CAIRS21 (4000135491/21/NL/GLC/ OV)

COTS AI Accelerators in Mixed-Criticality High-Performance Avionics for Reconfigurable Satellites: TPU versus Prominent Embedded Devices, Mitigation Techniques, SW Frameworks and AI/ML Model Uploading

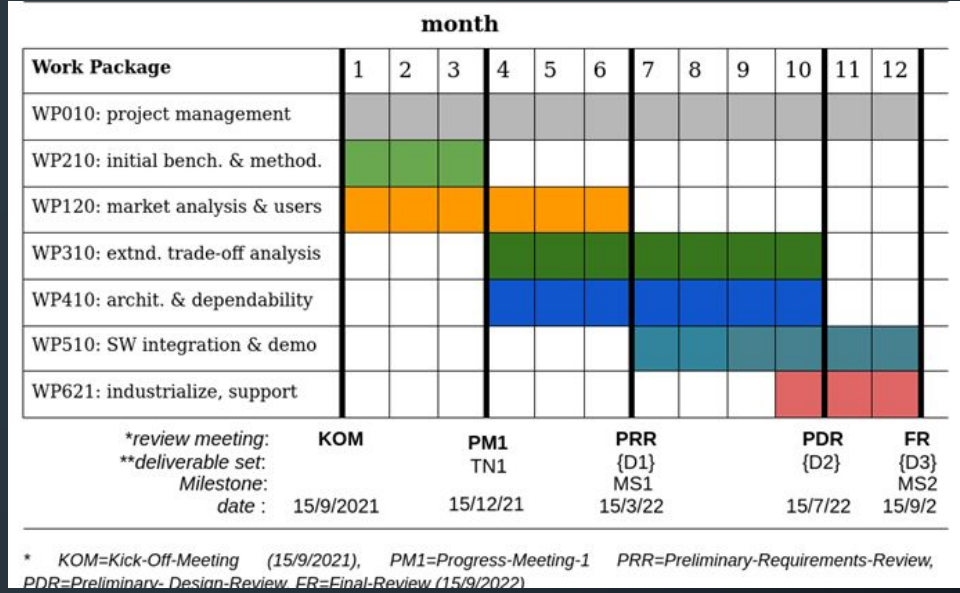
G. Lentaris, V. Leon, C. Sakos, P. Minaidis, D. Soudris (*NTUA*)

S. Vellas, M. Bernou, K. Panopoulou (*OHB-Hellas*)

Contents

- Project status, progress vs PRR
- Overview of results
  - Tradeoff analysis
  - Radiation testing
  - Architecture Integration
  - Market
- Demos
- Conclusion

Status, delta vs prev. review (PRR)



- **Status**: FA, all tasks & deliverables completed
  - promising results, COTS AI HW+SW worth to space
  - dissemination already at 2 conferences
    - ACCEDE@ES and VLSI-SOC@GR (Autumn 2022)
    - advertised at Bremen STExpo & Paris IAC (OHB), Budapest HIPEAC (NTUA). Now targeting journals.

- **Progress** vs PRR
  - Performance: more AI testing of NCS2, FPGA, TPU, GPU… Full consolidated comparison table.
  - Radiation: test at ESTEC, preparation for PSI
  - Architecture: started and completed Zynq+USB accel. HW+SW setup, virtualization. Demos!
  - Market: further discussion/exploration of industry, consolidation, outlook. Initial ICD.

Tradeoff analysis (work summary)

- after PRR: continued hands-on testing/optimizing benchmarks on all accelerators
  - TPU (of course), NCS2 (much more work), GPU (more), FPGA (mostly DPU on ZCU104 & Kria)
- mostly high-level AI frameworks and distinct example AI networks (a lot)
  - but also lower-level style (e.g., Matrix Multiplication on TPU, algorithm partitioning on NCS2)
  - and entire pipelines with space data (cloud segmentation, pose estimation, ship detection)

ble 1: Bibliography benchmarking results (latency and throughput)

