GSTP2-IMCM

ESTEC/Contract N° 13492/99/NL/PA

Final Presentation

December 2004

IMCM Project Assessment

- BRISC architecture
  - Targeted applications
  - Architecture overview
- IMCM ASIC HBRISC2 development logic
- HBRISC2 validation strategy
- HBRISC2 application development process
- Production& validation tools
  - Synthesizer
  - Macro Library
  - Assembler
  - Unit test tool
  - Validation of the macros and Simulink library
  - Win-Seracq tool
- IMCM demonstration application
- HBRISC2 perspectives
  - Current and coming applications
  - Towards HBRISC2 *Application*+ Tool
- IMCM conclusion

BRISC Architecture Overview

- **Targeted to the command of high performance electrical motors in critical applications**
- **True from the very first release of BRISC & continuously improved**
  - **Application key requirements**
    - Versatile, standard product
    - Hard real-time, with tight HW-SW synchronization
    - Fully predictable execution time
    - No corner-case caused by the use of digital domain
    - High visibility on the internal functioning
  - **Architecture impact**
    - Data path optimized for vector control, motor modeling...
    - Intrinsic high computing power: floating multiplication, division (HBRISC2)
    - Saturating arithmetic (no exceptions)
    - RISC machine
    - High digital function integration (avoid glue logic, on-board µP or FPGA...)
    - High Speed serial link dedicated to reporting
    - Straightforward compatibility to standard components (ADC, RAM, Flash, EEPROM,...)

BRISC Architecture Overview

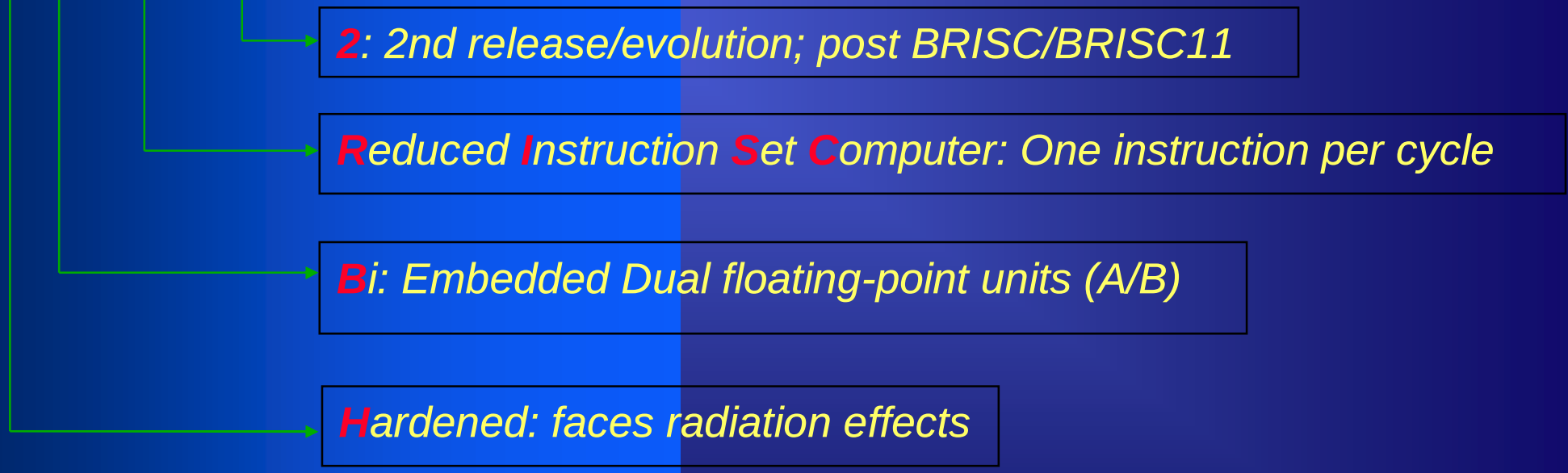
- **(H)BRISC2 is specifically targeted to Hi-Rel applications**
  - **Space environment**
    - Radiation effects (neutrons, heaving ions...)
      - Automatic SEU correction with :
        - minimal interruption and user-transparent execution
        - No impact on board frequency
    - Temperature: -55°C to 125°C
    - Technology reliability
  - **Aeronautics/Military environment**
    - Domain is more and more sensitive to SEU effects
    - Temperature: -55°C to 125°C
    - Technology reliability

BRISC Architecture Overview

- **Provide main peripheral functions required in loop systems (only HBRISC2):**
  - **Includes:**
    - Communication with sensors feed-back (ADC)
    - PWM generation
    - Communication with outer peripherals (on-board computer, debugging environment,...)
    - Automatic live at-power up
    - ...
  - **Autonomous functions to minimize CPU load impact**
    - Programmable in SW
    - Flexible with standard components

