2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Evaluating the Feasibility of a RISC-V core for real time applications using a Virtual Prototype

J. Santana, G. Pachiana, T. Markwirth, C. Sohrmann (FhG IIS/EAS),
B. Fischer, M. Matschnig (Siemens)

VALU3S

SIEMENS

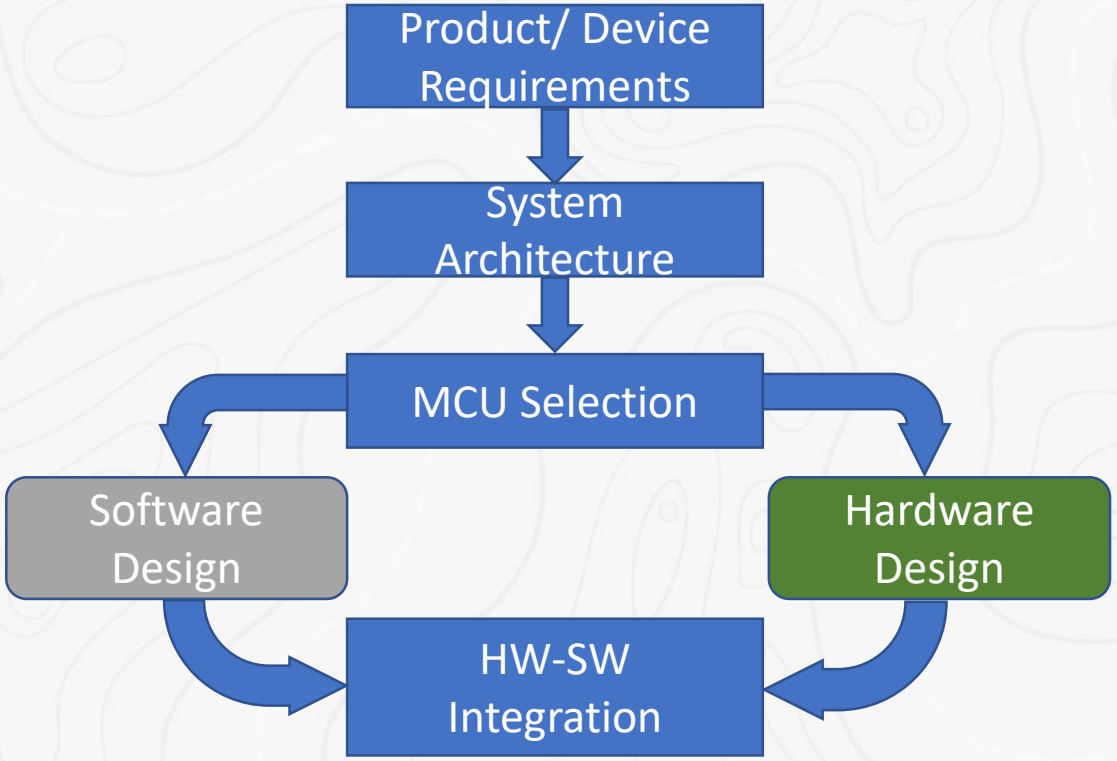
Fraunhofer IIS

accellera SYSTEMS INITIATIVE

Problem statement

- Prototyping of an embedded system

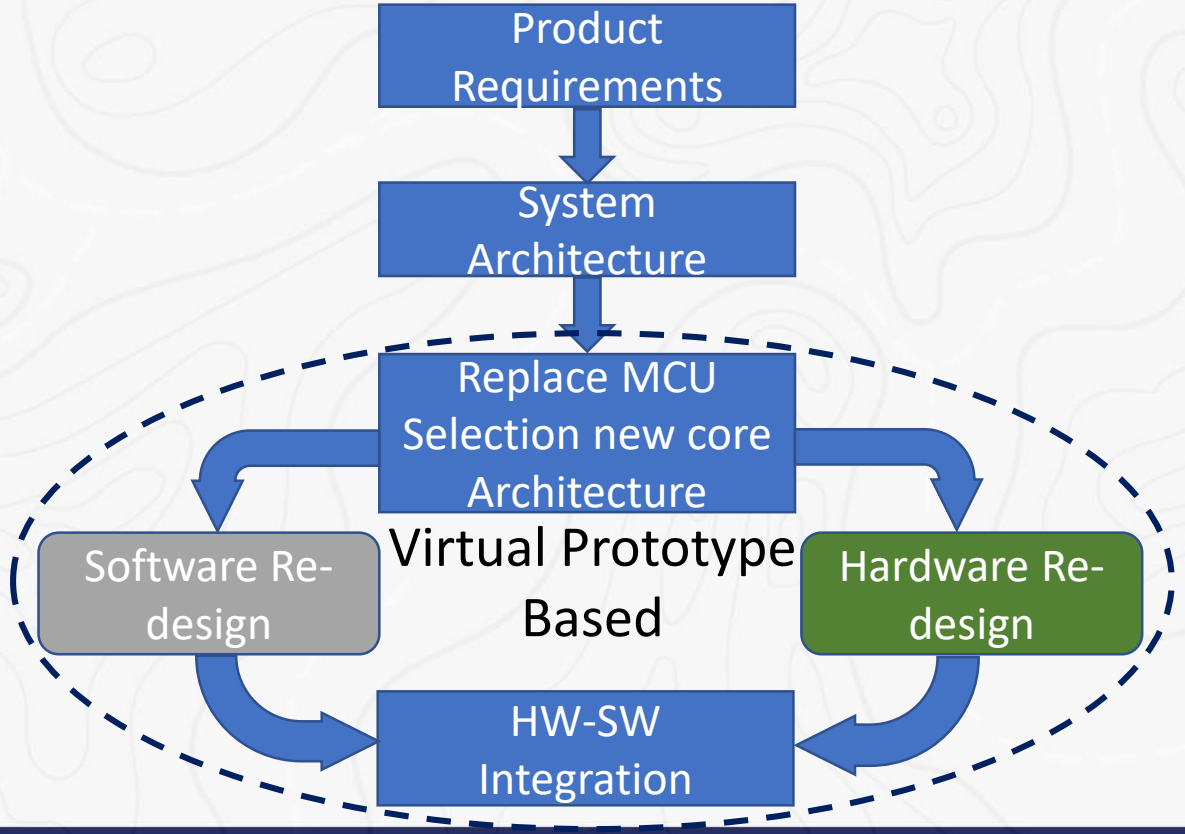
  How to V&V the system with relatively low effort (Time & Costs)?



Problem statement

- Prototyping of an embedded system with a new MCU core

  How to V&V the system with relatively low effort (Time & Costs)?



RISC-V HW/SW Ecosystem Adoption

- Reduced Instructions Set Computer-Five
  - SW Toolchain
  - Simulation and prototyping tools
  - HW implementations (Cores / SoCs)
  - Verification Frameworks

RISC-V Adoption

```
00010244 <main>:
   10244:   fe010113
   10248:   00112e23
   1024c:   00812c23
   10250:   02010413
   10254:   000f47b7
   10258:   24078793
   1025c:   fef42623
   10260:   f91ff0ef
```

i_riscv_hw_vp_top1

RISCV_HW_VP_TOP

LED

```
test_simple_blink $make run