Neural Network	ARM A53 (ms)	TPU-USB+X (ms)	TPU DevB (ms)	TPU DevB (FPS)	Jets Nano (FPS)	NCS2+i7 (FPS)	ZCU104 (FPS)
Unet Mv2	190.7	3.3	5.7	175			
DeepLab V3	1139	52	241	4		3.5	
DenseNet	1032	20	25	40		39.4	
Inception v1	392	3.4	4.1	244	76	93	192.5
Inception v2		13.4	20.8	48	54		
Inception v3		42.8	59	17	22		
Inception v4	3157	85	102	10	11	10	30.5
Inception-ResNet V2	2852	57	69	14			23.9
MobileNet v1	164	2.4	2.4	417	80	119	316.7
MobileNet v2	122	2.6	2.6	385	60	75	267.3
MobileNet v3		3					1.5
MobileNet v1 SSD	353	6.5	11	91	25	48	
MobileNet v2 SSD	282	7.6	14	71	39	58	79.2
ResNet-50 V1	1763	49	56	18	38	29	75.7
ResNet-50 V2	1875	50	59	17	36		37.2
ResNet-152 V2	5499	128	151	7	14		15.4
SqueezeNet	232	2	2	500	104	287	305
VGG16	4595	296	343	3	12		21.4
VGG19	5538	308	357	3	10		18.5
EfficientNet-EdgeTpu-S	705	5	5.5	182			119.4
EfficientNet-EdgeTpu-M	1081	9	10.6	94			83.2
EfficientNet-EdgeTpu-L	2717	25	30.5	33			33.3
Yolo tiny V3	190.7				25	46	122.8

Neural Network	ARM A53 (ms)	TPU mini (ms)	TPU DevB (ms)	TPU DevB (FPS)	Nano (10W) (FPS)	Nano (5W) (FPS)	MyriadX (FPS)	Zynq (FPS)
LSTM-JJ	6	-	2	500	5.4		-	700–2000
LSTM-weather	2.22	17.5	1.9	525			-	
MLP1 (FC, 10K neurons)	0.27	7.2	0.7	1428	385			
MLP2 (FC+FC, 3K neurons)	2.7	1.4	0.3	3300	172			
CIFAR10	2	3	0.4	2500	-		1000	10000+
MNIST	1	3	0.5	2000	-		3000	12000+
SHIPNET	22	4	0.6	1666	204		143	770–2100
MobileNet v1	167	17.7	3.6	277.8	53	17		316.7
MobileNet v2	134	14.3	3.1	322.6	66		51	
Inception v1	377	34.3	5.8	172.4	67	15		
Inception v2	601	221	18	55.6	48	10		
Inception v3	1433	721	54	18.5	20	3.7	19	
Inception v4	2951	~1400	103	9.7	2.7	DNR		30.5
EfficientNet (S)	691	21	6.2	161.3				
EfficientNet (M)	1069	57	9.9	101.0				
EfficientNet (L)	2696	233	27	37.0				
ResNet-50 V1					30			75.7
ResNet-50 V2					57			
UNet v2	192	275	26	38.5			12.3	
DeepLab v3	1107	472	206	4.9				
PoseNet ResNet-50	3747	929	93	10.8				
PoseNet ResNet-50	9244	6009	387	2.6				

nchmarking results (MLPs, latency)

Workload	FP Operations	Neural Network	ARM A53 (ms)	TPU DevB (ms)	Myriad X (ms)	ZCU104 (ms)
MLP Low	0.01 M	dense(1000,inputs=10,relu)	0.053	0.3	2.2	0.4
MLP Low	0.48 M	dense(800,inputs=100,relu) dense(400,relu) dense(200,relu)	0.53	0.3	2.3	0.5
MLP Mid	4.6 M	dense(2000,inputs=1000,relu) dense(1000,relu) dense(500,relu) dense(200,relu) dense(5,relu)	6.2	0.33	3.5	1.5

Table 4: Achieved Operations per Second

Workload	FP Operations	TPU DevB (GOPS)	MyriadX (GOPS)	Zynq (GOPS)
MatMult	1.15 G	57	-	-
MLP (low)	0.96 M	3.2	<0.41	-
MLP (mid)	9.2 M	25.6	<2.6	-
CNN (low)	0.1 G	166.7	<21.2	<90.9
CNN (mid)	3.6 G	240.0	<44.2	<295.1
CNN (high)	51.0 G	342.3	<236.4	<962.6
RNN	0.07 K	34.2	-	-

