If left unchanged, considerable impact on LeonSvf I/O time (as done via PCI)
- ✦ Solution: GRSPW2 model modified to transmit the whole data in one block
- ✦ Gained >50% performance for large data transfers

Operation	Real Time	MASCOT SDVF (word) Time Slips	LeonSvf (word) Time Slips	LeonSvf (block) Time Slips
FSW boot	40s	1s	16s	6s
2MB img acq from CAM	2s	2s	7s	3s
2MB img compr(3b) and transmission to TM stores in mass memory	18s	12s	2s	0.5s
2MB img bit packing and transmission to TM store in mass memory	27s	11.5s	14s	9s



SDVF Changes

LeonSvfLib changes for LeonSvf SPW core integration (additional study objective):

- ✦ Integration limited to the 2 SpW links with OBC IO boards because of max 8kB packet size FIFO buffer limitation in LEB SPW bridge (MASCOT uses 32kB SpW packets on instruments)
- ✦ LeonSvfLib OBC model modified to use direct ESS API (sendPacket(), getLinkStatus(), setStart(), setSpaceWireIf())



TSIM Compatibility Layer and ESS API Changes

Minor changes / corrections:

- ⇒ TSIM event registration and event queue made compatible to TSIM2 with 64 bit data structure pointers
- ⇒ Modified the TSIM dma_read/write to ensure the 2048 bytes limitation of the ESS patch/dumpMemory is enforced (necessary for DMA large data block transfers)
- ⇒ TSIM initialization of the ESS API to include SRAM as default
- ⇒ Fixed crashes of the TSIM Reset command

- ⇒ ESS Speed Factor setting was modified to accept also non-integer values
- ⇒ ESS Time method getOBCCycles corrected to return the correct cycles number.



OBSW Changes

Changes necessary to run MASCOT OBSW on LeonSvfLib:

- ✦ Adaptation of the PROM boot loader initialisation routine (bdinit.c) to the memory controller available on LEB (MCTRL instead of FTMCTRL in GR712RC)
- ✦ Remapping of the IP core control/status registers to the APBSLAVE addresses available on LEB
- ✦ Remapping of the GPIO and interrupt lines to adapt to the single GPIO core available on LEB (GR712RC has 2 GPIO cores)
- ✦ GRSPW2 OBSW drivers adapted to force cache miss during read operations due to missing bus snooping (Note: DMA data areas were at the end mapped to the I/O exclusion area, which is non-cacheable, so this change proved not to be necessary)

Development of a new OBSW driver for LeonSvf SPW core (additional study objective):

- ✦ Use of API developed by Airbus, adapted to MASCOT OBSW



Performance Evaluation

Approach:

- ✦ Identical OBSW and test cases executed on SDVF with LeonSvfLib and TsimLib
- ✦ Performance defined by Speed Factor (SF) = elapsed ST / elapsed PT where Processing Time (PT) is the time the simulator requires to advance by Simulated Time (ST)
- ✦ Simulation advances in time slices of 20ms of ST allowing evaluation of SF with fine granularity at different steps of the test



Performance Evaluation

Test scenarios:

- ✦ Two operational scenarios taken from the OBSW validation tests were used:
 1. CAM instrument image collection, compression, storage and retrieval
 2. MMEGA instrument science subcube + system images collection, compression, storage and retrieval
- ✦ They have several steps, which are representative of low/high CPU load and low/high I/O throughput (via SpW links)
- ✦ All test scenarios pass through the OBSW and mass memory initialisation before powering on the payloads. This initial part implies several accesses to IO board.
- ✦ CAM image and MMEGA subcube collection implies data transfer of 2MB images through SpW link
- ✦ The compression algorithm is wavelet compression which makes use of 32 bits floating point operations and uses full CPU time
- ✦ Compression is followed by data storage into the two hot redundant mass memories and retrieval for downlink