BLINK_LED
LED_ON
LED_OFF
LED_ON
LED_OFF
LED_ON
LED_OFF
```

C language/human coding

Assembler/ machine language

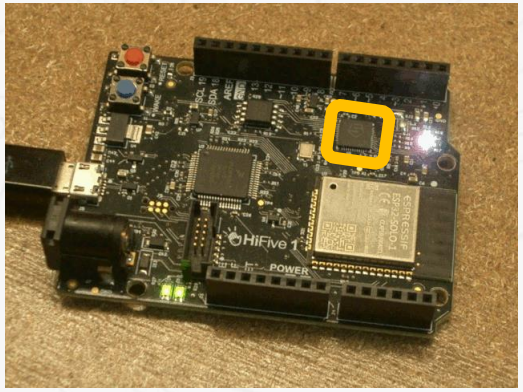
Instructions /Program memory

```
int main()
{
    unsigned int blink_status; //blink status
    unsigned int delay_tmp = 1000000; //Delay of

    /* Init minimize MUC HW to make it work in n
    InitMCU();

    /* Init SPI to basic normal working mode */
    InitSPI();

    for(;;)
    {
        /*  Blink ON LED */
        blink_status = blink_led(0, LED_ON);
        /* delay 1s */
        DelayMicroSecond(delay_tmp);
        /*  Blink OFF LED */
        blink_status = blink_led(0, LED_OFF);
        /* delay 1s */
        DelayMicroSecond(delay_tmp);
    }

    return 0;
}
```

```
addi    sp,sp,-32
sw  ra,28(sp)
sw  s0,24(sp)
addi    s0,sp,32
lui a5,0xf4
addi    a5,a5,576 # f4240 <__glo
sw  a5,-20(s0)
jal ra,101f0 <InitMCU>
jal ra,10228 <InitSPI>
li  a1,0
li  a0,0
jal ra,10348 <blink_led>
sw  a0,-24(s0)
lw  a0,-20(s0)
jal ra,1029c <DelayMicroSecond>
li  a1,1
li  a0,0
jal ra,10348 <blink_led>
sw  a0,-24(s0)
lw  a0,-20(s0)
jal ra,1029c <DelayMicroSecond>
j   10268 <main+0x24>
```