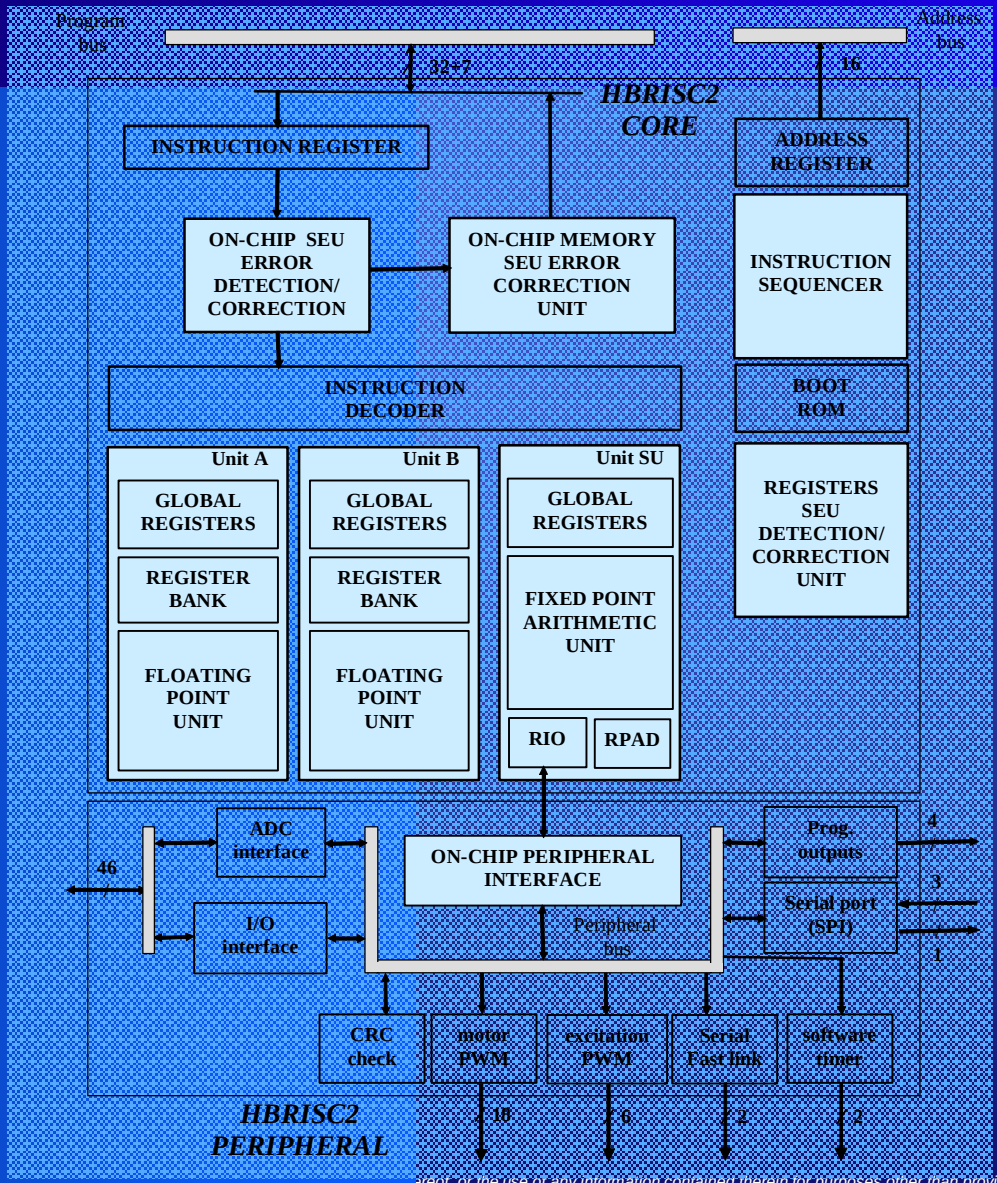
BRISC Architecture Overview

H B RISC 2: Latest release of BRISC architecture



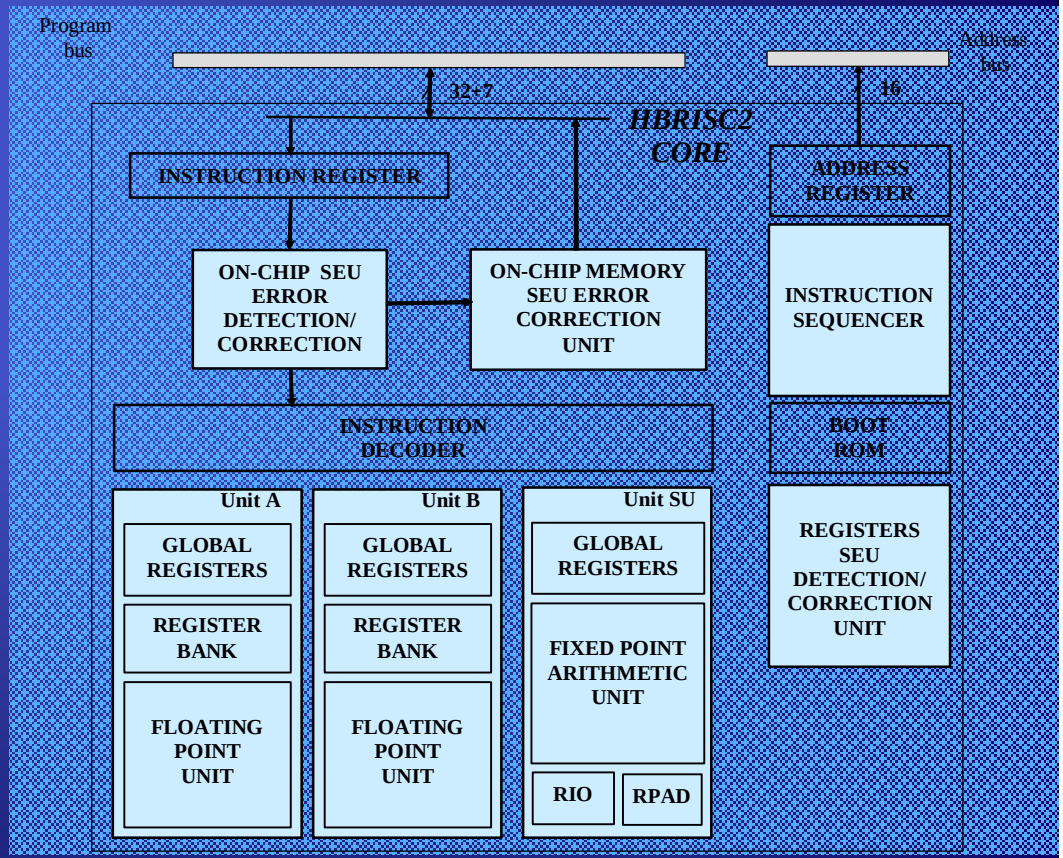
BRISC Architecture Overview

HBRISC2



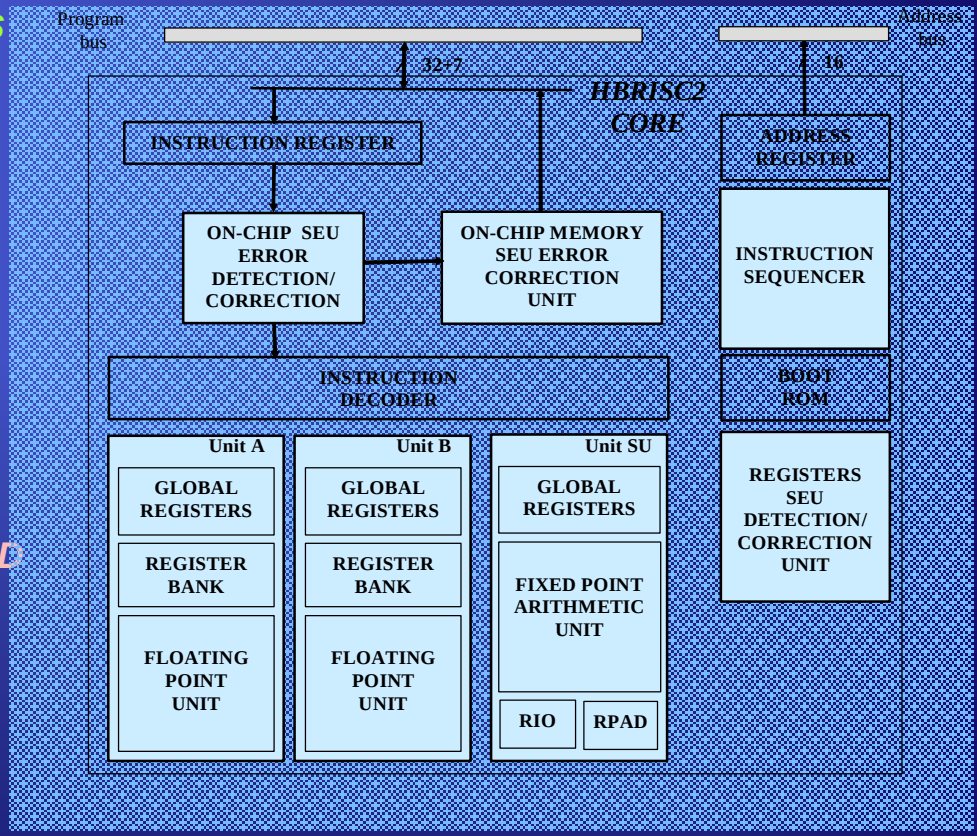
BRISC Architecture Overview

- **RISC architecture**
  - *Deterministic code execution time*
  - *Allows easy code optimization*
  - *Instruction decoding and sequencing is simple*
  - *Low circuit area*
- **Data definition: high precision and magnitude**
  - *16 bits mantissa, 2's complement*
  - *8 bits exponent, 2's complement*
- ***Dual floating point unit***
  - *Vectorial data-path with SIMD architecture*
  - *Exploits intrinsic parallelism of targeted applications*