Tradeoff analysis (results overview)

- Consolidated comparison table

accel: workload	vs ARMa53@1.5G	vs MyriadX (USB+PC)	vs JetNano GPU	vs Zynq FPGA DPU
MLP (low)	0.1–0.5x	7x	4x	2x
RNN (mid)	2–3x	(no support)	3-4x	1x
CNN (low)	2–5x	1x	5x	1-0.2x
MLP (mid)	5–20x	5-10x	10–20x	4x
CNN (mid)	20–60x	3-9x	2-8x	2-1x
CNN (high)	5–30x	1–0.5x	1–0.25x	1–0.1x

* mid-CNNs: 0.1-3 GFLOP (10ms-1s on ARM)

* mid-MLPs: ~0.01 GFLOP (1-10ms on ARM)

- TPU best for mid-size MLP & CNN by order of magnitude
  - speed: 10-100x vs CPU, almost 10x vs GPU/VPU, almost 2x vs FPGA (esp. in latency)
  - perf/Watt: ~10x CPU/VPU/GPU, more than 2x vs FPGA (TPU chip =2W / TPU SOM =5W)
- TPU worse in large AI & classical DSP (small on-chip memory, limited ops support)

Tradeoff analysis (extra details)

- **TPU**: models with ARM execution time near 1ms not worth accelerating (TPU overhead = 0.3ms)
- **TPU**: models+weights > 8MB (on-chip memory) begin slowing down the TPU execution (decrease HW efficiency)
  - e.g., single layer MLP > 8MB is executed completely off-chip, TPU becomes slower vs ARM (deceleration!)
  - e.g., CNN sizes 25-43MB (ResNet50_v1, Inception_v4) start performing 3-4x faster on Zynq FPGA DPU
- **TPU**: works only in latency optimization style (batch=1), hence throughput = 1/latency (e.g., unlike GPU)
- **TPU**: "mini" is one order of magnitude slower than "dev board" (co-processing arch integration very important!)
- **TPU**: Google advertises 4 TOPS, but in our own tests throughput < 0.4 TOPS (and small AI/MLPs = 0.04 TOPS)
- **TPU**: (from MLP & matmult analysis): TPU max. accel when model+weights ~8MB, when size >15MB then TPU decelerates ARM, should use <8000 param per layer and prefer AI with more layers but less neurons per layer
- **TPU**: does not support TF op *conv3D* (problem for video AI ?), or *repeatvector* (LSTMs), or *upsampling*
  - i.e., in few apps, TPU required modifying the given model to make work (life difficult for developer)
- **TPU**: supports only TF (life difficult when model given in Pytorch) and is like black-box (no low-level mitigation)
  - e.g., OHB cloud segmentation app was in Pytorch, model was very big → tried but could not port to TPU
- **FPGA**: Zynq MPSOC can fit 2 DPUs (exploit batch=2, but with same latency). The consolidated compari table includes such throughputs (for latency or 1-DPU, the TPU factor can almost double vs FPGA, or increase vs GPU)

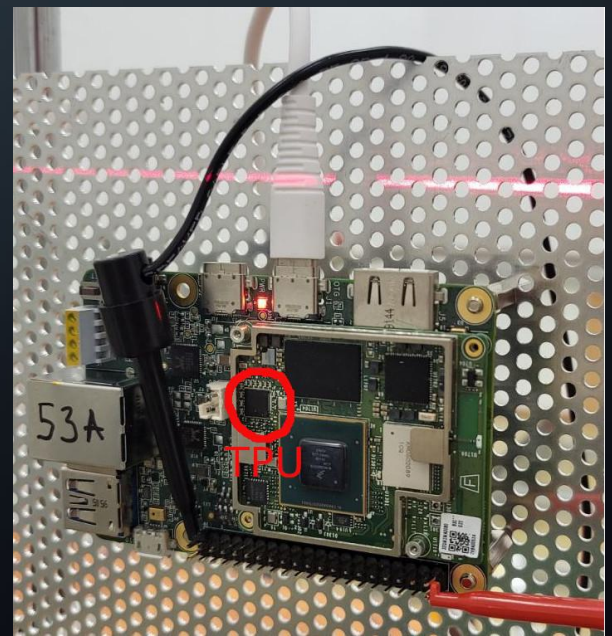
Tradeoff analysis (extra details)