RISC-V

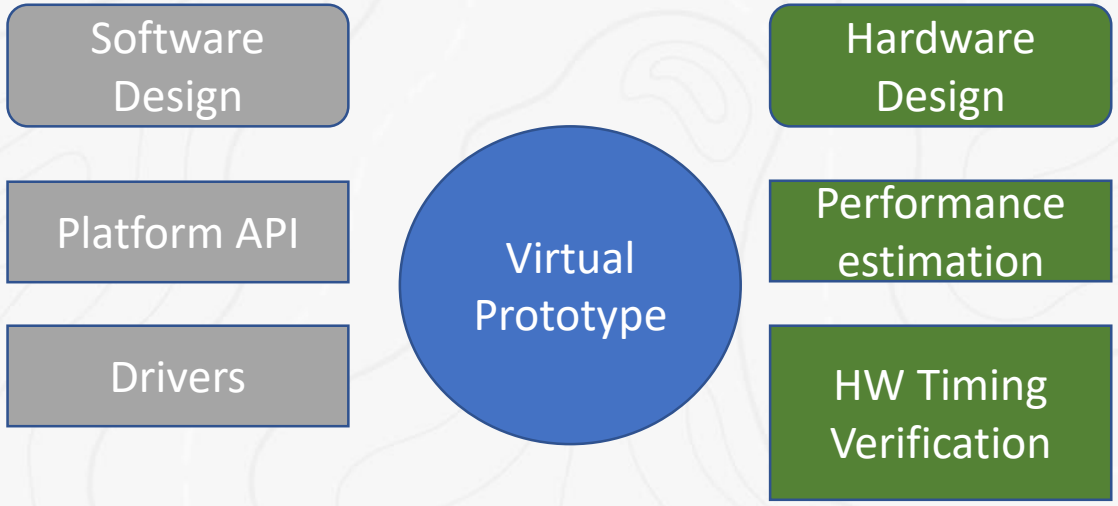
risc-v-branding-guidelines-and-materials



https://philmulholland.medium.com/modern-c-for-bare-metal-risc-v-zero-to-blink-part-2-overview-a1a6af8fede7

Method based on a Virtual Prototype

- What is a Virtual Prototype?
  - Executable software model of a hardware system.
- Virtual Prototypes are used by and for:
  - SW developers to support development phase and test cases
  - HW developers for architectural exploration and performance estimation

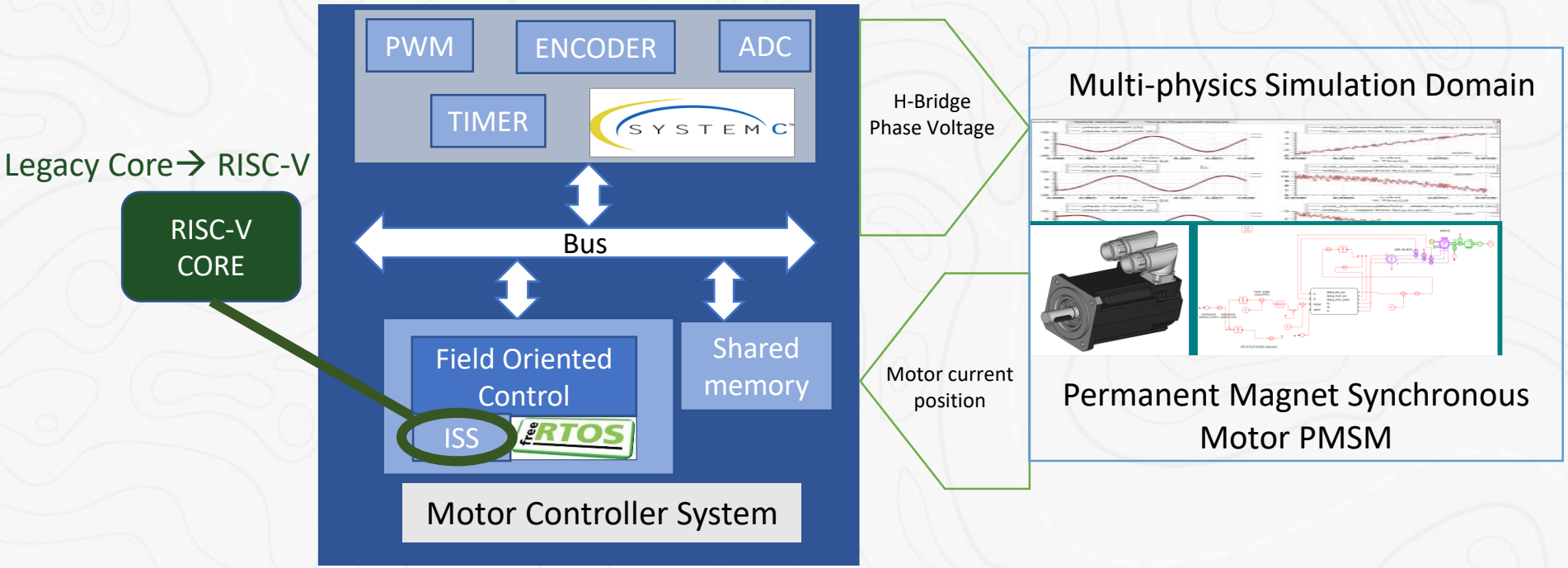


Method based on a Virtual Prototype

- Virtual Prototypes have advantages like:
  - Early available
  - Easy to setup
  - Easy to extend, e.g., by SystemC/SystemC-AMS
  - Enable fast simulation of the full system, compared with RTL
  - Reduce HW dependancy