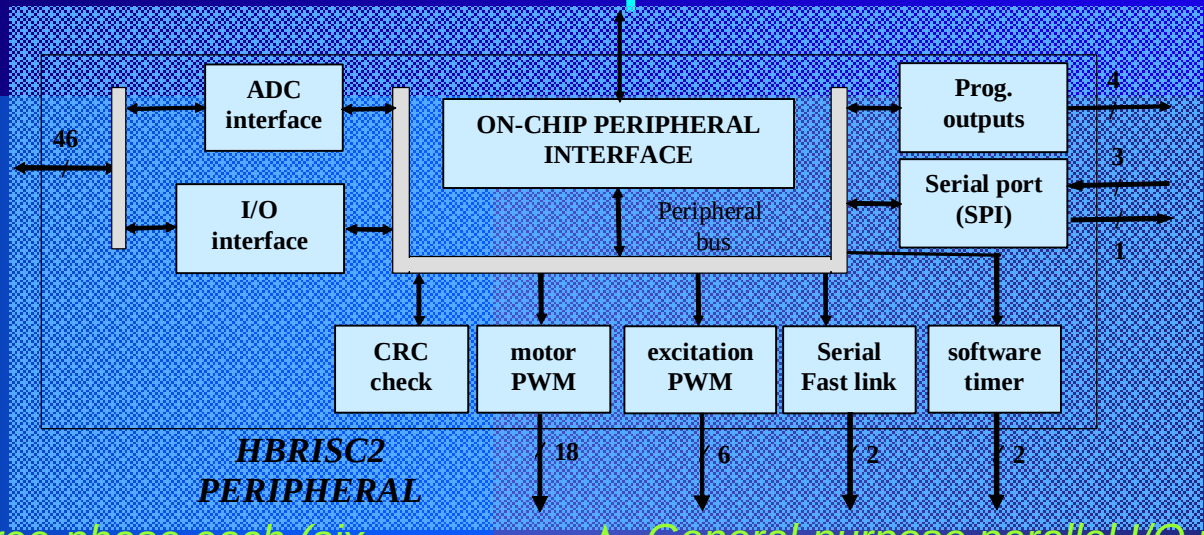


BRISC Architecture Overview

- *On-chip data memory: organized in 16 banks of 8 registers each*
  - *Avoid use of external memory to store frequently used data*
  - *Access time restricted to one instruction cycle (>< several cycles for external memory)*
- *SU unit: Special Unit dedicated to address computing*
  - *16-bit integer*
  - *Dedicated communication registers: RIO&RPAD*
- *Arithmetic*
  - *Saturating*
  - *Exception-free*



BRISC Architecture Overview: Peripheral Unit



- *3 independent three-phase each (six transistors), PWM generators*
  - ***Autonomous, fully programmable, dead time software control***
- *6 channels for sensors excitation*
  - ***Complementary outputs, programmable dead time***
- *SPI*
  - ***Communication with intelligent external peripherals***
- *High Speed serial link for data reporting and system-level debugging*
- *General-purpose parallel I/O port*
  - ***Programmable, designed for slow peripherals such as EEPROM or Flash memories***
- *Software timer*
  - ***Synchronization between software and hardware***
- *Programmable ADC interface*
  - ***Configurable for any ADC, up to 8 samples per software cycle***
- *CRC computation unit*
  - ***Checksum of uploaded code, from I/O or SPI***
- *4 general-purpose programmable outputs*

BRISC Architecture Overview: Radiation Effects

- ***LATCH-UP***
  - *Parasitic shortcut between VDD and GND*
  - *Caused by heavy ions*

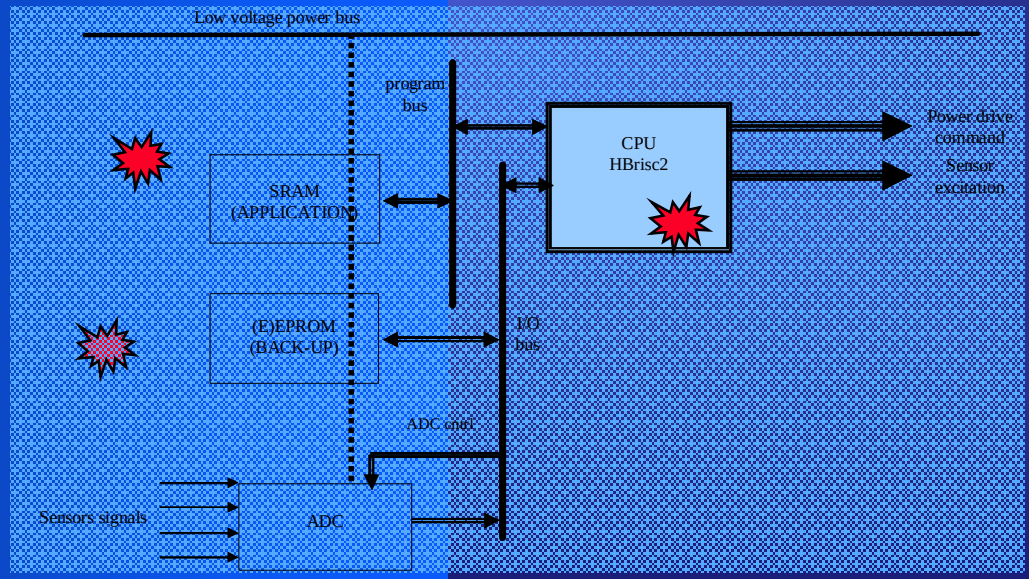
  → HBRISC technology (ATMEL MH1RT) guarantees Latch-Up threshold to 80 MeV/g/cm2

- ***TID (Total Ionizing Dose)***
  - *Undesired chip consumption increase starting from a threshold (in RAD)*
  - *Caused by heavy energy particles*

  → HBRISC technology (ATMEL MH1RT) guarantees 200 KRAD (Si) radiation level

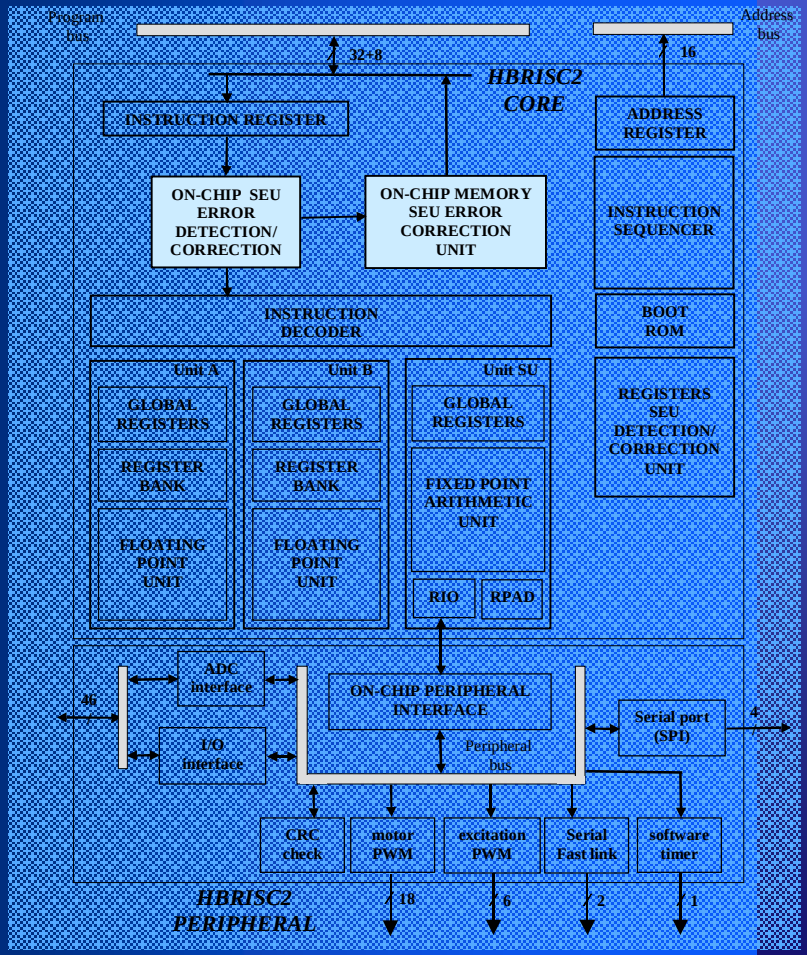
BRISC Architecture Overview: Radiation Effects