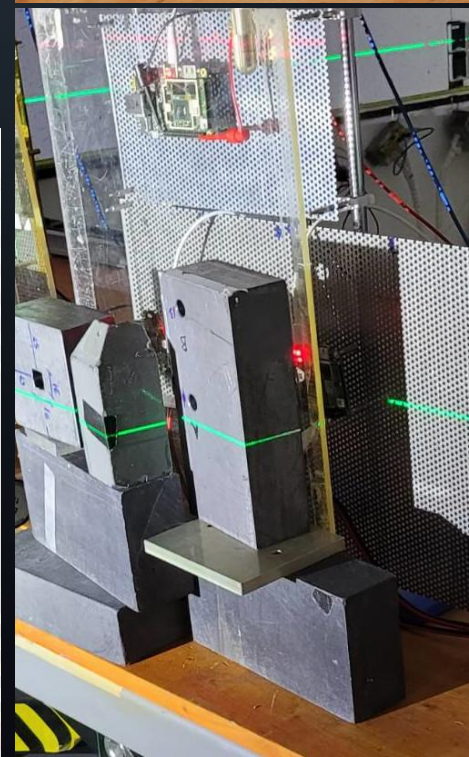
- **FPGA**: DPU achieved similar performance with by-hand HDL (within factor 2x, for small mid-workload shipnet)
- **GPU**: we can achieve an order of magnitude higher speed on bigger AGX embedded device, but at 30W
- **NCS2**: MX vs M2 = order of mag. faster AI due neural eng. MX same Watt as TPU but ~10x slower (1.5-2.5W)
- **NCS2**: currently does not support feedback loop, i.e., RNN/LSTM.
- **NCS2**: supports only FP16 for higher accuracy, unlike DPU+TPU that only support INT8 in HW
- **NCS2**: USB overhead for transferring data (e.g., 15-20ms for 1MPixel RGB image)
- **programmability**: trend for high-level dev in AI/ML, cannot be overlooked in space (AI complexity+evolution)
  - Edge(TPU) vs OpenVino(VPU) vs VitisAI(DPU): all relatively easy to use, starting from Python TF, regularly updated by vendors (important to keep up with AI trends). **EasyUse/LearningCurve = TPU > VPU > DPU.** Support many frameworks (esp. OpenVino). Vitis has less good documentation, more bugs, and more restrictions on supported networks than OpenVino (which however requires model optimizer step). Quantization required (easy for TPU, considerable step for VitisAI, data-agnostic for VPU also due FP16). VPU & FPGA allow low-level coding through other tools (MDK & Vivado). TPU has clearly easiest dev!
  - TPU downside: only few TF ops, only 8bit, only inference, only black-box (no low-level mitigations)
  - NCS2 downside: smaller community, requires more model details when porting/implementing AI
  - FPGA downside: DPU inflexible: big utilization, SoC reconfig. difficult (HW+SW), sophisticated co-location

Radiation test

- **Co-60** facility at ESTEC
- focus = **TPU chip** (not the board)
  - used DevBoard as the host
  - but, chip size very difficult to isolate with bricks (5x5mm^2)
  - but, additional board-level conclusions also nice (e.g., cubesats)