Use Case

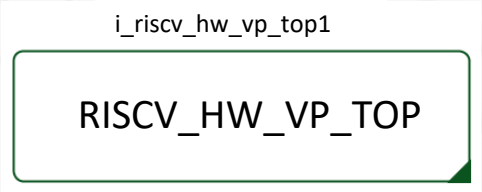
- Motor Controller Unit HW
- Field Oriented Control algorithm SW Application



Proposed methodology

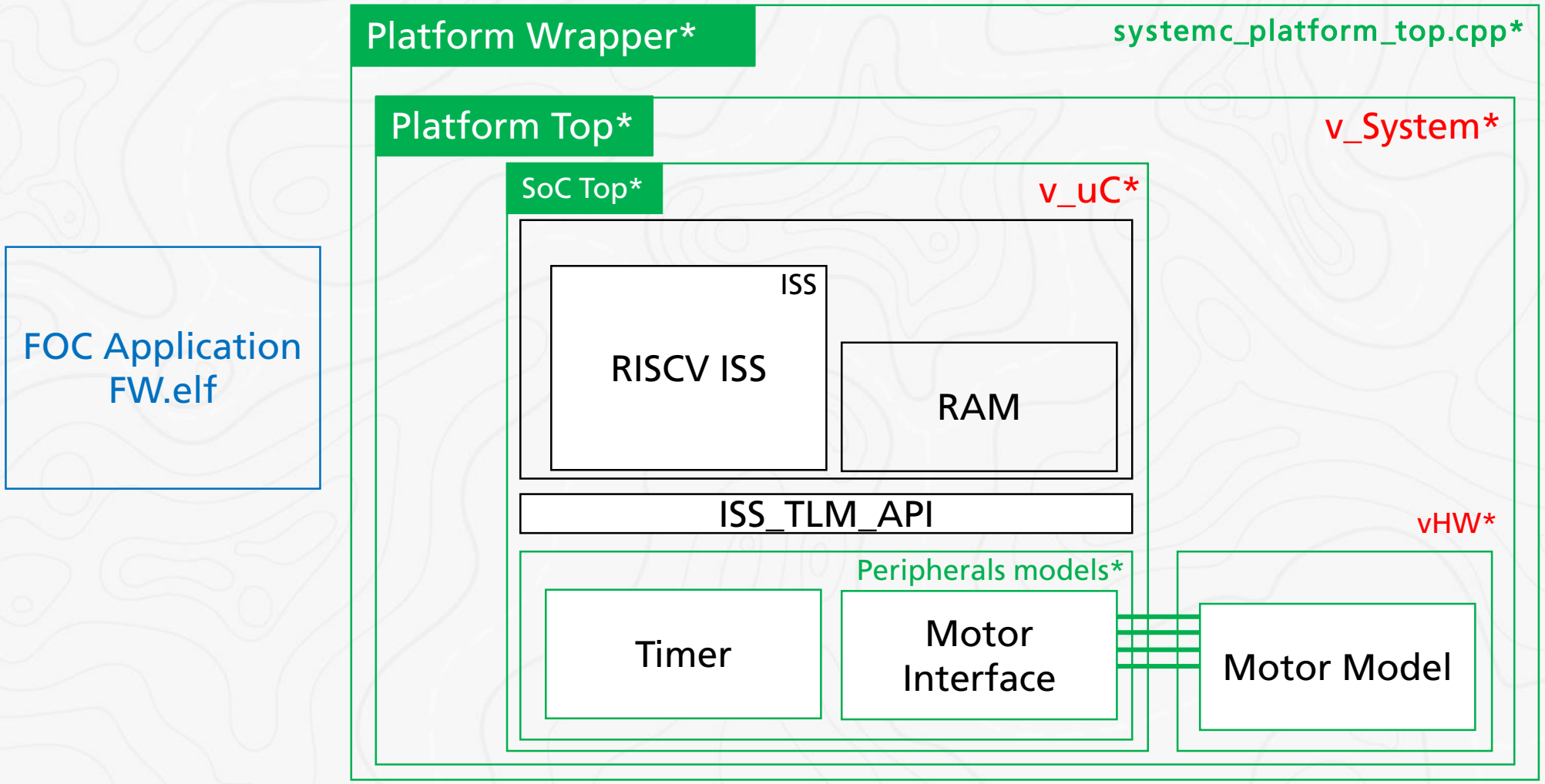
- Mapping of available requirements
- Build Untimed VP, reuse existing model IP blocks
- Porting software application