- ***SEU (Single Event Upset)***
  - ***Bit flipping on storage cells (register, memory cell…)***
  - ***Caused by heavy ions radiations***
  - ***Critical components in typical HBRISC board: CPU registers, SRAM, Flash/EEPROM***



BRISC Architecture Overview: Radiation Effects

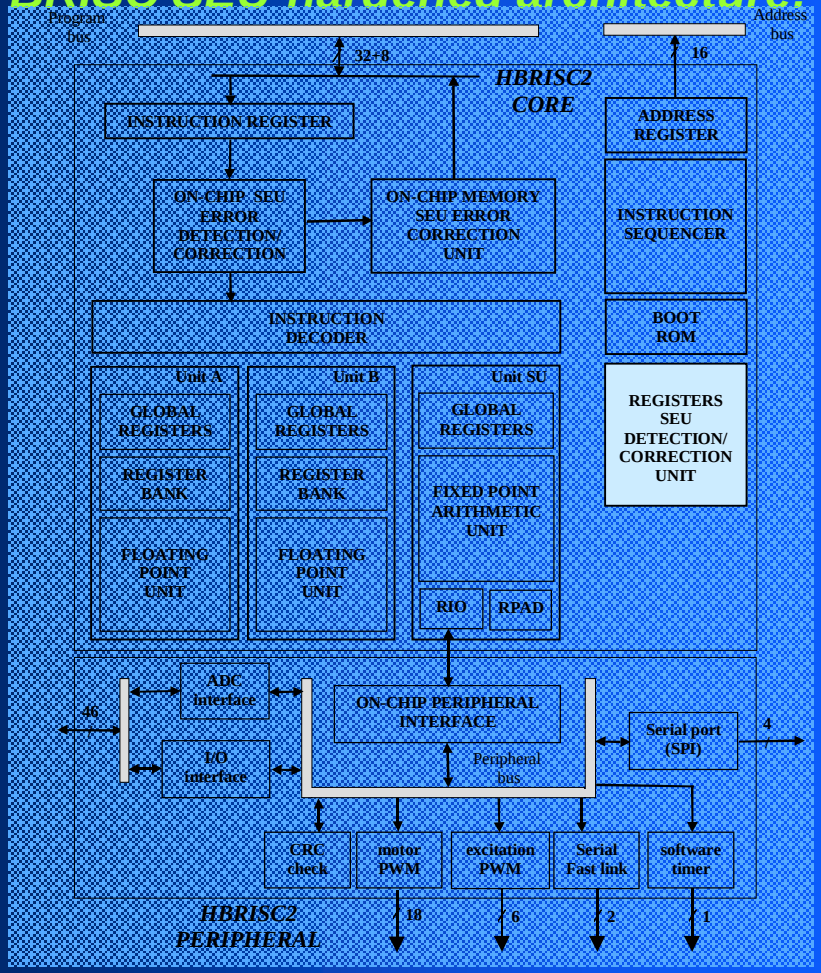
- ***BRISC SEU-hardened architecture: instruction/data SRAM protection***



- "LIVE" CORRECTION BEFORE INSTRUCTION DECODING
- AUTOMATIC SRAM CORRECTION ON HBRISC2 STOLEN CYCLES
- MODIFIED HAMMING CODE
  - ⇒ ***detects and corrects all 1-bit errors***
  - ⇒ ***detects all 2-bit errors***
  - ⇒ ***detects part of multiple-bit errors***
- NO ADDED DELAY DURING INSTRUCTION FETCH
- HARDWARE REPORTING

BRISC Architecture Overview: Radiation Effects

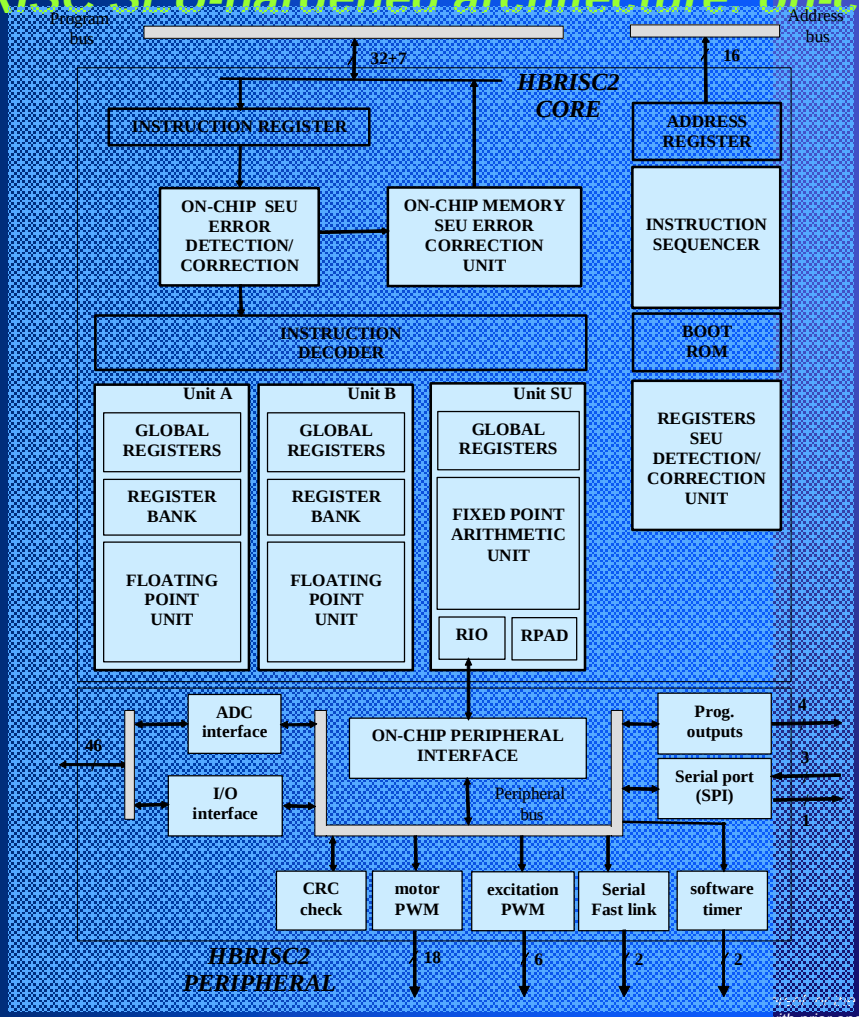
- ***BRISC SEU-hardened architecture: on-chip (bank register) SRAM protection***



- DETECTION/CORRECTION MADE DURING INSTRUCTION EXECUTION
- ON ERROR, RE-EXECUTION FOLLOWS CORRECTION
- MODIFIED HAMMING CODE AS PER EXTERNAL SRAM
- HARDWARE REPORTING