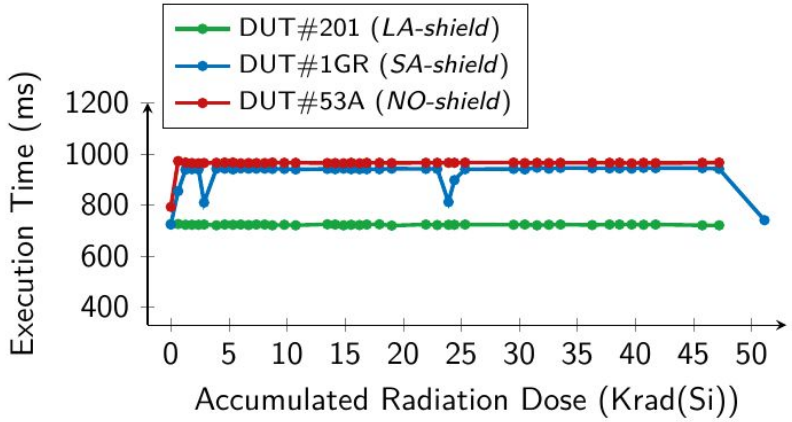
- **setup**
  - 3 DUTs, distinct shielding: *nothing, small hole, large hole*
    - compare, improve conclusions (e.g., w.r.t. PCB)
  - all connected to laptop+PSU (remote control over web)
    - no power cycle (to avoid ancillary component fails)
  - regularly execute 3 AI benchm. on ARM+TPU (per 2-3h)
    - during test, measure V-I, performance, data errors
  - dose rate = 340 rad(Si) / hour



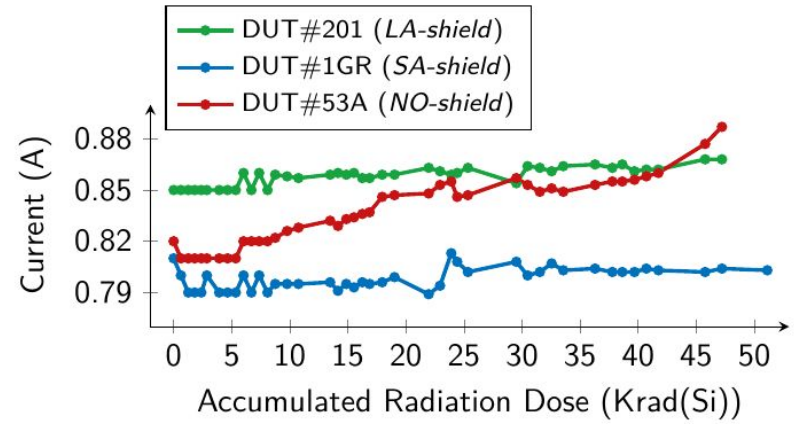


Radiation test

- **50 Krad(Si)** → **test passed** (digital chip TPU)
  - all 3 chips operated correctly throughout the test (139+ h)
    - zero errors, consistent performance (40+ runs per DUT)
  - small current increase on unshielded DUT (9%)
    - attributed to analogue parts of PCB
  - LA-/NO-shield PCBs inoperable after 1st reboot (@47Krad)
    - but SA-shield continued up to 51Krad with 2.7Krad/h
  - TPU, ARM-A53, DDR, eMMC: didn't reach breaking point!
  - annealing + aging test (7-day @ 75^0C) → DUTs still good
    - failed USB restored, throughput decreased by 30-50%
- overall, **TPU chip seems OK for LEO missions**
  - 10-year up to 1000Km. OMERE 5.6.0, incl.=0-98^0, 1-5mm(Al)
  - or 1-3y at 2000Km with 5mm(Al). Need board-level testing.



(a) Avg. execution time on TPU (object detection benchmark)



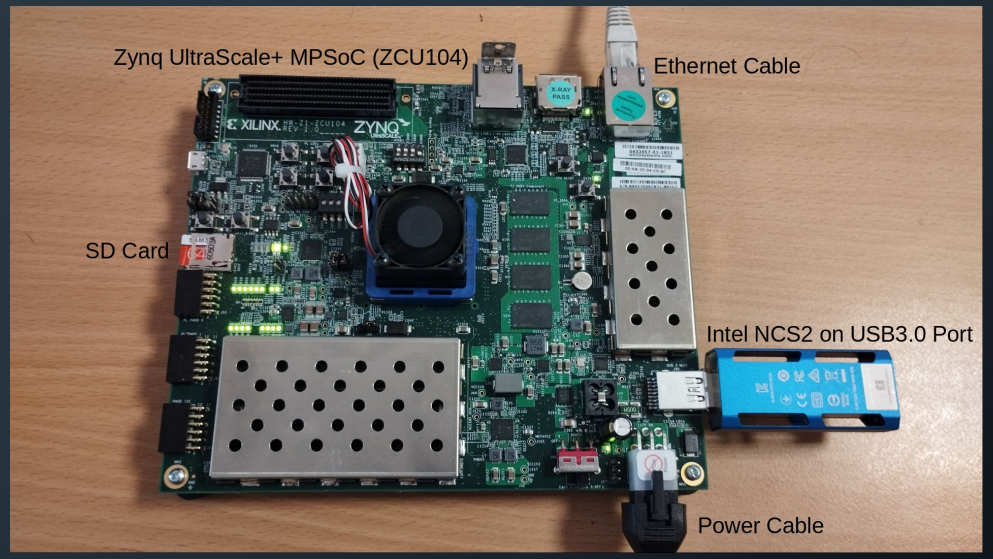
(b) Current during processing (for entire PCB, supply=5V)

