

CoRA projects: Model-based HW/SW co-design for reconfigurable platforms

ADCSS2020

October 2020

Tiago da Silva Jorge (GMV)
tiago.jorge@gmv.com

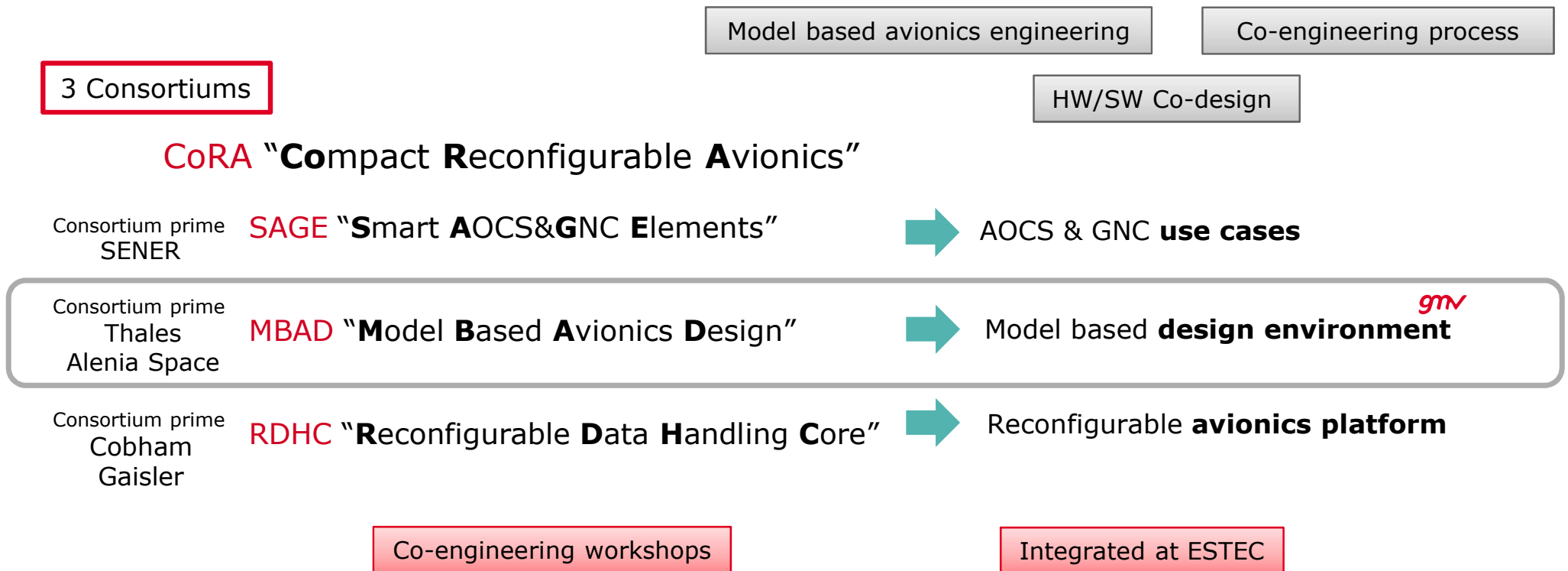
ESA UNCLASSIFIED - For Official Use



European Space Agency

CORA-MBAD PROJECTS

CoRA SAGE & CoRA MBAD & CoRA RDHC



CORA-MBAD PROJECTS

CoRA MBAD & CoRA MBAD for ZynQ 7000

Consortium prime
Thales
Alenia Space

MBAD "Model Based Avionics Design"



Model based **design environment**

gmv

HW/SW co-design toolchain providing functionality to easily deploy functional blocks in either HW or SW implementations, from identical source models

GR740

SpW

BRAVE NG-MEDIUM

taste

Xilinx Zynq-7000
SoC ZC706

gmv

CoRA-MBAD for ZynQ 7000

Adapt toolchain to a ZynQ 7000 SoC target, motivated by low-cost missions that will use platforms based on COTS

CORA-MBAD PROJECTS GOALS



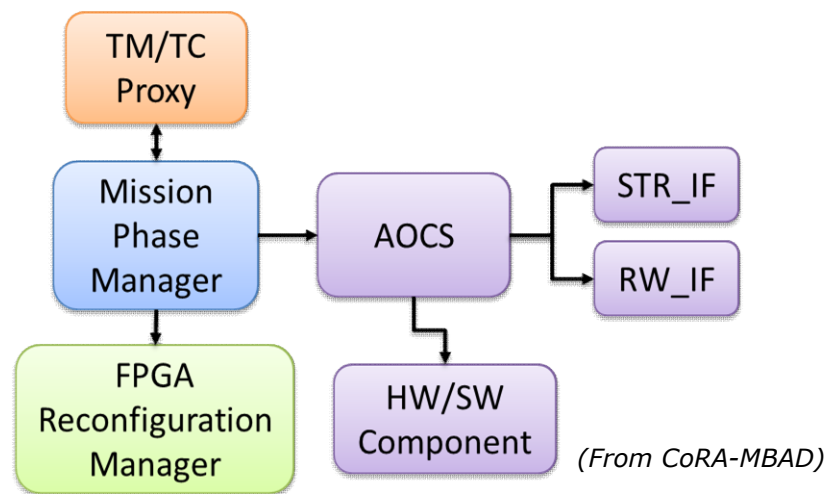
Main objectives:

- Provide functionality to easily deploy functional blocks in either **HW or SW implementations**, from identical source models
- Follow a **model-based approach**
- Achieve a high degree of toolchain **automation** (e.g. via code generation)
- Support **reconfigurable** avionics platforms
- Address representative **use cases**

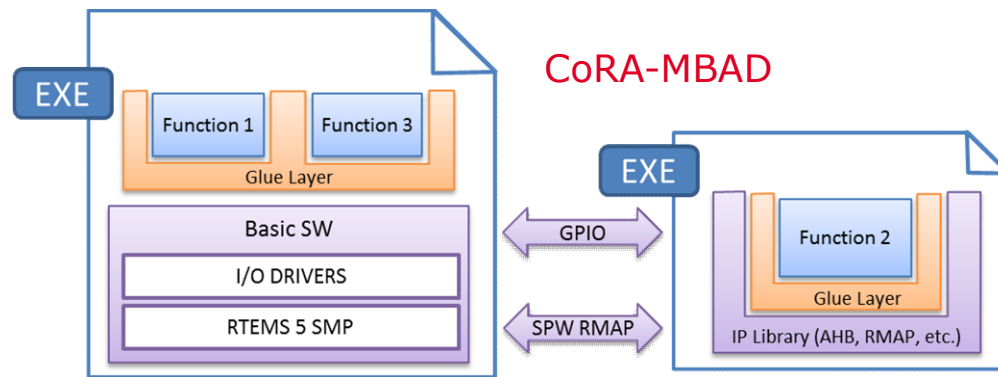
CORA-MBAD PROJECTS

GOALS

- Modeling the mission **phase transitions** in the form of state machines
- **Reconfiguration** of the embedded system by modifying the current functionality of the software and hardware (FPGA) parts
- Production of a **binary** to be deployed on the processor and the different **bitfiles** that correspond to the different **FPGA configurations** defined



CORA-MBAD PROJECTS RECONFIGURABLE TARGETS



(From CoRA-MBAD)



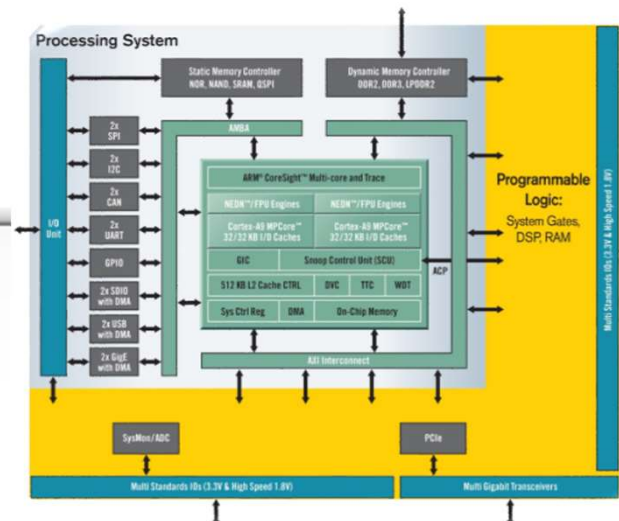
BRAVE NG-MEDIUM

CoRA-MBAD for ZynQ 7000

Xilinx Zynq-7000 SoC ZC706 Evaluation Kit

PS: Dual ARM Cortex-A9 core processors

PL: Artix-7 FPGA



CORA-MBAD PROJECTS CAPABILITIES

MBAD System allows to model the flight software in the form of functional blocks (C, Ada, Simulink, etc.) with provided and required interfaces