Item	Specification
SysReq_1	32bits RISCV MCU SoC
SysReq_2	FOC Algorithm based on FreeRTOS



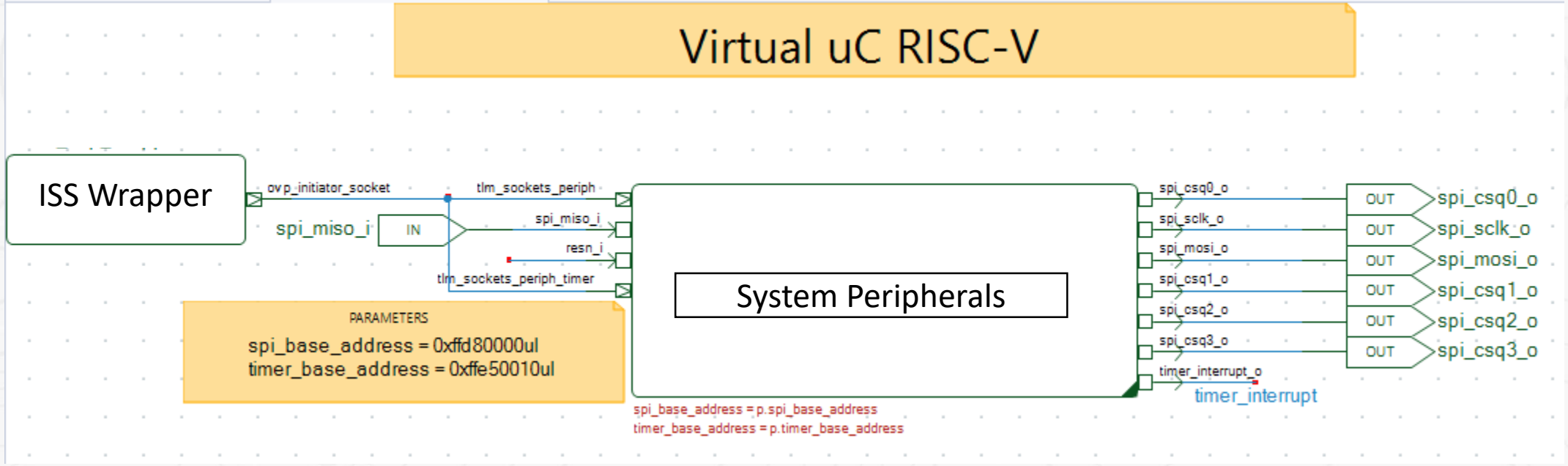
FOC Application FW.elf

Virtual Prototype Integration

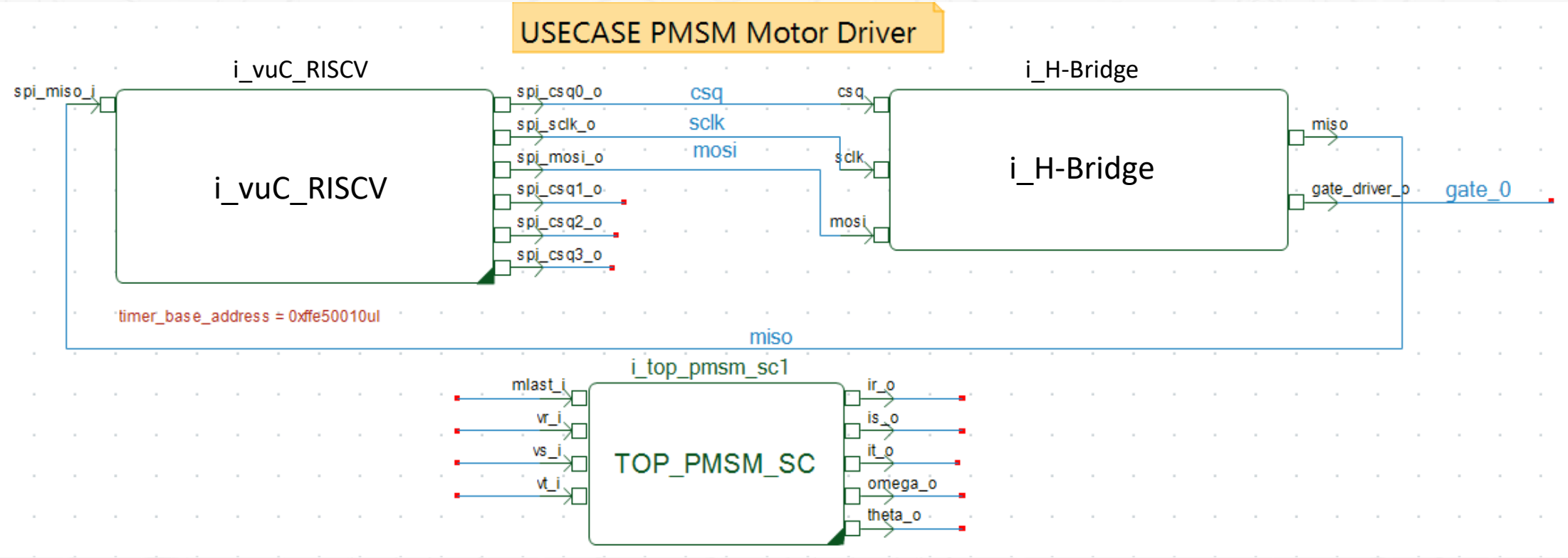


In the way to proof the concept

- Microcontroller model components