Latest[*]: neutron test, promising results! Only 10's errors at 10MeV with flux 3×10^6 n/cm^2/s. Generally, fluence ~7×10^{12} n/cm^2 → cross-section ~10^{-9} cm^2 (with small damage on results)

[*] S. R. L. Rech Junior et al., "High Energy and Thermal Neutron Sensitivity of Google Tensor Processing Units," in IEEE Transactions on Nuclear Science, vol. 69, no. 3, pp. 567-575, March 2022

HW+SW Architecture, Integration

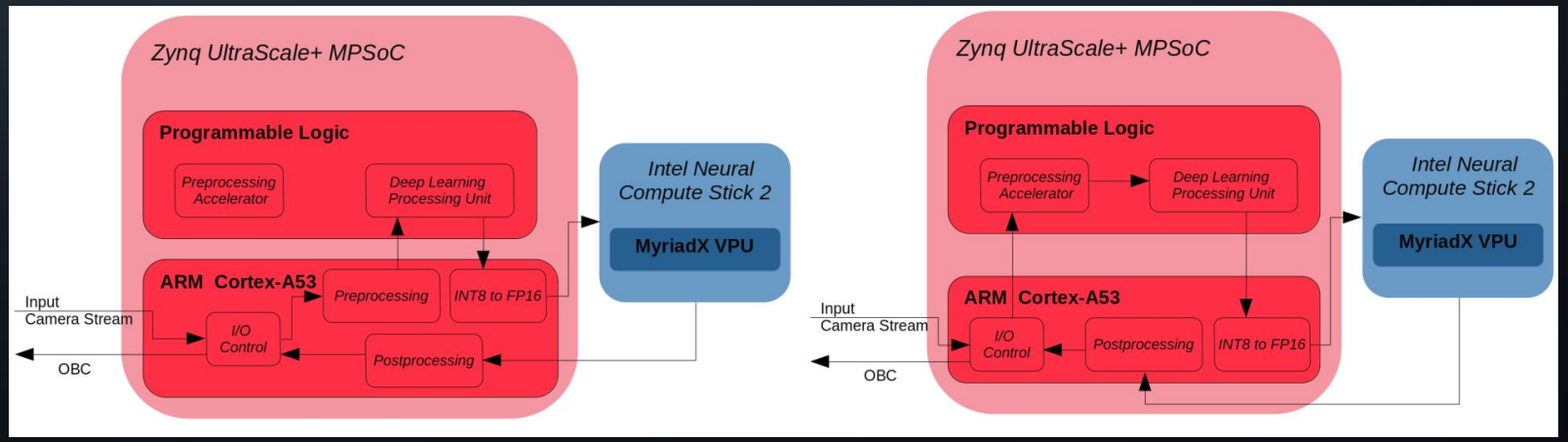
- Successful integration of AI USB Accelerator (NCS2) on SoC FPGA Platform (ZCU104), both in terms of HW and SW
- HW: System communication via USB3.0 port
  - SuperSpeed (5 Gbps) during workload execution
  - internal comm done via Zynq's AXI interconnect



- SW: OpenVINO libraries built for ZCU104 (aarch64 architecture)
  - Utilization of toolchain for RPi3 (ARM Cortex-A53 based single-board computer)
  - Removed unused/incompatible components, included Python bindings
  - Linked with libusb library with no Udev support, to allow for containerization