BRISC Architecture Overview: Radiation Effects

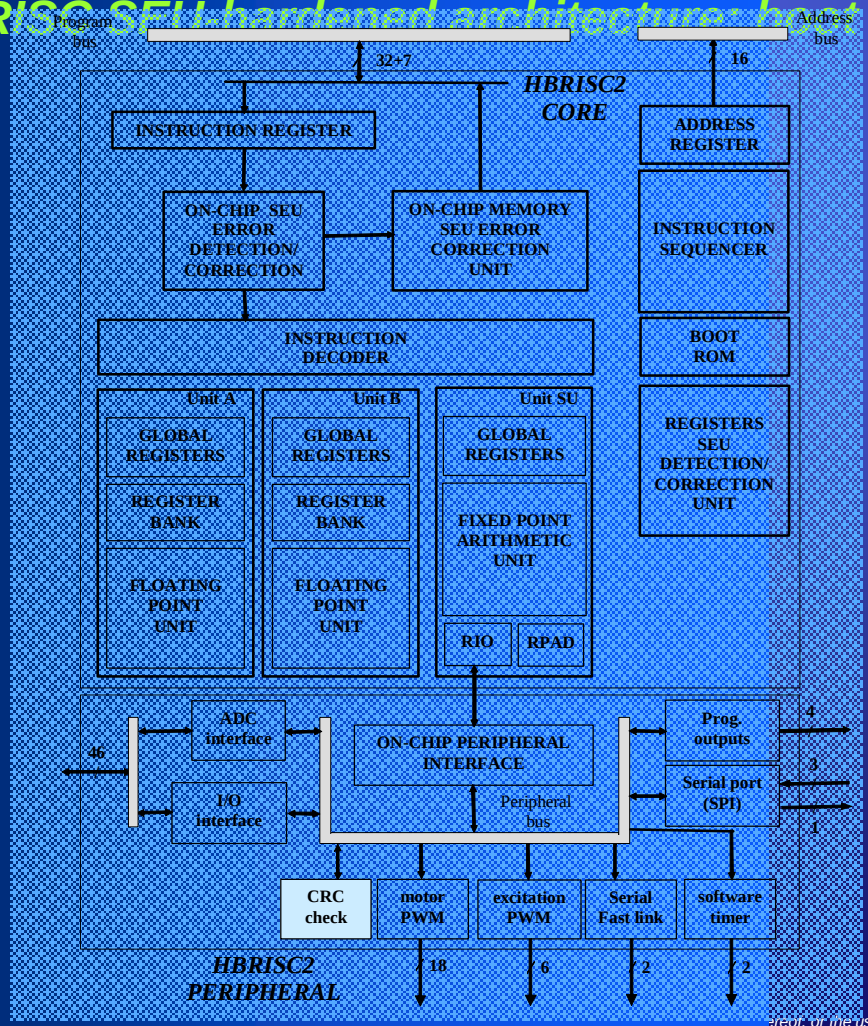
- ***BRISC SEU-hardened architecture: on-chip registers protection***



- USE OF "HARDENED-SEU" REGISTERS AND LATCHES
- AVAILABLE IN ATMEL MH1RT STANDARD LIBRARY
- ALL REGISTERS/LATCHES HAVE BEEN HARDENED EXCEPT/
  - BANK REGISTERS (TOO AREA CONSUMING
  - JTAG BOUNDARY_SCAN CHAIN (UNUSED ON FLIGHT)

BRISC Architecture Overview: Radiation Effects

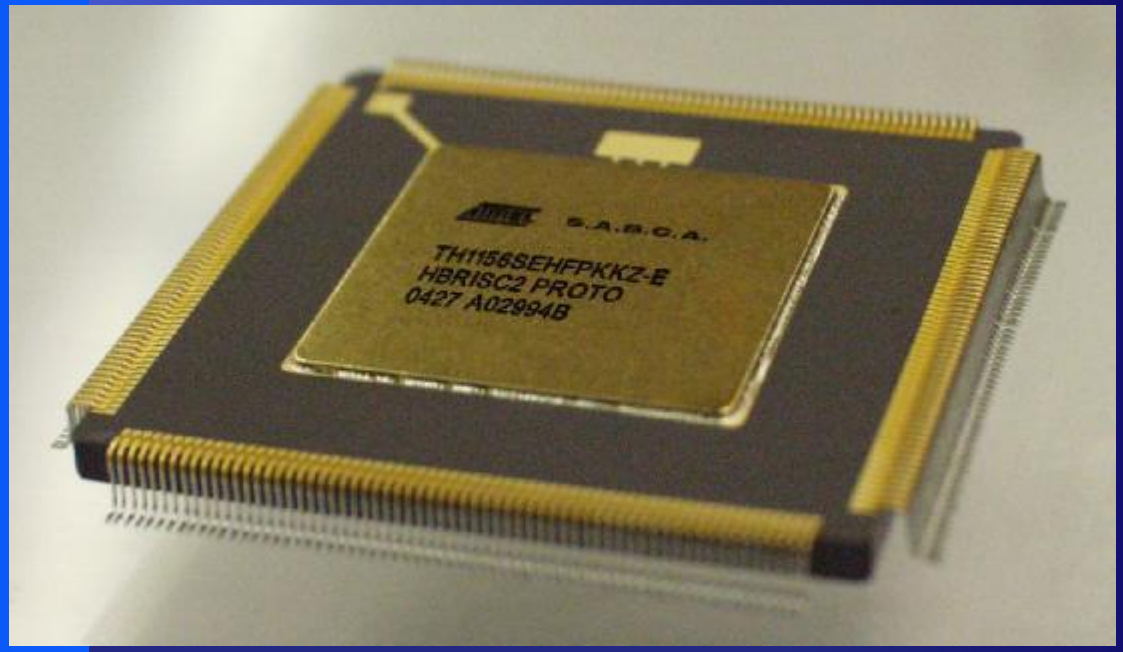
- ***BRISC SEU hardened architecture: boot rom protection***



- BASED ON CRC BLOCK
- ONLY ERROR DETECTION IS POSSIBLE (NO CORRECTION PERFORMED)
- HARDWARE AND SOFTWARE REPORTING

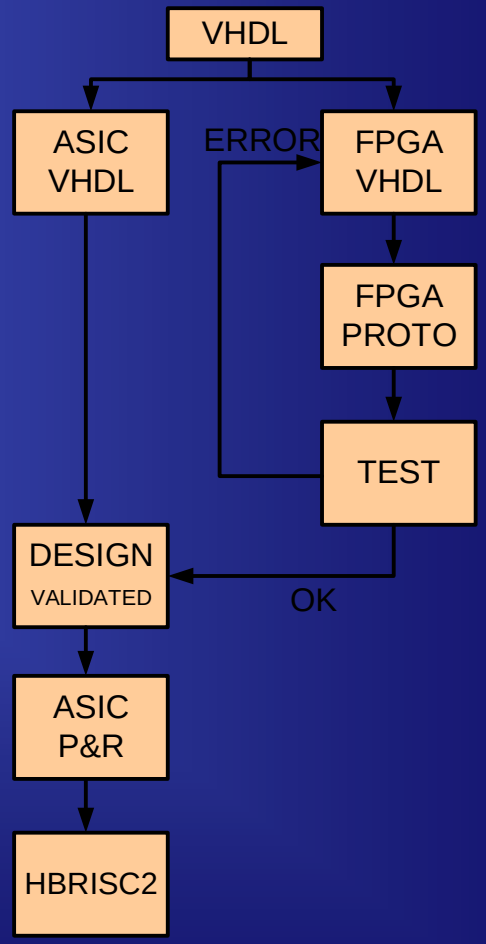
Brisc Architecture Overview: ASIC Technology