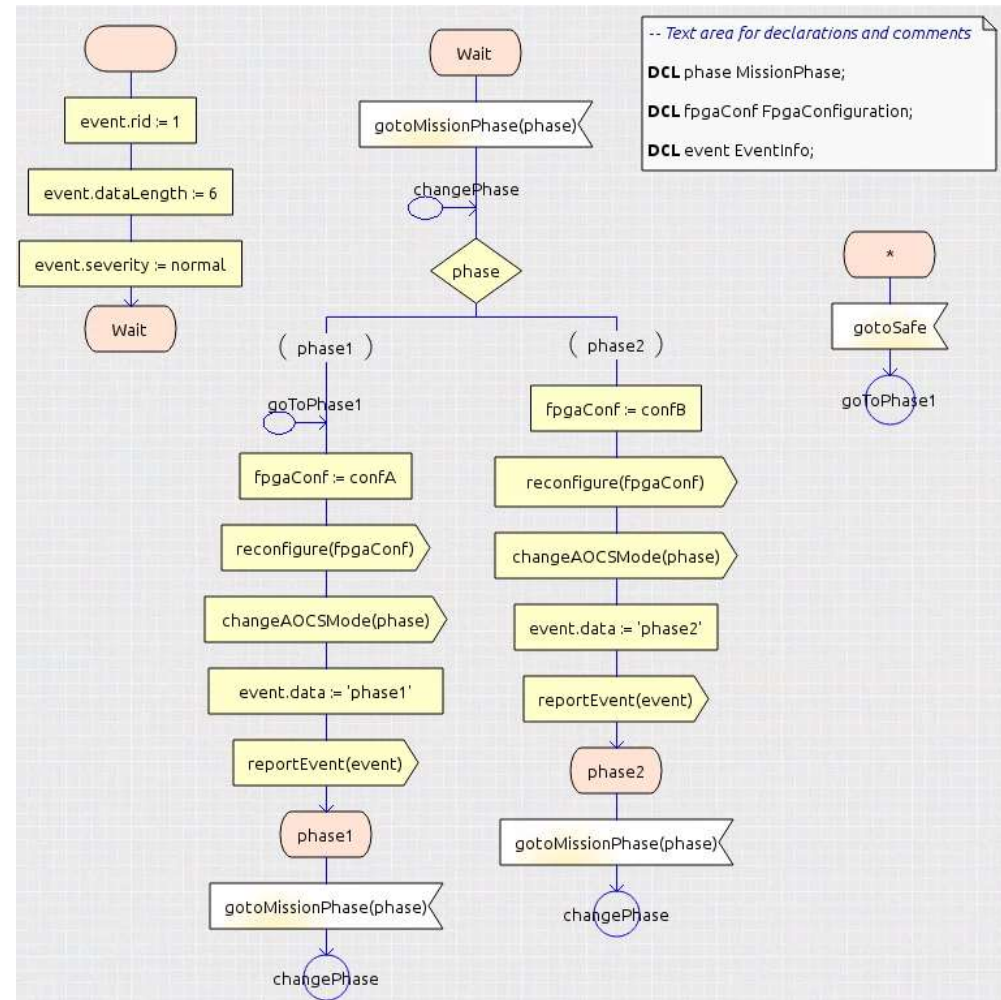
Possibility to **tag a functional block** as active on a FPGA configuration

MBAD System allows to model the **mission phase** management in the form of state machines

SDL (Specification and Description language) via OPENGEOCODE editor

Mission Phase state machines will drive the **reconfiguration** of the system

By notifying a change of mission phase to SW functional blocks
By reconfiguring the FPGA



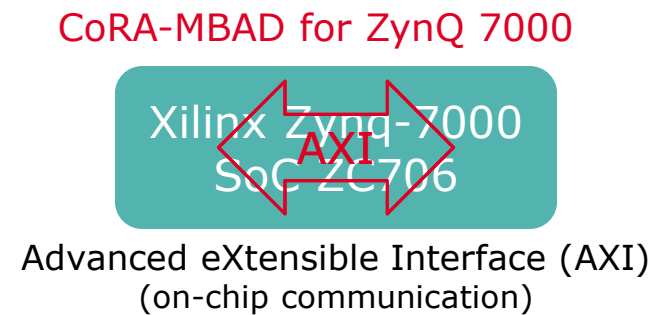
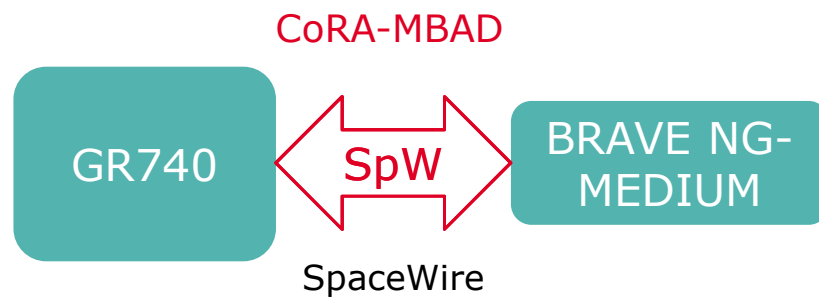
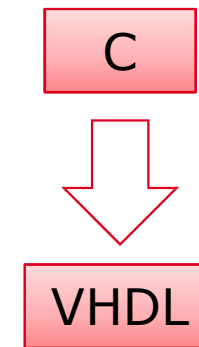
(From CoRA-MBAD)

CORA-MBAD PROJECTS

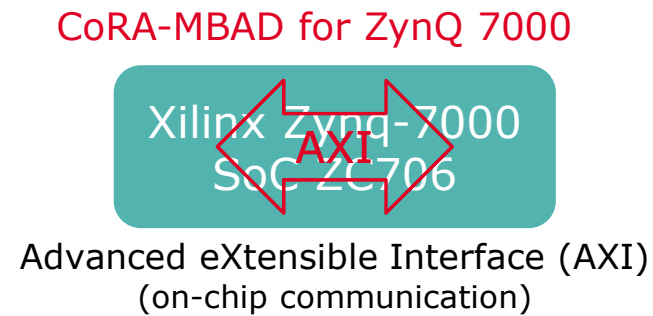
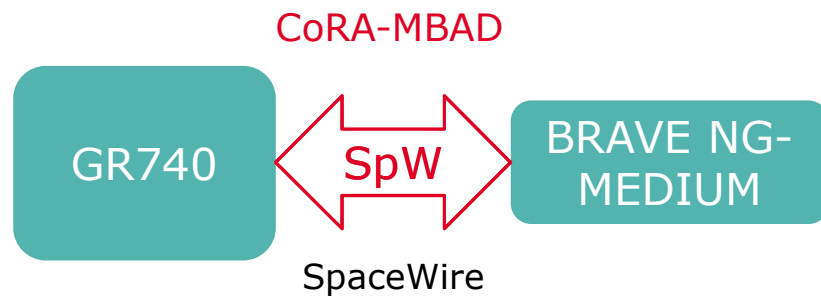
CAPABILITIES

To switch between HW and SW forms, the toolchain implements the **automatic transformation** of C source code (whether manually written or generated by a code generator like those in Matlab/Simulink) into Hardware (VHDL) source files.

To perform an automatic generation of the needed **consistent communication interfaces** supporting the exchange of commands and data between functional blocks executed on the processor and on the FPGA.

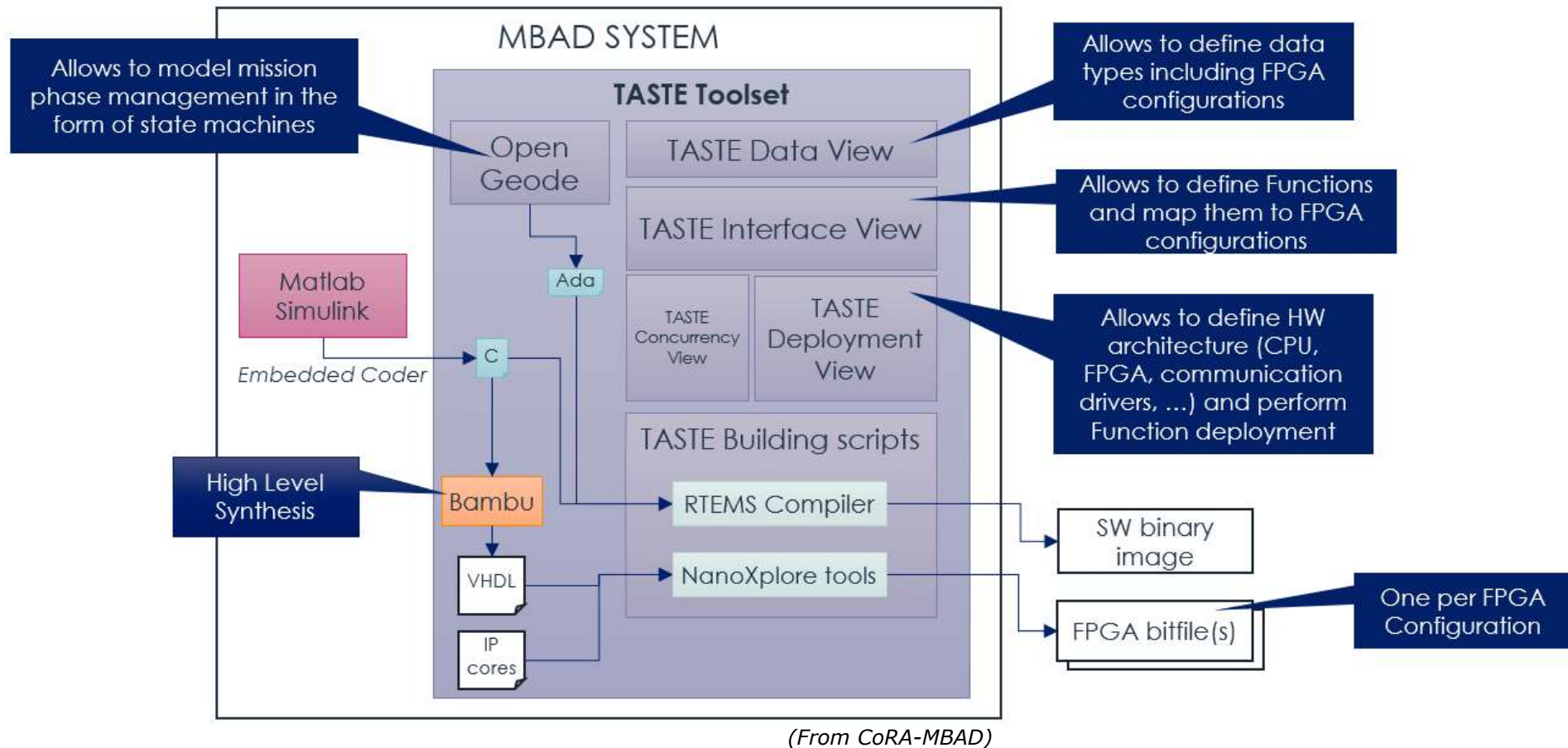


CORA-MBAD PROJECTS CAPABILITIES



Modelling recommendations and constraints	
Extension of capabilities of TASTE Toolset (importantly the DMT module)	
Integration of Bambu HLS Tool	
CoRA RHDC Middleware library (e.g. RMAP/SpW)	Advanced eXtensible Interface (AXI) communication interface for on-chip communication.
RTEMS-5 Cross compiler from Cobham Gaisler	RTEMS 5.1 custom built cross compiler for ARM Cortex A9
NanoXplore tools	Xilinx FPGA development toolchain (Vivado)