HW+SW Architecture, AI algorithm partitioning

- Data Preprocessing
  - Small Workloads mapped to ARM Cortex-A53 (Python, OpenCV, NumPy)
  - otherwise, acceleration on FPGA Fabric (C++, Vitis Vision-HLS, VHDL)
- Inferencing
  - First heavyweight CNN layers on DPU (quantization to INT8 precision)
  - Final layers on NCS2 (Partition-Aware Training, FP16 precision)
- Results Postprocessing on ARM Cortex-A53 (single- or multi-threaded)



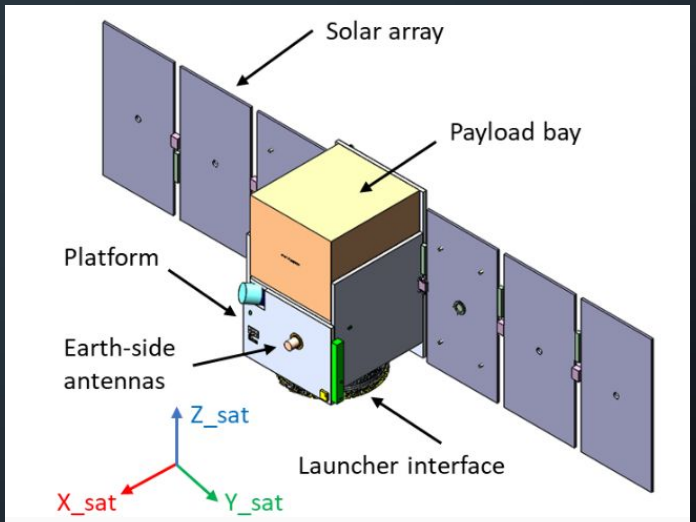
HW+SW Architecture, containerization & orchestration

- No official Docker image for the latest Vitis-AI and Xilinx Runtime
- Custom container based on root filesystem of official, pre-built PetaLinux images
  - but, host OS must run on the same Xilinx Runtime version
- OpenVINO libraries included in the container image
- Kubernetes orchestration using K3s, a lightweight Kubernetes distribution
  - Single binary file of less than 100 MB
  - Control Plane: Typical commercial x86/x64 laptops, computers
  - Worker Nodes: Zynq devices with USB accelerators

Market, Interface Control Document (ICD)

Prepared a first version of an ICD for the TPU SoM

- General Configuration
- Interfaces (Mechanical / Electrical)
- Power limitations and measured performance
- Software
  - Proposed virtualisation approach
- Microsatellite Platform Limitations
  - OHB-Group LuxSpace Triton X Platform as reference
  - typical I/Fs, peak data generation, power, etc



Market, Industrialization Plan, Outlook

- OHB-Hellas strategic interest in technology development towards platform re-usability and AI processing on board
  - Identify Market needs and offerings
  - Efficient AI processing on board
  - Data reduction, Compression
  - Address time-critical applications
  - Targeting to develop a virtualisation approach → baseline for SaaS concept

- CAIRS21 is a critical first step towards this goal
  - Continuation of the work in future ESA-funded activities envisioned

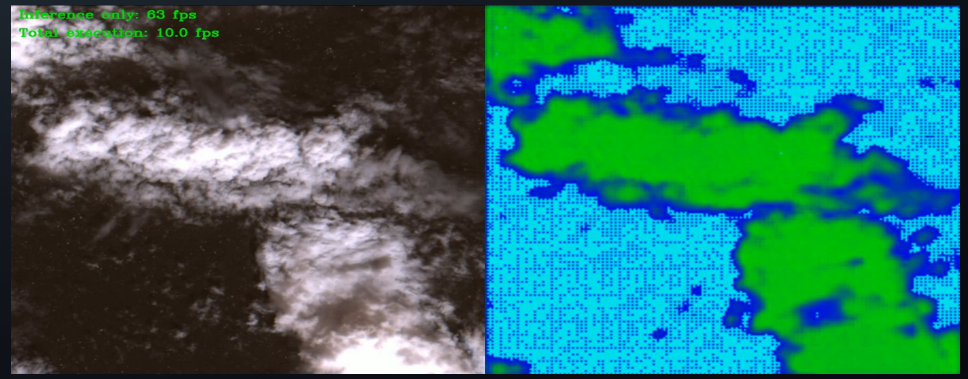
Demos on Edge TPU

Ship Detection

- Model: CNN, input:128x128 RGB, model size 161 KB
- Dataset: space data from kaggle
- Demo: video output with visualised detections
- Performance: Inference ~800 FPS (F=128x128rgb), system ~450 FPS