CAM Test Results

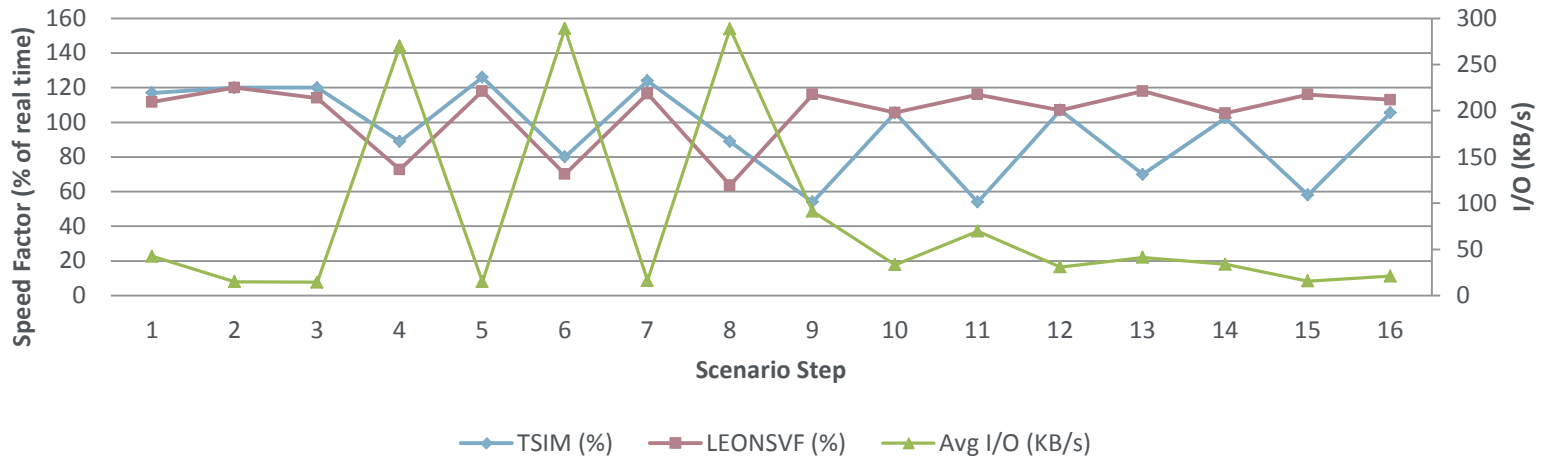
Scenario Step	#	TSIM			LEONSVF		
		speed factor	CPU Load (%)	avg I/o (KB/s)	speed factor	CPU Load (%)	avg I/o (KB/s)
SetupTest	1	116.931	100	42.556	111.675	100	42.539
Power on the CAM	2	120	17.255	15.242	120	28.627	14.681
Reset Img Buf 0	3	119.992	18.039	14.669	113.989	29.804	14.187
Acquire Image in Buf 0	4	88.888	18.039	270.470	72.727	29.804	269.196
Reset Img Buf 1	5	125.982	18.431	14.904	117.993	30.196	14.886
Acquire Image in Buf 1	6	80	18.431	269.901	70	30.196	307.740
Reset Img Buf 2	7	123.997	18.039	17.475	116.503	33.333	14.870
Acquire Image in Buf 2	8	88.889	26.667	269.811	63.636	40.392	307.670
Compress lmg 0 with bPacking MODD 16 and store in MM	9	53.997	100	126.957	115.986	100	55.319
Wait for bPacking MODD 16 download	10	105.448	17.647	33.266	105.633	28.627	33.365
Compress lmg 1 with bPacking MODD 14 and store in MM	11	53.997	100	104.741	115.991	100	34.722
Wait for bPacking MODD 14 download	12	106.952	17.647	30.806	106.951	29.019	30.918
Compress lmg 2 with RevComp and store in MM	13	69.992	100	63.515	117.991	100	19.227
Wait for RevComp download	14	102.564	17.647	33.807	105.263	29.019	33.843
Compress lmg 1 with 4bdata and store in MM	15	57.997	100	16.038	115.993	100	15.266
Wait for 4bdata download	16	105.541	17.647	20.999	112.994	28.235	20.992

⚡ Note: OBSW CPU load differs by ~10-12%. After aligning Leon3 cache configuration, the difference reduced to 6%