CORA-MBAD PROJECTS ARCHITECTURE



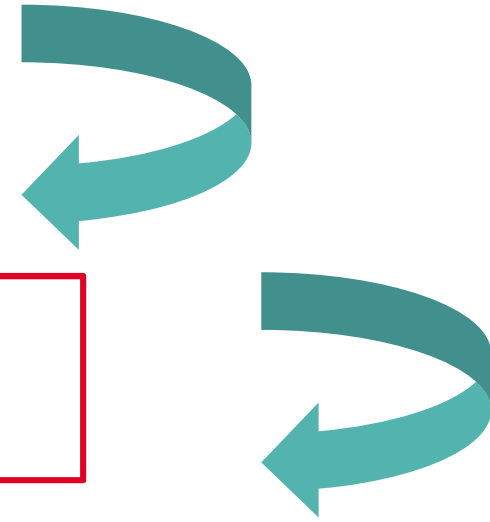
CORA-MBAD PROJECTS

SW-HW INTERACTION

Glue code **controlling** whether it is possible to call HW accelerator

Bus commands to **write** input parameters, **trigger** activation of HW accelerator and **read** output parameters

VHDL code **triggering** the activation of HW accelerator and **reporting** the outputs

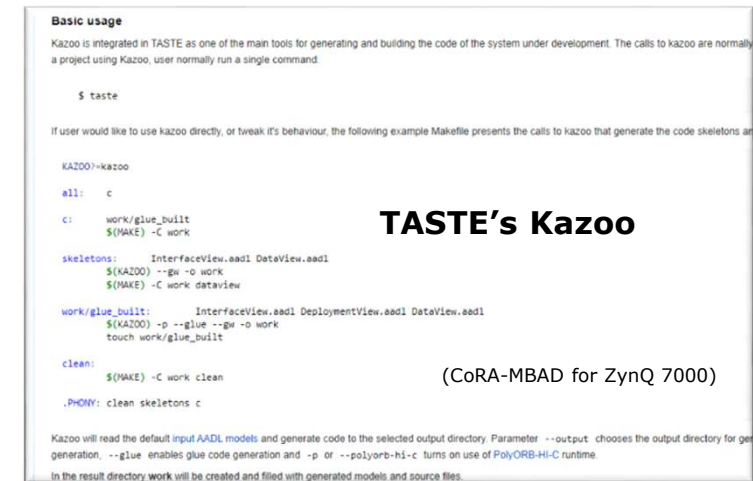


CORA-MBAD PROJECTS TOOLING

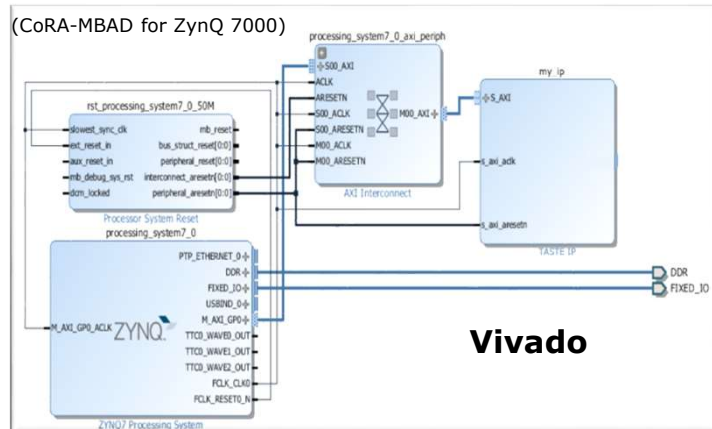
Note that both **TASTE** and **Bambu** are **open-source** SW tools, so subsystems built in pure C can be synthesized and executed on the FPGA with no external dependencies. Bambu is FPGA vendor independent, hence it can be used with minor adaptations needed for each FPGA specific component.



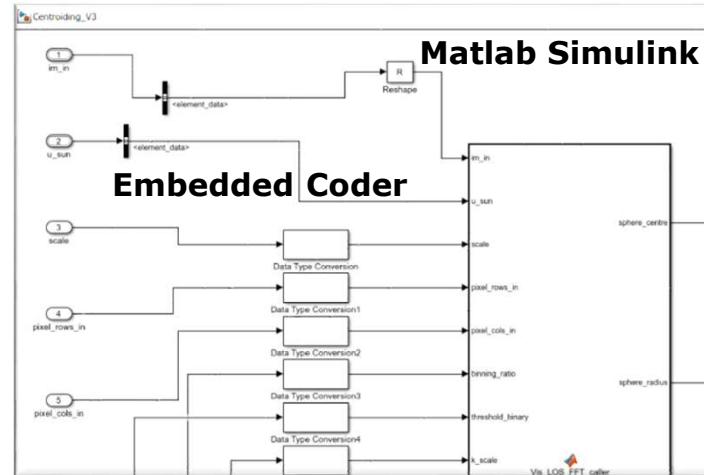
<https://taste.tools>



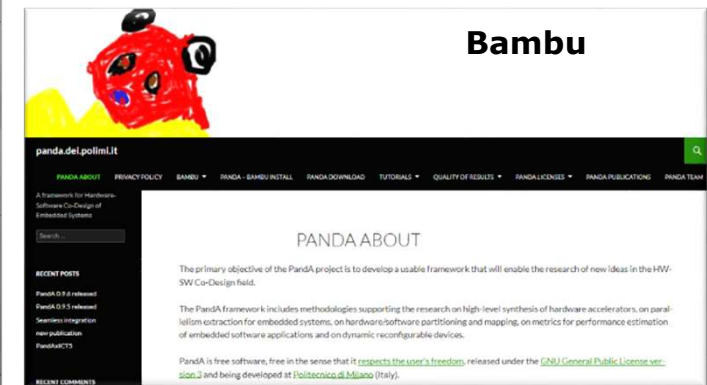
<http://taste.tuxfamily.org/wiki/index.php?title=Kazoo>



<https://www.xilinx.com/products/design-tools/vivado.html>



<https://www.mathworks.com/products/embedded-coder.html>



<https://panda.dei.polimi.it/>

CORA-MBAD PROJECTS

TOOLING - Bambu



<https://panda.dei.polimi.it/>

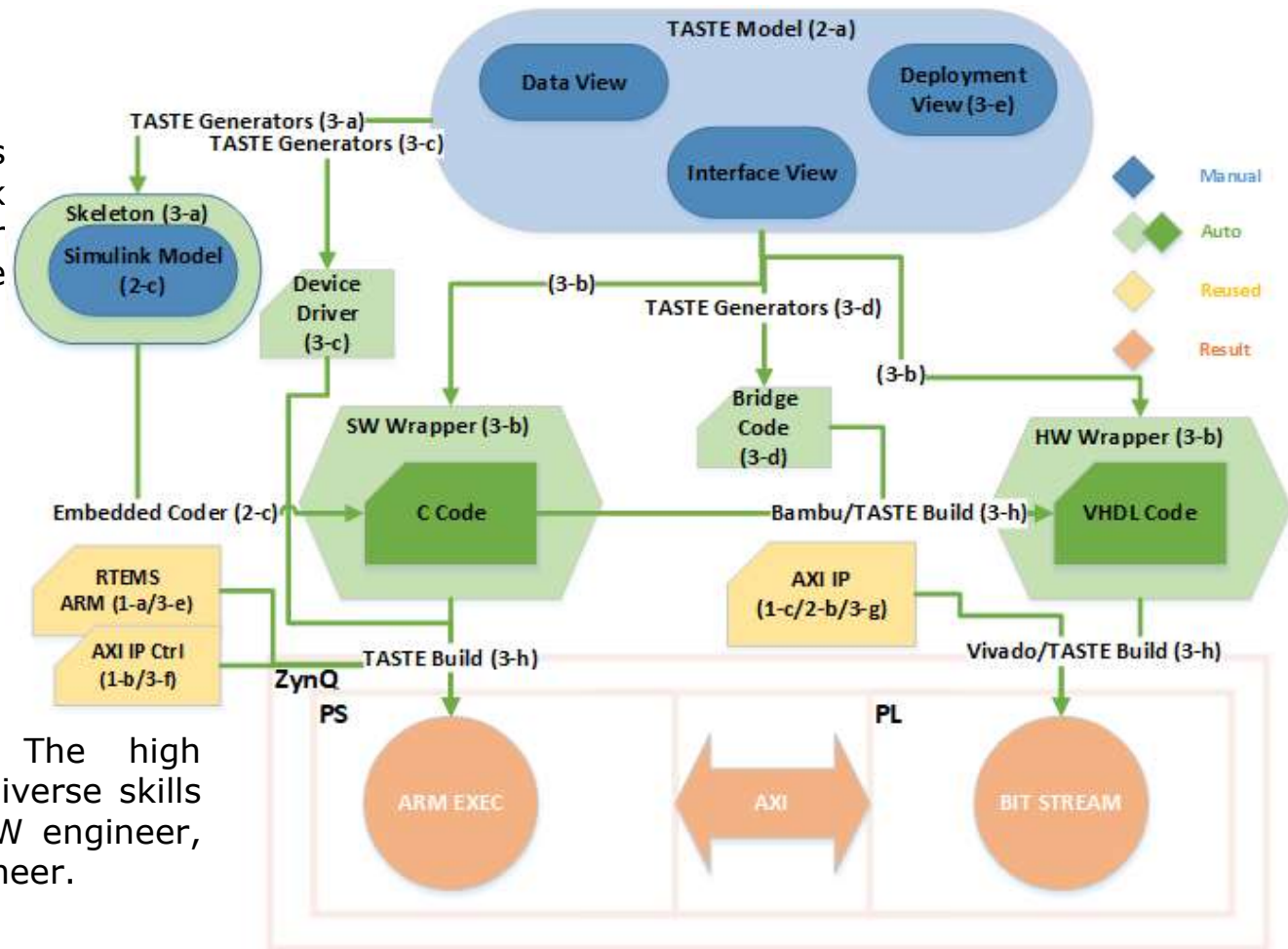
- Modern **HLS** tool
- Developed at **Politecnico di Milano (Italy)**
- Within the **PandA** framework
- **GPL v3**
- Now part of **TASTE!**
- Interfacing with **GCC/CLANG-LLVM**
- Complete support for **ANSI C** (except for recursion)
- Source code **optimizations**
- **Target-aware** synthesis
- Integrated **testbench** generation and simulation
- Back-end: automated interaction with commercial **synthesis tools**
 - FPGA: Xilinx ISE, Xilinx Vivado, Altera Quartus, Lattice Diamond, NanoXplore
 - ASIC: Synopsis Design Compiler

CORA-MBAD PROJECTS ORCHESTRATION

A **pivot open toolchain** gluing all elements together: **TASTE**, being an open framework targeting heterogeneous systems, is particular suitable to **integrate and orchestrate** all the other necessary elements.