- ***ATMEL MH1RT***
  - *Rad-Tolerant*
  - *Latch-up immune (> 80 Mev/g/cm2)*
  - *SEU hardened registers*
  - *Gate-array 0.35 µm*
  - *5V I/O pads*
  - *3.3 V Core*
- ***HBRISC2***
  - *Matrix: MH1_156E*
  - *Package: MQFP-256*
  - *4 on-chip dual-port RAM*
    - *Bank register*
    - *SPI interface buffers*



IMCM ASIC -*HBRISC2*- development logic

- ***Final product is an ASIC (Application Specific Integrated Circuit)***
  - ***Simuations runs are time-consuming: extensive testing is difficult to achieve***
  - ***Major design error must be avoided: ASIC iterations are long and expensive***
  - ***HBRISC2 architecture is complex to test: thorough testing is mandatory…***
- ***Solution: Test Platform with FPGA prototyping***
  - ***Early debugging of HBRISC2 VHDL sources***
  - ***Thorough tests can be made on breadboard***
  - ***A physical implementation is available: SW team can start developing macros and application***
  - ***Design-to-silicon cycle is short (typically a few hours)***
  - ***Note: "At-speed" tests(60 MHz) are not feasible***



IMCM ASIC -*HBRISC2*- development logic

- ***VHDL-based design***
  - ***Schematic-based designs are nowadays obsolete***
  - ***Portable design (suits FPGA prototyping methodology)***
    - *Differences between ASIC VHDL and FPGA VHDL:*
      - *Embedded on-chip memories*
      - *Programmable outputs*
  - ***Increase readability***
  - ***Currently supported and encourages by all founders***

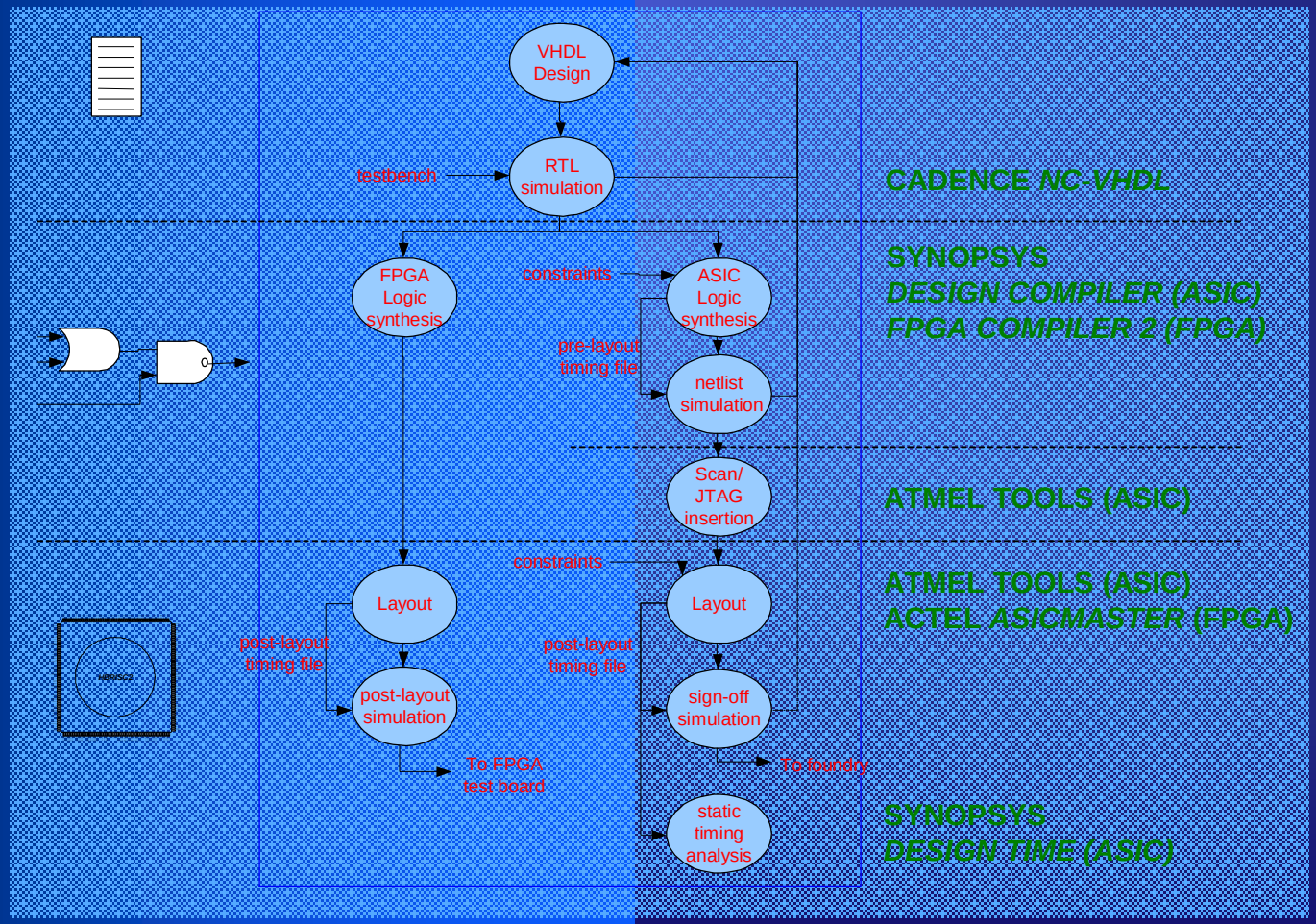
- ***Use of industry well-known CAD tools: Synopsys, Cadence…***
  - ***Efficiency is world-wide recognized***
  - ***Already owned by SABCA for past and current designs***
  - ***Scan/JTAG insertion and ATPG is made by Atmel***
  - ***Layout made by Atmel***

IMCM ASIC -*HBRISC2*- development logic

- ***Implementation: Selected FPGA is Actel ProASIC A500K***
  - ***On-board reprogrammable***
  - ***SABCA already works with Actel***
  - ***Design partitioned in two FPGAs:***
    - *Core Unit* *A500K270*
    - *Peripheral unit* *A500K180*

IMCM ASIC -*HBRISC2*- development logic

- ***Design Flow***



Hbrisc2 Circuit Validation Strategy: Items Under Test

- Purpose: verification versus TNS0015 ("*HBRISC2 design and architectural specification*").
- Items under test (described in VP0263 "HBRISC2 validation plan):
  - Functional
    - **Instruction set**
    - **Peripherals Units: PWM, HSSL, SPI**
    - **Special features: Upload,...**
  - AC parameters
    - **I/O propagation delays**
    - **I/O setup/hold timings**
    - **Max frequency**
  - DC parameters
    - **I/O pads voltages levels**
    - **I/O pads current levels,**
    - **Consumption**
  - Environment
    - **Temperature (-55°C;25°C;125°C)**
    - **SEU correction mechanism**
    - **...**

Hbrisc2 Circuit Validation Strategy: Hbrisc2 Models

- 1st Model: ASIC
  - **Final product: PIN number is 5962-01B0106QYCFPK**
  - **Corresponds to circuit manufactured by ATMEL**
- 2nd Model: VHDL RTL model
  - **VHDL description of HBRISC2 design**
  - **Definition is sufficient to verify functional requirements but not**
    - AC parameters
    - DC parameters
    - Environmental requirements except SEU