In the way to proof the concept



Proposed methodology

- Timing characterization of existing system

HW – *Motor Controller*

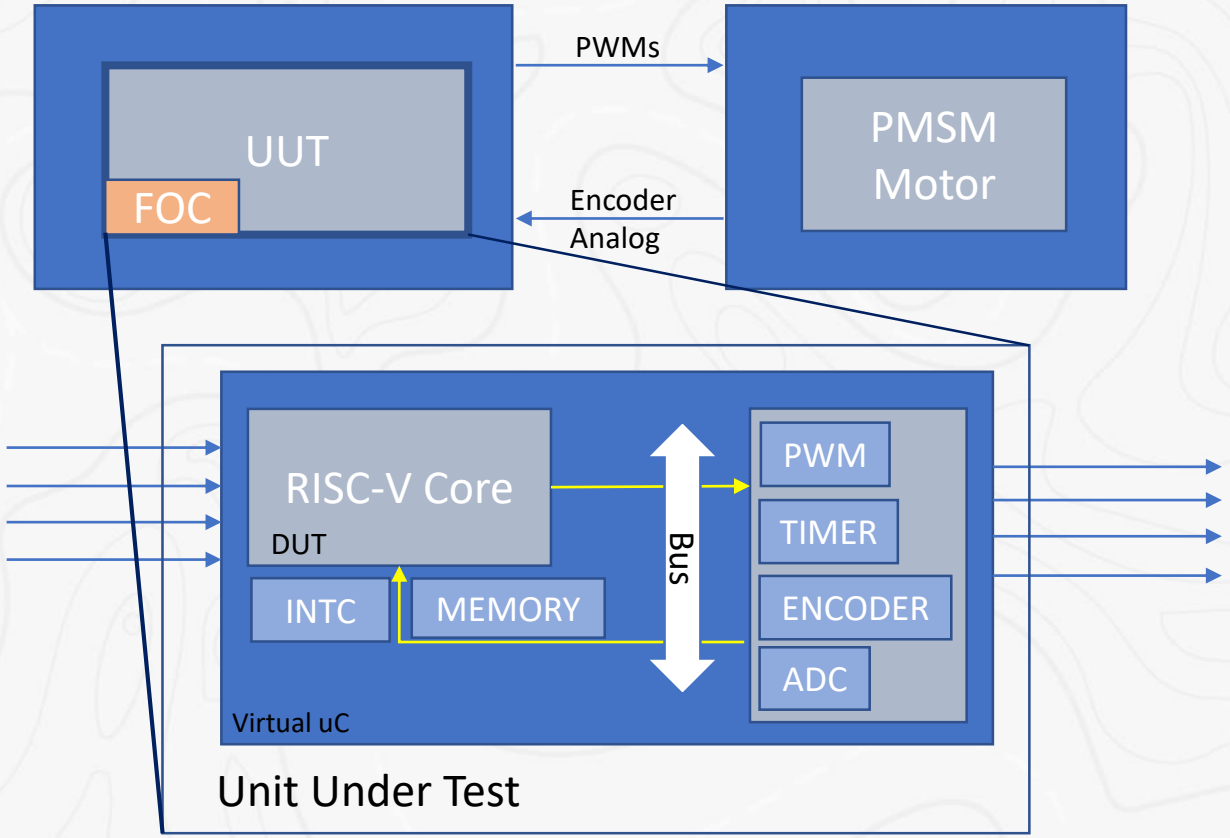
- Unit Under Test(UUT)

Complete embedded system *(Motor Controller)*

- Device Under Test(DUT)