Multifaceted team in co-engineering: The high technical degree of the activity required the diverse skills and close collaboration of a: SW engineer, HW engineer, design environment engineer, and domain engineer.



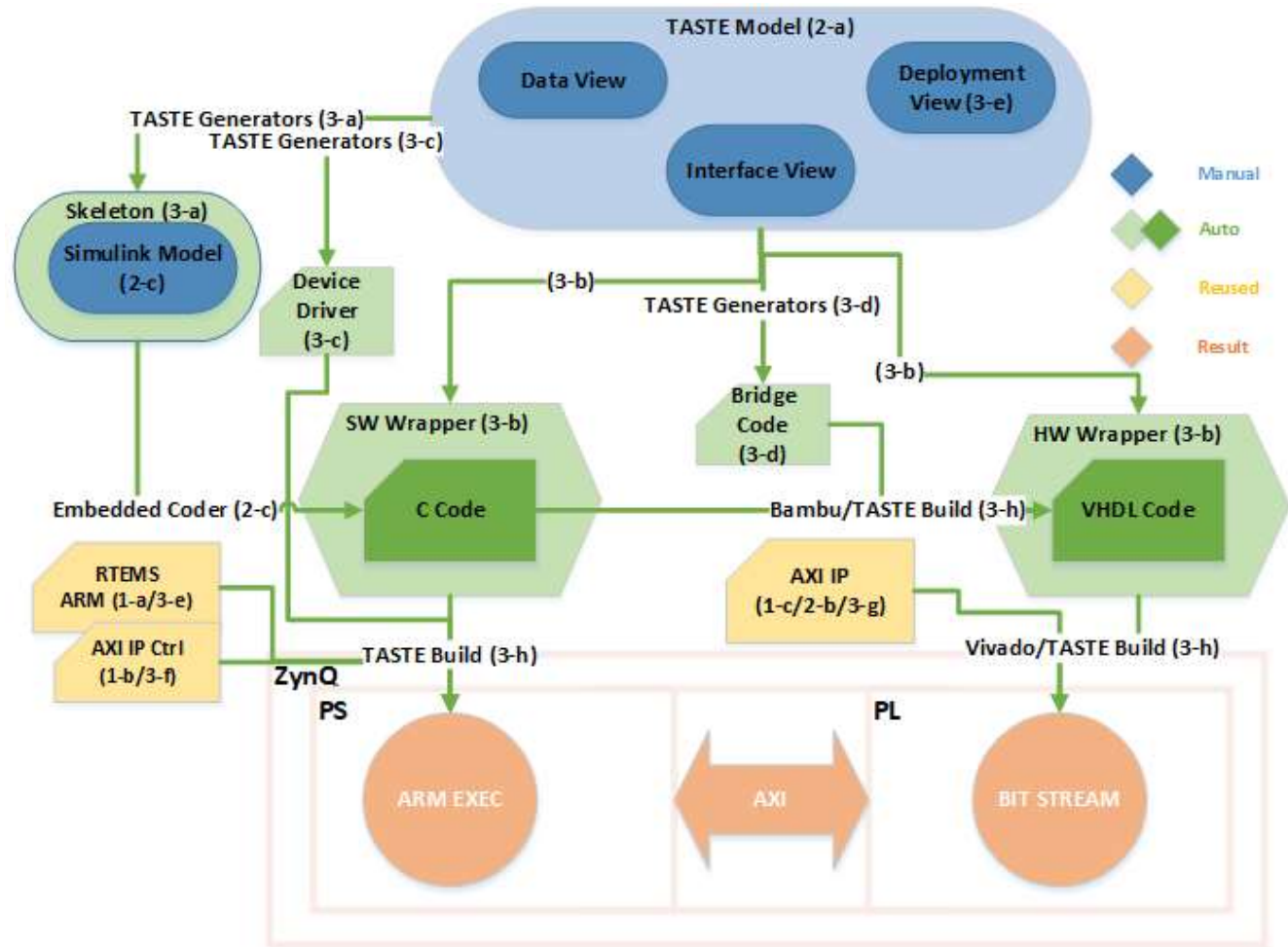
(CoRA-MBAD for ZynQ 7000)

CORA-MBAD PROJECTS ORCHESTRATION

HW/SW dispatcher

FPGA reconfiguration engine

Header file generation needed for reconfiguration,
containing information about generated
bitfile sizes and memory addresses



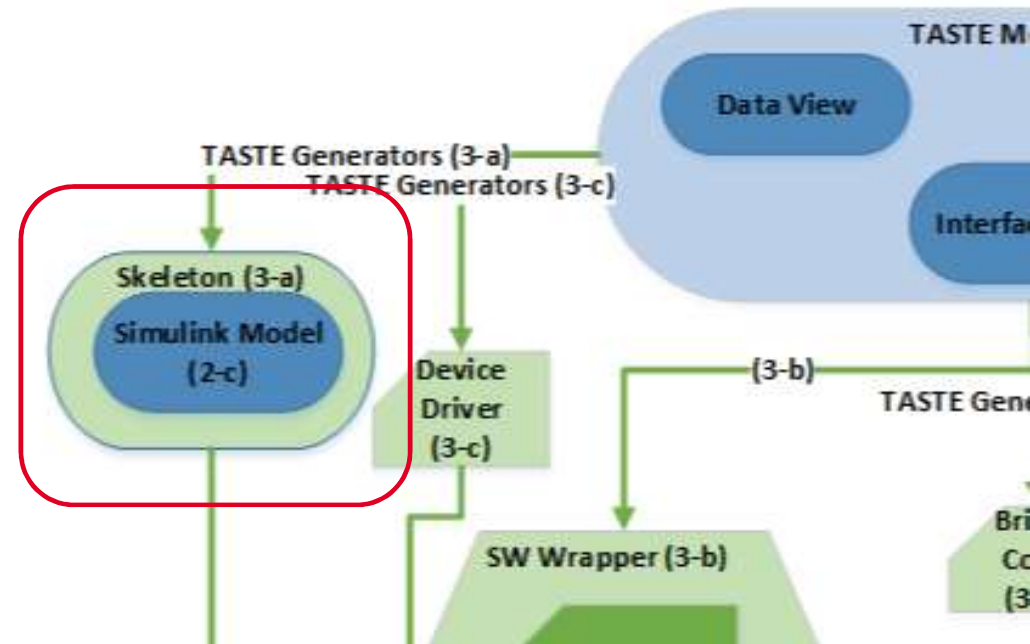
(CoRA-MBAD for ZynQ 7000)

CORA-MBAD PROJECTS

WHAT IS GENERATED

From a common ASN.1 data model and an AADL minimalistic component interface model it consistently and automatically exports:

- **Interface definition** in the target language of choice with consistent inputs and outputs (in our demonstrator a Simulink model)

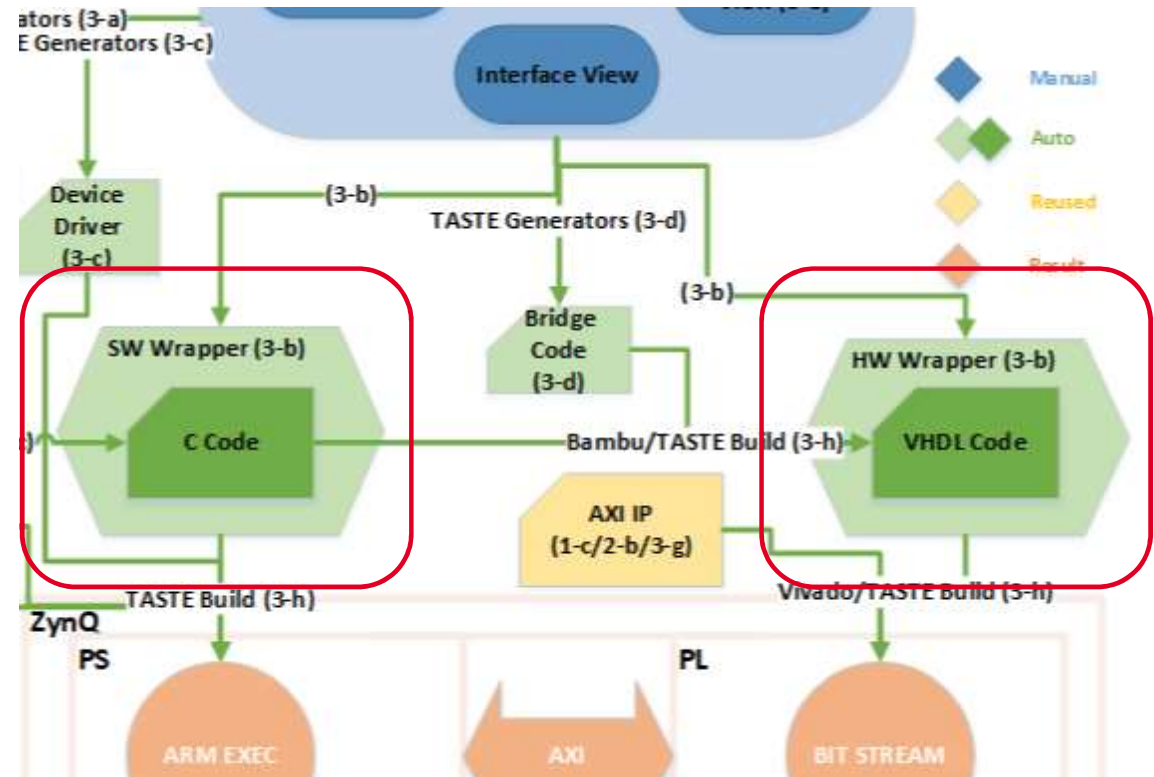


CORA-MBAD PROJECTS

WHAT IS GENERATED

From a common ASN.1 data model and an AADL minimalistic component interface model it consistently and automatically exports:

- **SW and HW wrapper** interface code that transparently guarantees the correct communication between the target's functions. Importantly these interface wrappers automatically grow or shrink according to the number and type of inputs and outputs.