Cloud Segmentation

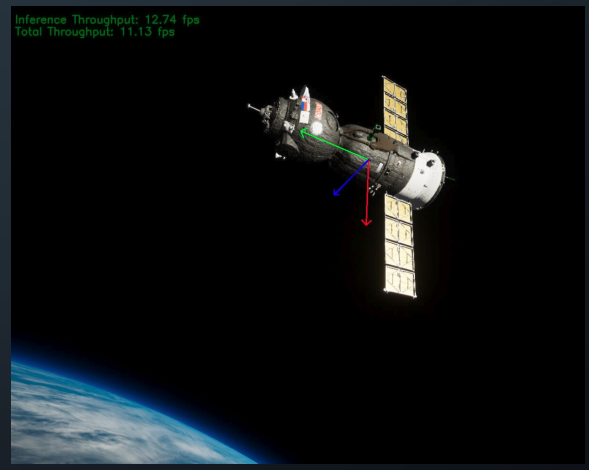
- Model: U-Net, custom built, model size = 7 MB, fits on-chip memory
- Dataset: space data from Sentinel-2, from 1024 to 224 pixels on-the-fly
- Demo: video output side-by-side image & segmentation output
- Performance: Inference ~70 FPS, system ~35 FPS (F=1Mpix), accuracy: 83% validation set



Demos on FPGA+NCS2

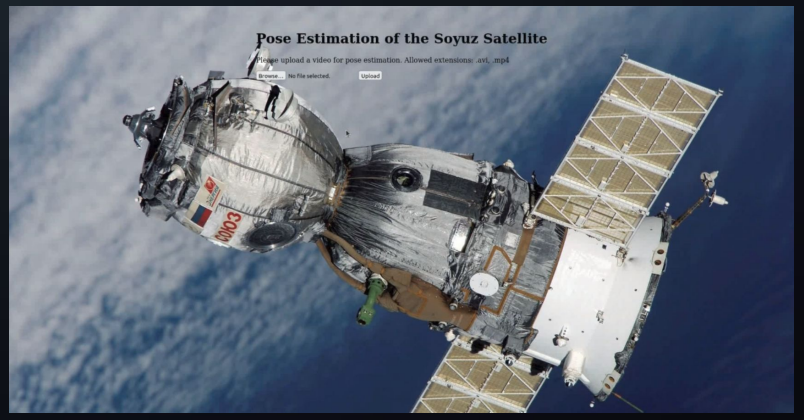
Satellite Pose Estimation

- Model: CNN based on Resnet-50, input: 640x512 RGB
- Dataset: space data of Soyuz (synthetic or real footage)
- Demo: Video with visualization of predicted pose
- Performance: Inference ~13 FPS, system ~11 FPS
- Accuracy: 0.68m localization error, 7.3° orient. error



Containerization + Orchestration (K3s Cluster)

- Application: Web App performing pose estimation on user-provided videos of the Soyuz Satellite
- Remote Deployment on Zynq+NCS2 over internet



Conclusion

- CAIRS21 completed successfully! Immersive study for 4+ engineer/students, results useful for near-future high-performance space avionics design & AI-based missions
- TPU promising, but also has caveats
  - Speed (for mid-size AI, not others), almost: 100x CPU, 10x VPU/GPU, 2x vs FPGA. Power: 2(5)W
  - Programmability: easiest development, but limited to few TF ops (and 8bit inference), black-box
  - Radiation tolerance: TPU chip at 50+ Krad TID (good for 1000+ Km 5+ years LEO)
- Well-integrated mixed-architecture of HW+SW (MPSoC+DPU+USB with SW, for AI)
  - virtualization+orchestration, efficient AI partitioning, high-level frameworks, performance preserve
- Market needs such solution (AI, full HW+SW system, easy dev/use, virtualization)

Future: do more TPU rad/env testing, consider SOM for cubesats or chip for PCB design