Microcontroller core

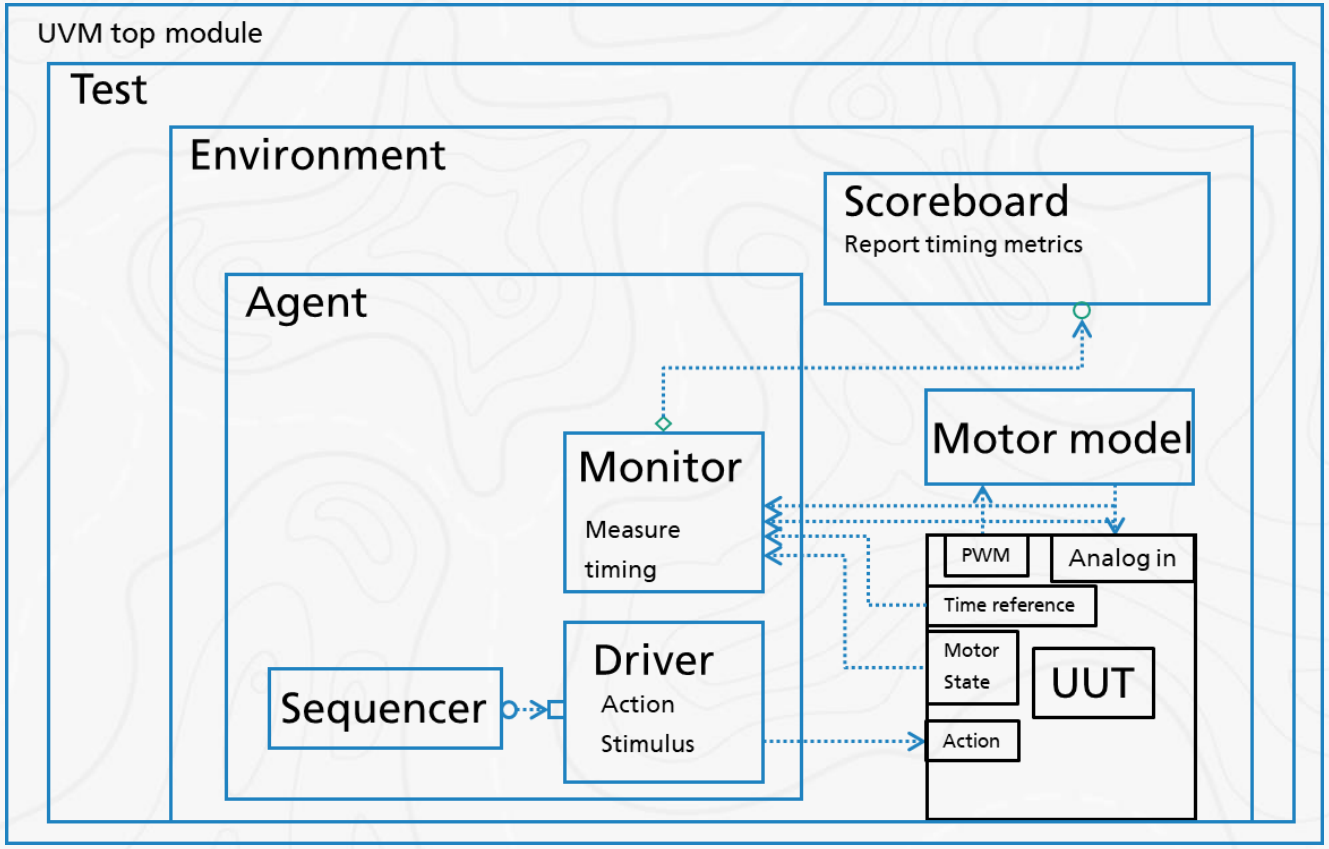
- SW – *Field Oriented Control*



UVM-SystemC testbench

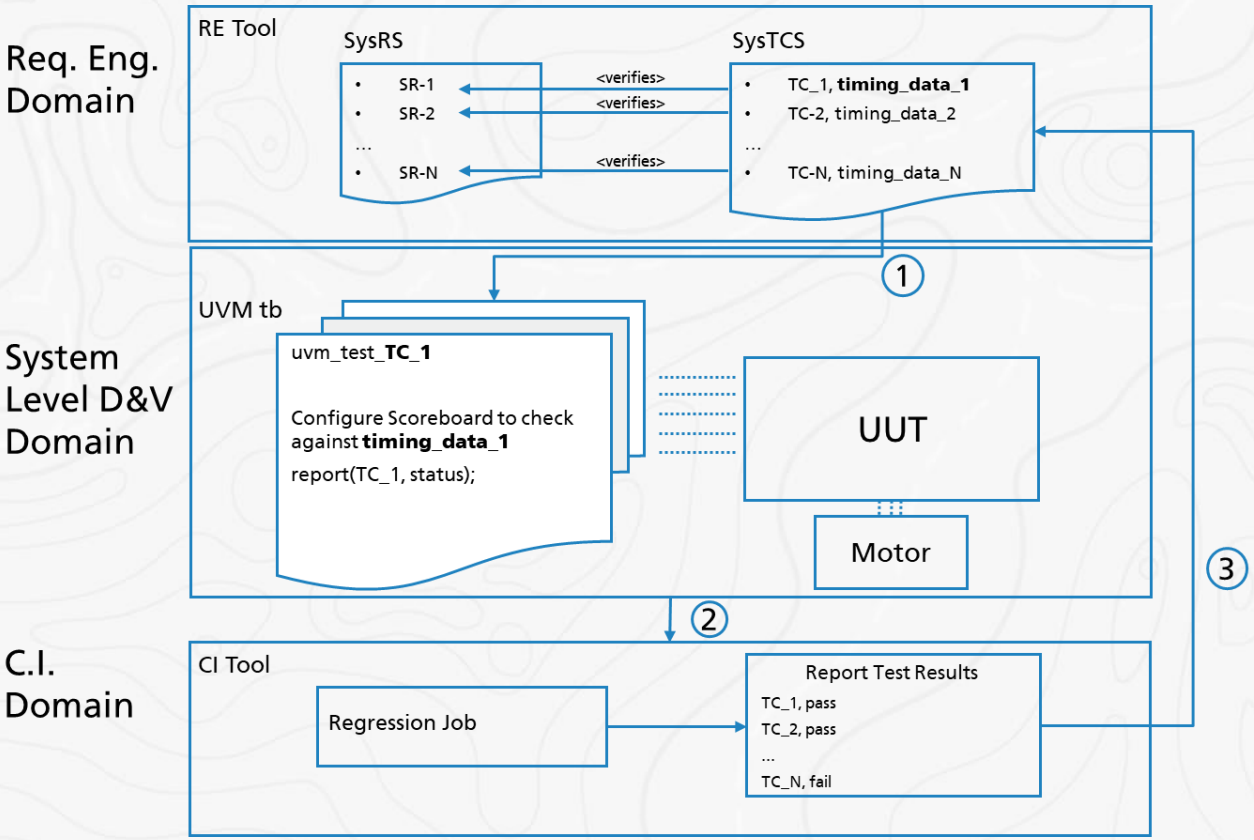
UVM Support legacy system characterization

- Configure motor parameters
- Control motor states
- Manage stimulus sequences by a constrained random time
- Monitor results and fill the scoreboard



Proposed methodology

- Verify timing contrains and compare



① UVM test case implementation

②Status and log of each test run by CI Tool

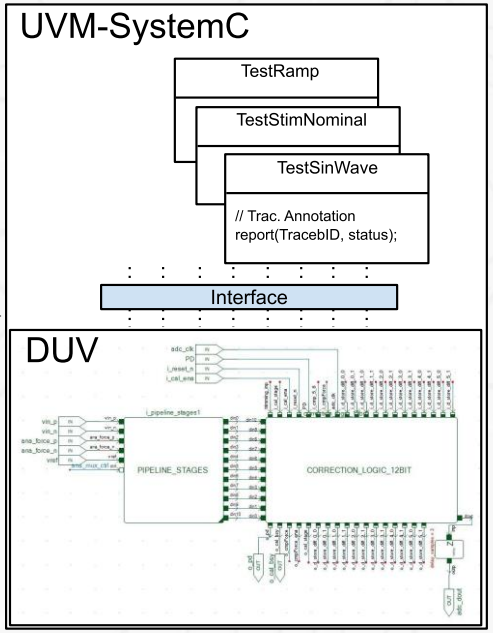
③Pass/Fail feedback to RE Tool

Link to Requirements Management flow

- Example use case

System Test Case Specification (Polarion)

①

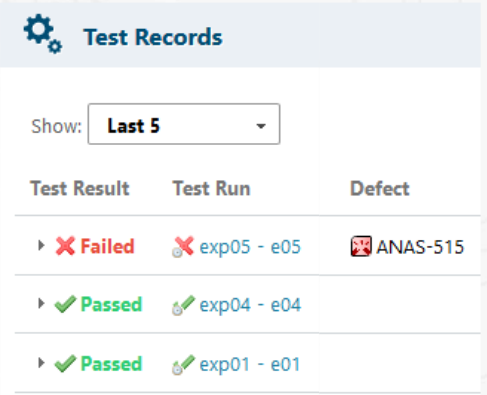


UVM-SystemC Testbench (COSIDE)

②

Regression Run (Jenkins)

③



Publish Results (Polarion)

Summary

- Simplification of the hardware exploration process
- Virtual CPS help to identify issues with application software long before an intended redesign
- Reuse of test cases -> continuously refine requirements and testbenches
- Evaluate viability of the new concept prototype while still complying with System Requirements Specification

Questions

VALU3S

www.valu3s.eu

ACKNOWLEDGEMENT
This work has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852 (VALU3S). The JU receives support from the European Union's Horizon 2020 research and innovation program and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.