(CoRA-MBAD for ZynQ 7000)

CORA-MBAD PROJECTS

WHAT IS GENERATED (ZYNQ)

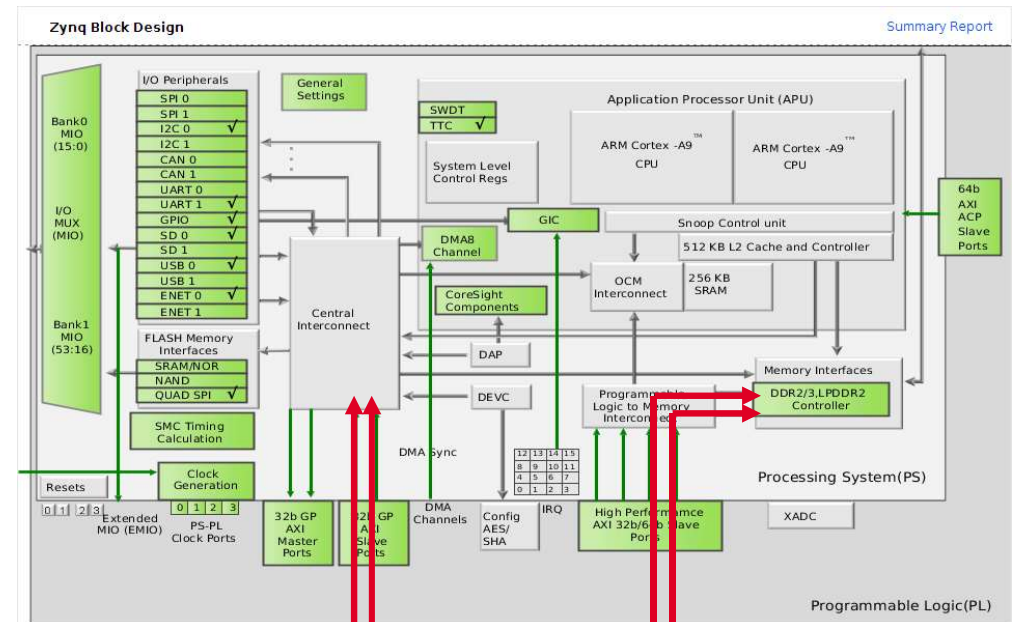
CoRA-ZynQ makes use of AXI bridges of Zynq-7000 architecture to connect PS with PL.

Three independent interfaces are implemented in order to provide different capabilities:

- one AXI interface used to write and read configuration registers
- one AXI interface fully devoted to write and read large blocks of memory inside FPGA
- one AXI stream interface to support stream data processing

The number of registers or memories addressed through AXI interfaces can be configured to optimize the resource allocation of the FPGA.

In addition, AXI stream transmission can be directed to achieve up to 32 different destinations through the same interface.



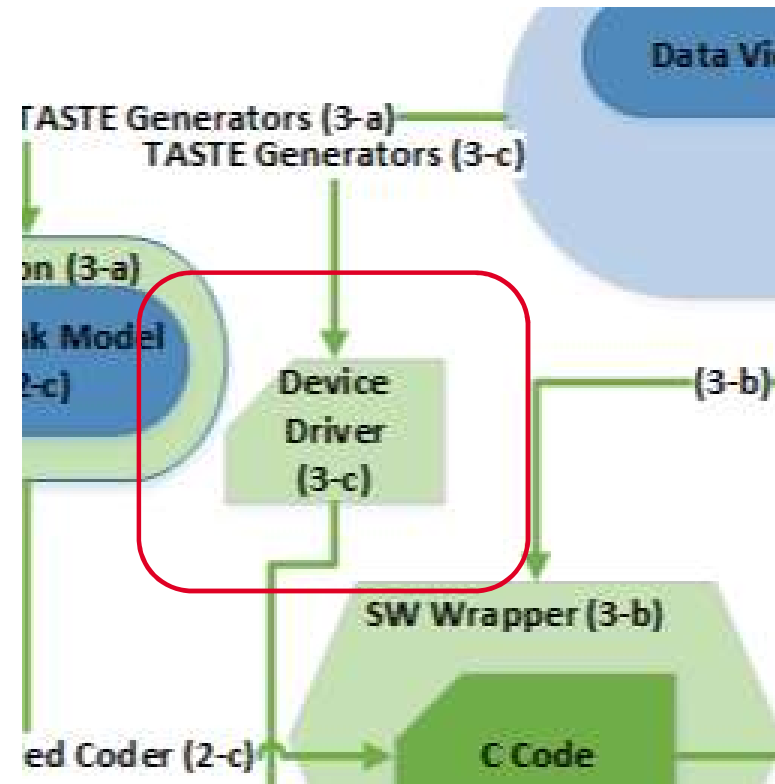
TASTE WRAPPER

CORA-MBAD PROJECTS

WHAT IS GENERATED

From a common ASN.1 data model and an AADL minimalistic component interface model it consistently and automatically exports:

- **SW device driver** to provide SW-HW communication with the HW implementation of the target function.

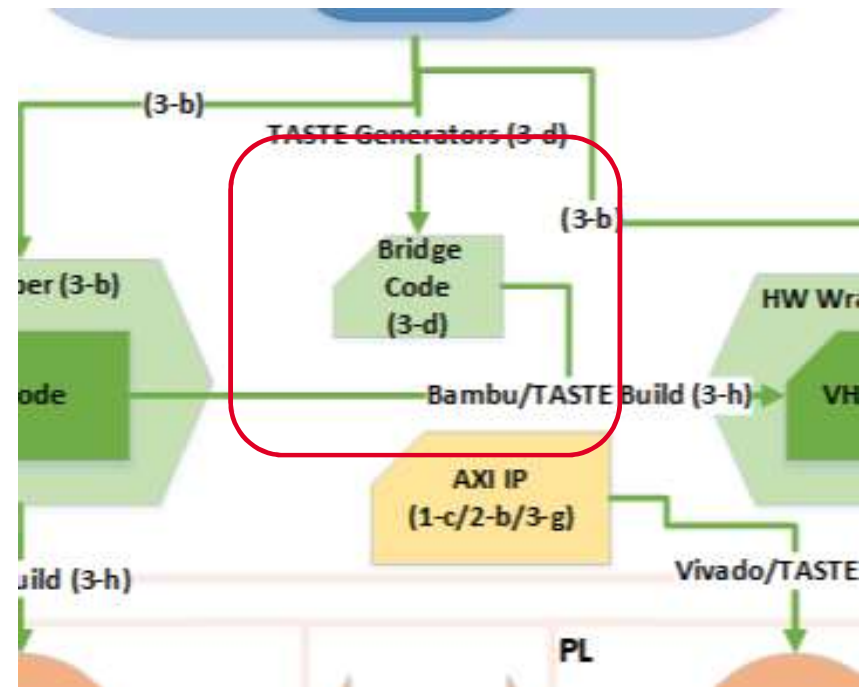


CORA-MBAD PROJECTS

WHAT IS GENERATED

From a common ASN.1 data model and an AADL minimalistic component interface model it consistently and automatically exports:

- **"Bridge" code** with the necessary adaptations and extra inputs needed in the transition between two autocoding tools, in this case between Embedded Coder and Bambu.



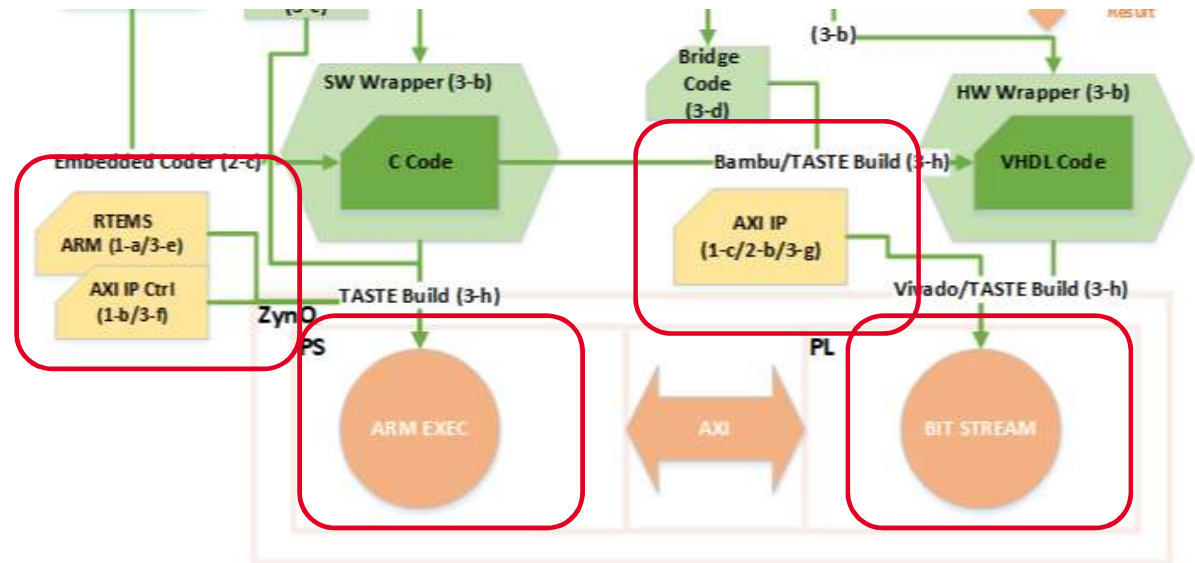
(CoRA-MBAD for ZynQ 7000)