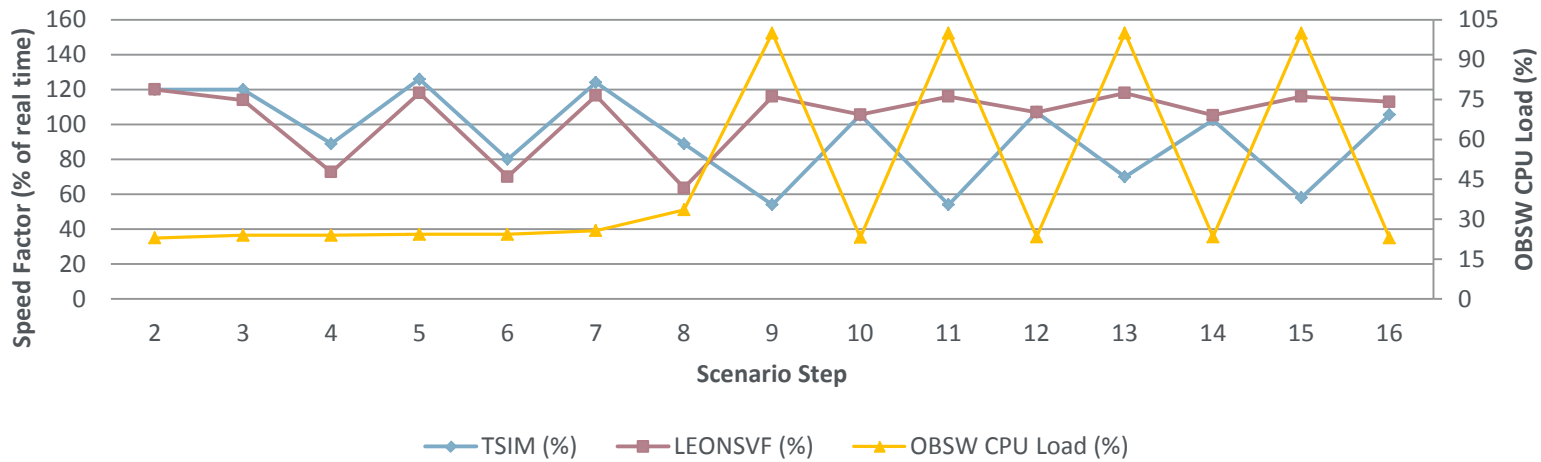


Performance Comparison Results Summary

Speed Factor vs IO



Speed Factor vs OBSW CPU Load





Conclusions

✦ **LeonSvf integration into MASCOT SDVF**

- ✦ After a series of modifications of MASCOT SDVF and the MASCOT OBSW, the LeonSVF LEB was successfully integrated
- ✦ All MASCOT subsystems could be transparently controlled by the OBSW without any known issue or restriction
- ✦ The existing TSIM compatibility layer significantly simplified the integration

✦ **Performance comparison:**

- ✦ The speed factor of the SDVF varied between 1.2 and 0.5 at high loads, both on the TSIM and LEB
- ✦ At higher CPU loads the LEB-based SDVF performs better than TSIM-based SDVF (1.0 versus 0.5), while at higher I/O loads the performance reverses
- ✦ In normal load cases, the two platforms have similar speed factor at around 1.2



Limitations – Possible Improvements

✦ Representativeness

- ✦ LeonSvf is likely to require VHDL adaptation for specific SOC architectures
- ✦ Alternatively the OBSW may need adaptation for the LeonSVF architecture

✦ Memory access limitations

- ✦ ESS GDB server is crashing on non 32 bit word based memory access
- ✦ ESS API DMA calls crash with non 4 bytes aligned addresses
- ✦ The GDB client behaviour is strongly limited by the above mentioned limitations and leads often to crashes of the whole ESS API
- ✦ This made integration of LeonSvf GDB server with Eclipse/CDT IDE for debugging impossible

✦ LeonSvf SPW core limitations

- ✦ 8kB FIFO buffer of LEB cradle SPW bridge
- ✦ SPW core interrupt not raised in case of packets larger than 3kB received
- ✦ Other unresolved problems with LeonSvf SPW core / OBSW driver



THANK YOU FOR YOUR ATTENTION