CORA-MBAD PROJECTS

WHAT IS GENERATED

Additionally, TASTE:

- integrates the custom **cross-compiler** as part of a new deployment target
- links with the **BSP** to use the necessary **HW drivers**
- orchestrates the calls to all needed **autocode and compilation tooling** - e.g. forwarding the Embedded Coder output as a Bambu input together with the generated consistent bridge
- calls the **synthesis, placement and routing** facilities and generates FPGA bitfiles



(CoRA-MBAD for ZynQ 7000)

CORA-MBAD PROJECTS

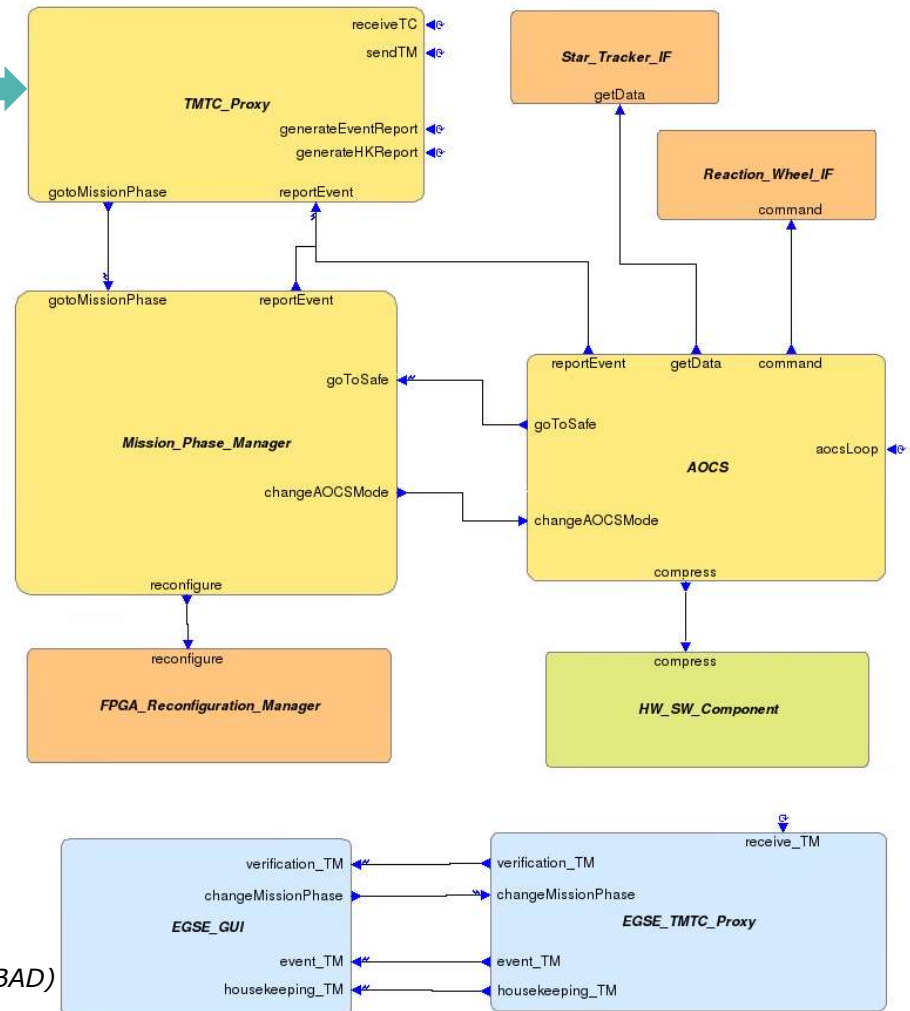
USE CASES (GR740/NG-MEDIUM)

Use Cases



1. FPGA Reconfiguration
 - Test integration of FPGA reconfiguration engine from CoRA RDHC
 - FPGA Reconfiguration Manager
2. External Interfaces
 - Test integration of CoRA RDHC Middleware library
 - TMTC, GPIO, SpW RMAP, SpW CPTP
3. STR Based Safe Mode
 - Simulink model adapted from ESA R&D study « STR Based Safe mode »
 - Fixed-point arithmetic complexifies AOCS/GNC design and implementation
 - **Memory Blocks: 78/56 (136%)**
4. Image Compression
 - Discrete Wavelet Transform part of CCSDS 122.0-B-2
 - Synthesis possible with small images

Use Case Architecture

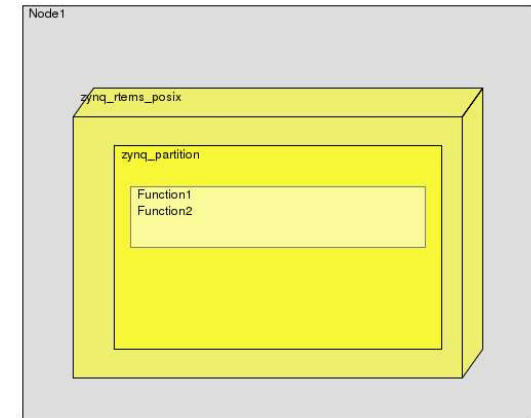
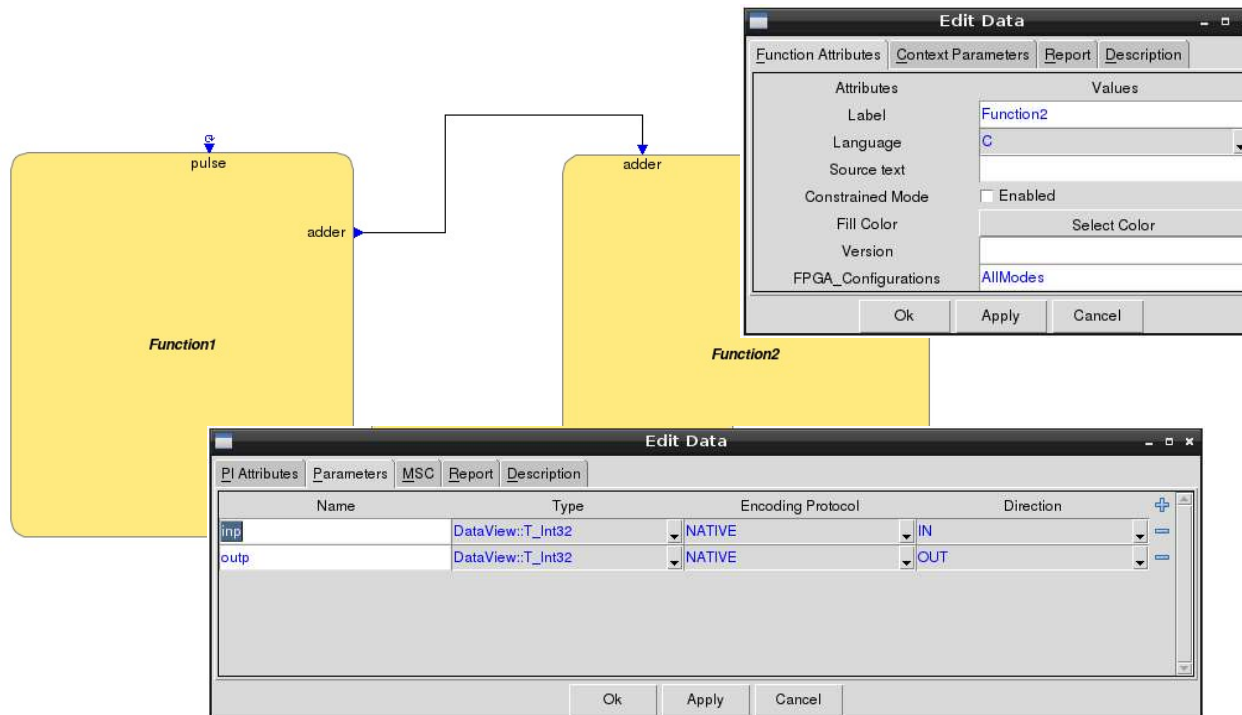


(From CoRA-MBAD)

CORA-MBAD PROJECTS USE CASES (ZYNQ)

Simple Use Case – Prime numbers

- Compute prime numbers from C code, on Zynq-7000:
 - Caller function in C, running on ARM.
 - Compute function in C, running on FPGA.



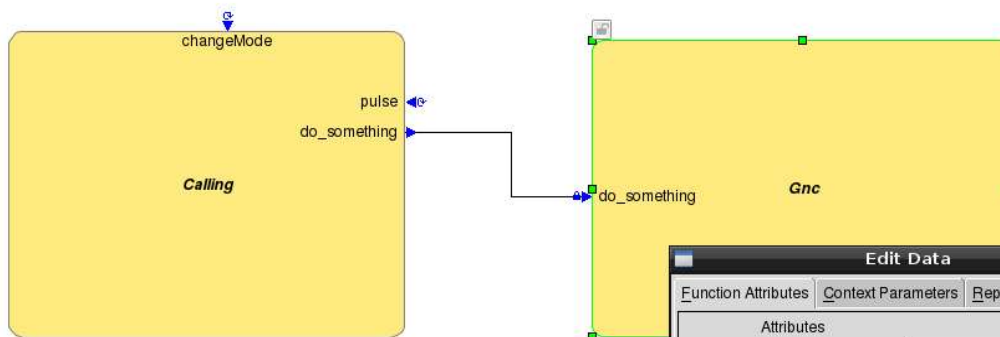
```
void function2_PI_adder
(const asnlScdT_Int32 *IN_inp,
 asnlScdT_Int32 *OUT_outp)
{
    // Write your code here
    if( *IN_inp % 2 == 0){
        *OUT_outp = 2;
        return;
    } else {
        for (int i=3; i <= *IN_inp / 2; i+=2){
            if(0 == (*IN_inp%i)){
                *OUT_outp = i;
                return;
            }
        }
    }
    *OUT_outp = *IN_inp;
}
```

```
[function1_PI_pulse] Sent 8, Got 2
[function1_PI_pulse] Sent 9, Got 3
[function1_PI_pulse] Sent 10, Got 2
[function1_PI_pulse] Sent 11, Got 11
[function1_PI_pulse] Sent 12, Got 2
[function1_PI_pulse] Sent 13, Got 13
[function1_PI_pulse] Sent 14, Got 2
[function1_PI_pulse] Sent 15, Got 3
[function1_PI_pulse] Sent 16, Got 2
[function1_PI_pulse] Sent 17, Got 17
[function1_PI_pulse] Sent 18, Got 2
[function1_PI_pulse] Sent 19, Got 19
```


CORA-MBAD PROJECTS USE CASES (ZYNQ)

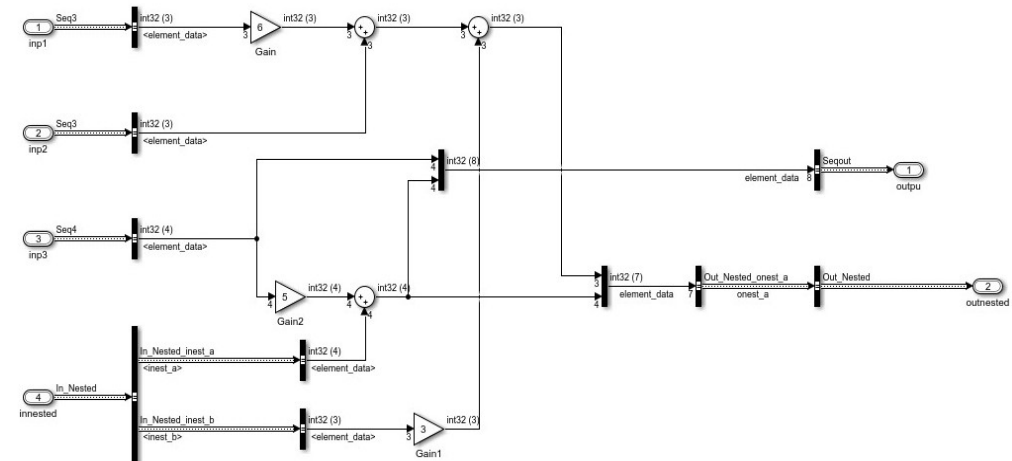
Simple Use Case – GNC

- Compute simple GNC algorithm from Simulink code, on Zynq-7000:
 - Caller function in C, running on ARM.
 - Compute function in Simulink, running on FPGA.



Name	Type	Encoding Protocol	Direction
inp1	DataView::Seq3	NATIVE	IN
inp2	DataView::Seq3	NATIVE	IN
inp3	DataView::Seq4	NATIVE	IN
outpu	DataView::Seqout	NATIVE	OUT
innested	DataView::In_Nested	NATIVE	IN
outnested	DataView::Out_Nested	NATIVE	OUT

Edit Data	
Function Attributes	Context Parameters
Label	Gnc
Language	Simulink
Source text	
Constrained Mode	<input type="checkbox"/> Enabled
Fill Color	Select Color
Version	
is_Component_Type	
is_instance_of	
FPGA_Configurations	modeA,modeB



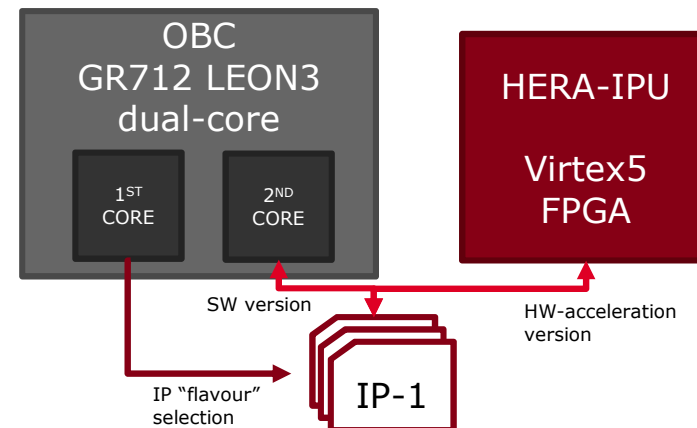
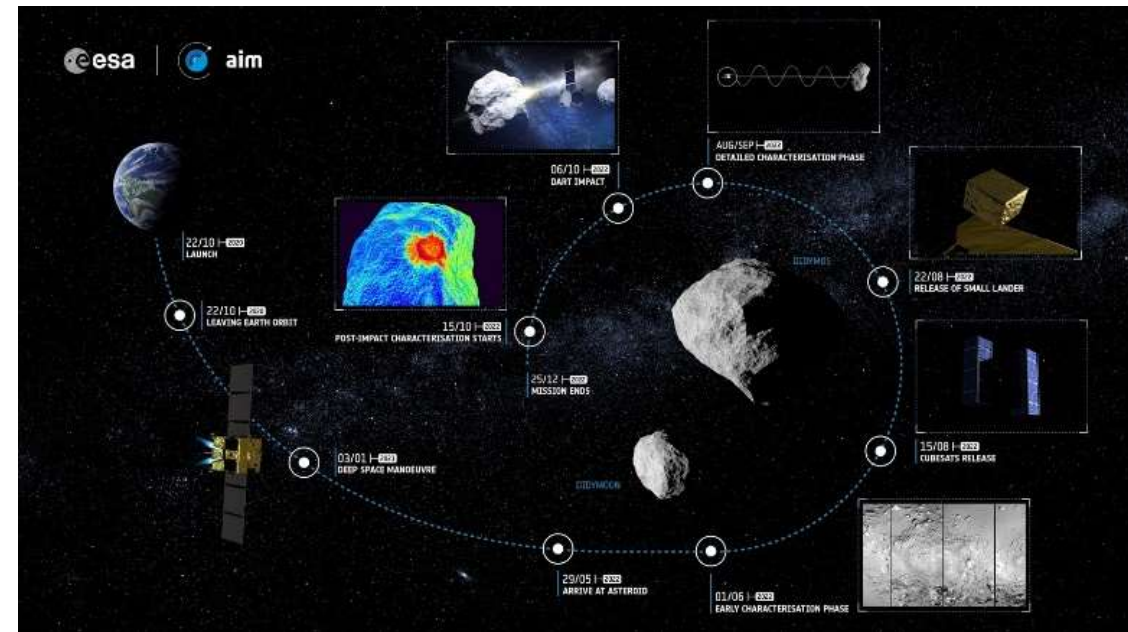
```

Data View | Interface View | Deployment View | Concurrency View | AADL
1 TASTE-DataView DEFINITIONS ::=
2 BEGIN
3 IMPORTS T-Int32, T-UInt32, T-Int8, T-UInt8, T-Boolean FROM TASTE-BasicTypes;
4
5 TASTE-Boolean ::= BOOLEAN
6
7 -- ASN.1 Type definitions must start with an uppercase
8 -- Field names in records (SEQUENCE) must start with a lowercase
9 -- Underscores are not allowed: use dash (hyphen)
10
11 MyInteger ::= INTEGER (-411673351.. 763937697)
12 -- MyReal ::= REAL (0.0 .. 10.0)
13
14 Seq3 ::= SEQUENCE (SIZE (3)) OF MyInteger
15 Seq4 ::= SEQUENCE (SIZE (4)) OF MyInteger
16 Seqout ::= SEQUENCE (SIZE (8)) OF MyInteger
17
18 In-Nested ::= SEQUENCE {
19   inest-a SEQUENCE (SIZE (4)) OF T-Int32,
20   inest-b SEQUENCE (SIZE (3)) OF T-Int32
21 }
22
23 Out-Nested ::= SEQUENCE {
24   onest-a SEQUENCE (SIZE (7)) OF T-Int32
25 }
26
27 -- Simple types:
28
    
```

CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

- Autonomous Vision-Based Navigation
- Proximity operations
- Computer-vision algorithms are computationally costive in terms of execution time and memory
 - Offload SW Processor
 - Acceleration by means of FPGA
 - Large data to be processed 1024*1024 pixels
 - Real-Time high-performances
- HW-accelerated functions
 - Computer-vision Lambertian sphere matching of asteroid body
 - Feature Detection and Tracking surface terrain of asteroid
 - Interchange SW functions with FPGA ones and among them

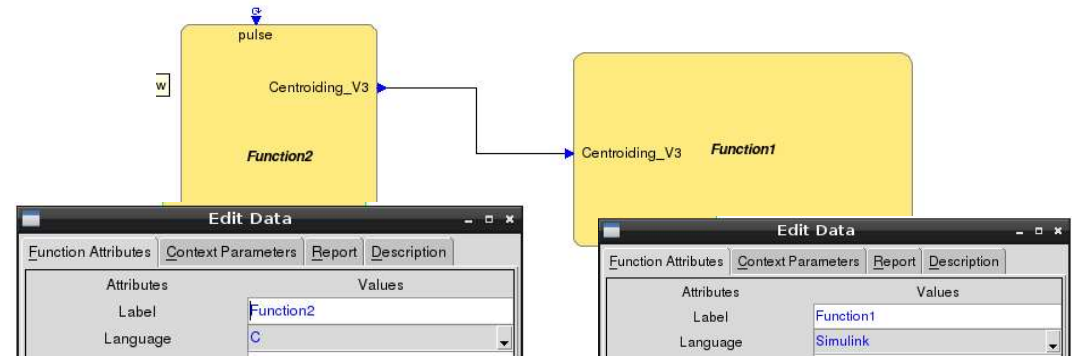
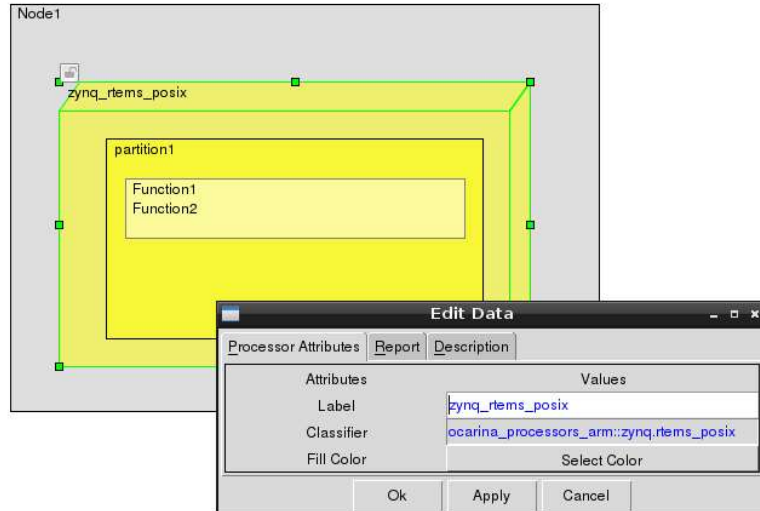


CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

TASTE model designed for Use Case algorithm integration:

- ASN.1 Data types tailored to use case IOs.
- Interface View with a calling interface to the Simulink generated algorithm.
- Deployment on Zynq-7000 target.



Attribute	Data View	My Real	Native	IN	OUT
pixel_rows_in	DataView::MyReal	NATIVE	IN		
pixel_cols_in	DataView::MyReal	NATIVE	IN		
binning_ratio	DataView::MyReal	NATIVE	IN		
threshold_binary	DataView::MyReal	NATIVE	IN		
k_scale	DataView::MyRealSeq5	NATIVE	IN		
k_scale_step	DataView::MyReal	NATIVE	IN		
auto_k_scale	DataView::MyReal	NATIVE	IN		
radius_peak	DataView::MyReal	NATIVE	IN		
margin_radius_rel	DataView::MyReal	NATIVE	IN		
COB_diff_thr	DataView::MyReal	NATIVE	IN		
CAM_FOV	DataView::MyReal	NATIVE	IN		
asteroid_real_diameter	DataView::MyReal	NATIVE	IN		
sphere_centre	DataView::MyRealSeq2	NATIVE	OUT		
sphere_radius	DataView::MyReal	NATIVE	OUT		
best_valid_idx	DataView::T_BestIdx	NATIVE	OUT		

CORA-MBAD PROJECTS

USE CASES (ZYNQ)

HERA mission

The Matlab design reused is not tailored for a HW implementation (e.g. no parallel nor fixed-point design) which naturally represents some challenges to the autocoding facilities and HW resource usage.

Such tailoring was not yet performed due to project scope and time availability.

SW RUN



```
[[Function2] Startup...  
[function1] Centroiding_V3_initialize  
[Function2] PI_pulse  
[function1] Centroiding_V3_step  
[Function2] PI_pulse done!  
[Function2] Elapsed Time - 2.005620539 s  
Centroiding_V3 done with iteration 0  
    sphere_centre      {520.000000, 500.000000}  
    sphere_radius      201.791794  
    cent_valid_flg     TRUE  
    apparent_altitude  22115.820312
```

```
Expected sphere_radius 201.791794 (0x40693956605ee569), got 201.791794 (0x4069395660000000)  
Expected apparent_altitude 22115.820313 (0x40d598f4800218df), got 22115.820312 (0x40d598f480000000)
```

CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

Instead, leverage to the maximum possible extent the **configurability and autocoding strengths of the toolchain**, avoiding any manual work, e.g. by exploring:

- the rich Embedded Coder and Bambu **options**
- types of possible SW-HW **interfaces** generated (e.g. external memory access, streaming type parameters) and resulting HW resource allocation.

some Bambu options relevant:

```
bambu --compiler=I386_CLANG4 --experimental-  
setup=BAMBU-TASTE --no-iob --clock-period=10 -O2 ...
```

some Embedded Coder options relevant:

long long support
reference interfaces

Some metrics of the Bambu generated HW kernel:

```
<item stringID="XILINX_SLICE" value="30202"/>  
<item stringID="XILINX_SLICE_REGISTERS" value="53786"/>  
<item stringID="XILINX_SLICE_LUTS" value="98319"/>  
<item stringID="XILINX_BLOCK_RAMFIFO" value="83"/>  
<item stringID="XILINX_IOPIN" value="0"/>  
<item stringID="XILINX_DSPS" value="92"/>  
<item stringID="XILINX_POWER" value="0.416"/>  
<item stringID="XILINX_DESIGN_DELAY" value="24.028"/>
```


CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

Bambu options are enriched for the specific HERA mission use case and its needs on Panda-Bambu v0.9.7:

- User allocation of external memory
- Floating point computation considerations
- Specific TASTE setup

Bambu options (example):

bambu

```
--compiler=I386_CLANG4
--experimental-setup=BAMBU-TASTE
--funroll-loops
--no-iob
--clock-period=5
--fno-inline -O2
--speculative-sdc-scheduling -DSTATIC=
--top-fname=bambu_Centroiding_V3
--generate-interface=INFER -v4
--panda-parameter=none-registered-ptrdefault=1
-lm --xml-memory-allocation=memory_allocation.xml
--evaluation --device-name=xc7z045-2ffg900-VVD
--no-clean --max-sim-cycles=1000000000
--max-ulp=1073741825
--bram-high-latency=4
--memory-ctrl-type=D21 bambu_Centroiding_V3_array.c
Centroiding_V3.c
ert_main.c rtGetInf.c rtGetNaN.c rt_nonfinite.c
```

CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

Hardware limitations

Some considerations should be taken into account when complex systems like HERA, composed by several intensive algorithms in terms of memory access and allocation, are implemented on FPGAs:

- Design should implement as less accesses to memory as possible in order to increment the performance of the application.
- Accesses to memory should have a low latency (ideally embedded memory blocks into FPGA).

However, in some cases it is impossible to allocate all the needed memory into RAM blocks and external memories are required.



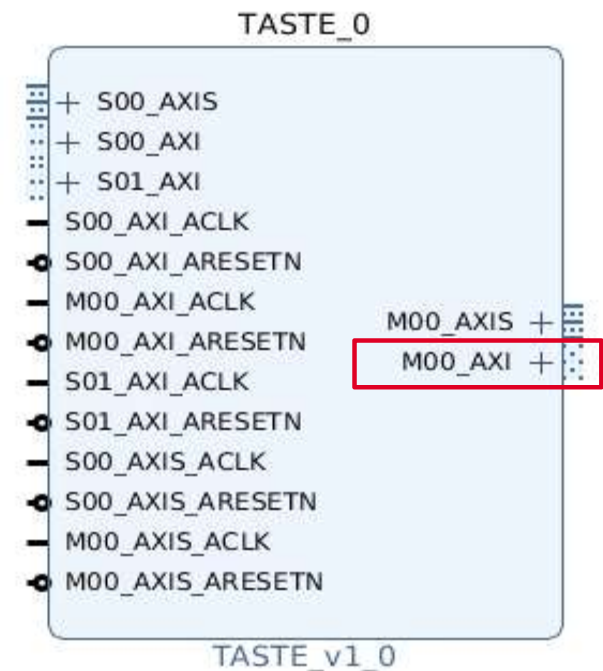
CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

Advanced TASTE Wrapper

In order to solve the memory limitations in HERA design (or any other design intensive in memory) a new AXI interface has been added to connect with any external memory controller with an AXI user interface.

This new interface allows the generated Bambu design to read and write from/to an external memory. In the specific case of HERA design, the SODIMM DDR3 allocated in ZC706 board and accessible by the PL.



CORA-MBAD PROJECTS USE CASES (ZYNQ)

HERA mission

Advance TASTE Wrapper Status

- Advance TASTE Wrapper IP generated and interfaces validated in simulation
- HERA block design implemented

To be performed

- Validation of HERA design in simulation
- Validation of HERA design in ZC706



CORA-MBAD PROJECTS

CONCLUSIONS

- TASTE's Kazoo facilitates toolchain evolution
- Valuable feedback to NanoXplore and Bambu
- Co-engineering process requires continuous communication / collaboration between the different teams (e.g. increased collaboration between SW and HW engineers)
- Co-working during development and validation phases is very beneficial (3 workshops CoRA-MBAD showed great benefits, was fun and motivating)
- Role of Design Environment Engineer becoming increasingly important
- Generated SW-HW interface and HW interface with Bambu kernel for computation is otherwise very challenging - CoRA-MBAD allows for rapid implementation iterations
- FPGA resources can be bottleneck (to blame the tool also)
- The configuration of the autocoding tools goes a long way in making integration easier and resulting in more efficient designs
- TASTE's toolchain allows for easy scalability of HW interfaces, fitting applications needs

CORA-MBAD PROJECTS

FUTURE WORK

- Improve autocoding and synthesis results (improving usage of HW resources)
- Define best way to support floating point arithmetic on FPGA
- Increase HW capacity (e.g. BRAVE LARGE/ULTRA, SpaceFibre)
- Leverage highly integrated SoC platforms

- Stress test TASTE translators for robustness
- Integrate new tool versions (more capable and tailored)

- Address more use cases, targetting diversified purposes (e.g. speed, memory, interfaces, etc.)
- Support partial reconfiguration
- Define sophisticated reconfiguration manager

- Support advanced interfaces for modern EGSE integration

CoRA projects: Model-based HW/SW co-design for reconfigurable platforms

ADCSS2020

October 2020

Tiago da Silva Jorge (GMV)
tiago.jorge@gmv.com

ESA UNCLASSIFIED - For Official Use



European Space Agency