Hbrisc2 Circuit Validation Strategy: Hbrisc2 Models

- 3rd Model: VHDL gate model
  - **Corresponds to netlist (gate-level) of HBRISC2 design**
  - **Model integrates post-layout timings**
  - **Definition is sufficient to verify functional requirements but not**
    - DC parameters
    - Environmental requirements except SEU
- 4th Model: FPGA Emulator
  - **HBRISC2 Prototype used during for design debugging**
  - **Only functional requirements can be tested**

Hbrisc2 Circuit Validation Strategy: Test Means

- 1st test platform: VHDL simulator
  - ***Allows RTL-level, gate-level (pre & post layout simulations)***
  - ***Suited for small test patterns (simulation is slow especially during gate-level)***
  - ***Easy debugging (full visibility of all internal nodes)***

- 2nd test platform: Atmel tester (Sentry 15)
  - ***Use of final physical implementation***
  - ***Number of test vectors is limited (Atmel quota)***
  - ***Atmel tester frequency limited to 20 MHz.***

Hbrisc2 Circuit Validation Strategy: Test Means

- 3rd test platform: HBRISC2 test board
  - ***Use of final physical implementation***
  - ***"At-speed" test is possible at ambient temperature.***

- 4th test platform: FPGA emulator
  - ***Use of an intermediate implementation of HBRISC2.***
  - ***Interest is limited regarding HBRISC2 validation but permits :***
    - *Early macros/application development.*
    - *Efficient VHDL design debugging*
    - *Re-use of test patterns in the scope of HBRISC2 (ASIC) validation*

Hbrisc2 Circuit Validation Strategy: Verification Matrix

	VHDL Simulator	*HBRISC2 Test Board*	*Atmel Tester*
Functional	**Limited** MUT: GATE	**Re-use of tests developed with FPGA emulator (macros…)** MUT: ASIC	**Limited** MUT: ASIC
DC parameters	**NA**	**NA**	**All DC parameters** MUT: ASIC
AC parameters	**All AC parameters (including max. frequency)** MUT: GATE	**Max. frequency @ ambient temperature** MUT: ASIC	**All parameters except max. frequency** MUT: ASIC
Environment	**SEU tests** MUT: GATE	**NA**	**Temperature and supply voltage** MUT: ASIC

MUT = Model Under Test

Hbrisc2 Circuit Validation Strategy: Test Results

- **Functional**
  - **Successfully passed on HBRISC2 board and Atmel tester**
  - **Complete Test Report is in TR0024**

- **AC parameters**
  - **Successfully passed on Atmel tester**
  - **Successfully passed on GATE modelHBRISC2 board**
    - Max frequency on STA: 54 MHz
  - **Successfully passed on HBRISC2 board**
    - Tests run @ 60 MHz (at ambient temperature)
  - **Complete Test Reports is in TR0024 and TN0021**

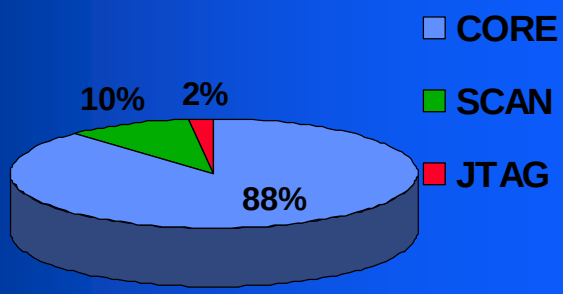
Hbrisc2 Circuit Validation Strategy: Test Results

- **DC parameters**
  - **Successfully passed on Atmel tester: measurements within specification in AID (TNS0211).**
    - Dynamic consumption (Core+I/O): < 0.5 W
  - **Complete Test report is in TR0024**

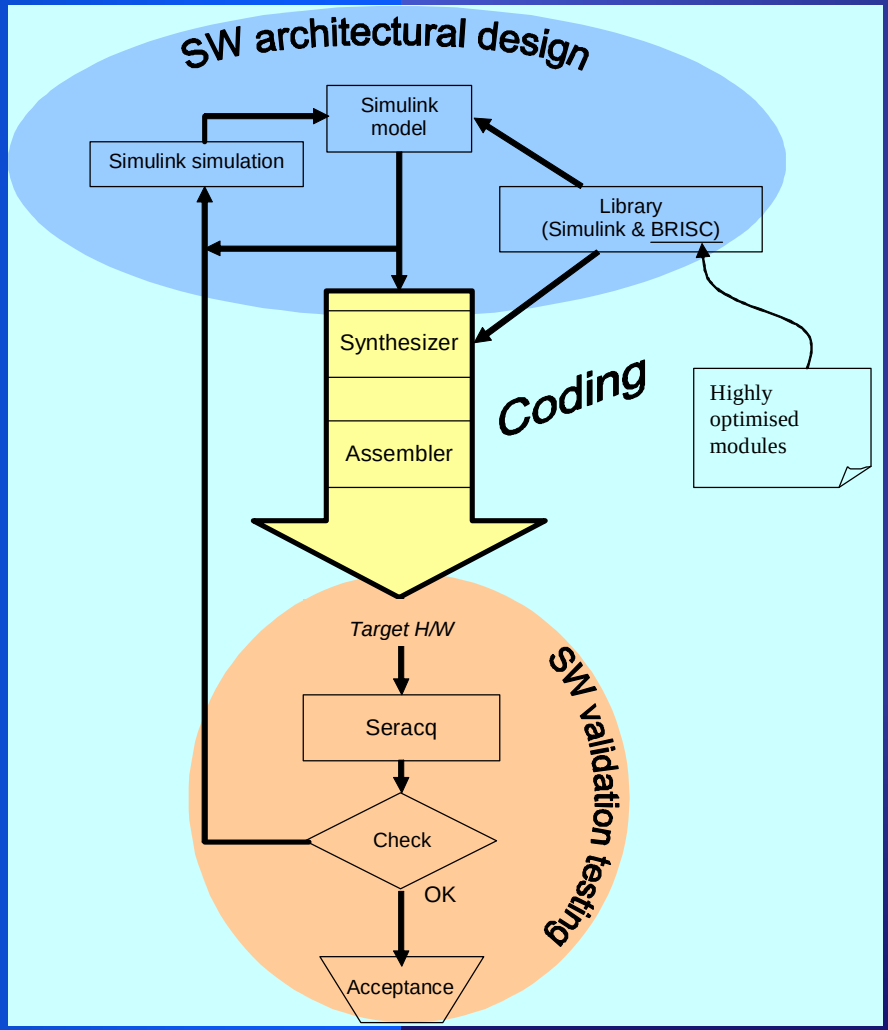
- **Environment**
  - **Successfully passed on Atmel tester: measurements within specification in AID (TNS0211).**
  - **Test report in TR0024**
  - **SEU test plan in VP262 “SEU validation plan”**
  - **SEU test report in TR0257 “SEU validation report”**

Hbrisc2 Circuit Validation Strategy: Hbrisc2 Status

- PRE-layout frequency: 50 MHz
- POST-layout frequency: 54 MHz
- Design area: *260 074 cells*
- ATPG Test coverage: *97.53 %*
- cells sites used*:* *54.05%*
- IO sites used*:* *40.29%*
- SEU Flip-flop count*:* *7039*
- Non-SEU flip-flops*:* *597*
- Latches count*:* *39*
- Cells distribution:



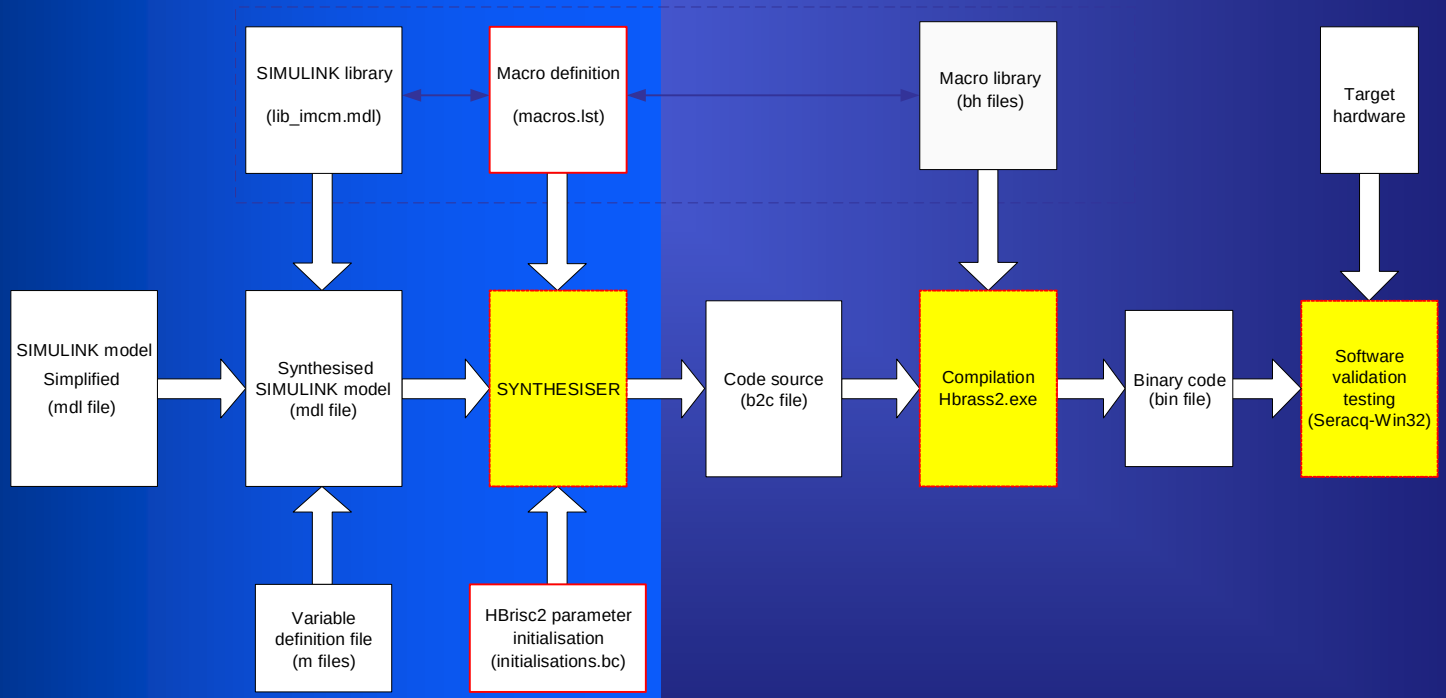
HBRISC2 application development process



Description of the Production and validation tools

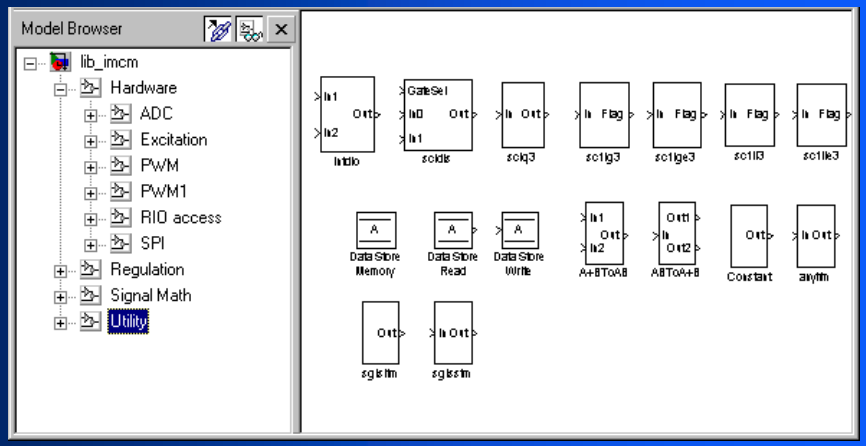
- A certain number of tools have been developed around Brisc processors in order to allow efficient production of good quality code.
- The main tools are the following:
  - The synthesiser that takes a Simulink diagram and generates assembly code.
  - The Assembler that produces binary code.
  - The Seracq-Win32 that supports tests on target hardware. It performs uploading of a program, running of command scripts, and acquisition and visualisation of parameters transmitted through the HSSL.
  - The total_nw program that allows unit tests of the macros to be executed all together in a single run.
  - A set of scripts that allows unit tests of the Simulink blocks to be performed automatically.

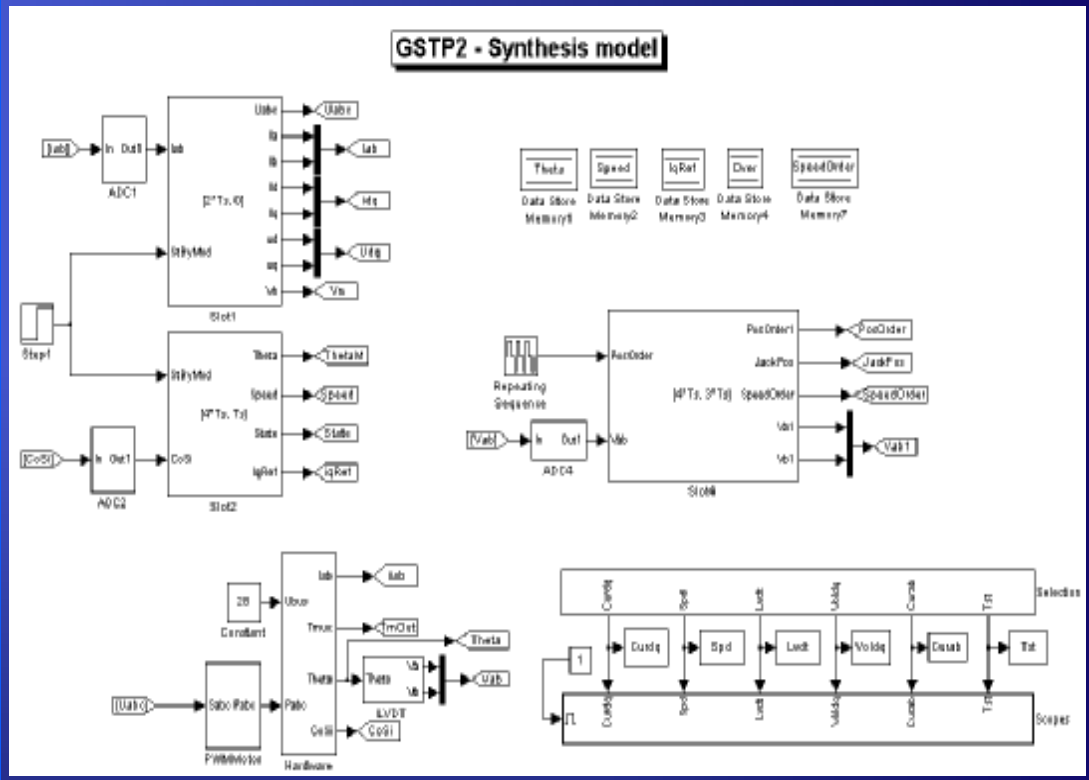
Production chain of HBRISC2 application SW



The synthesiser and the software libraries (1)

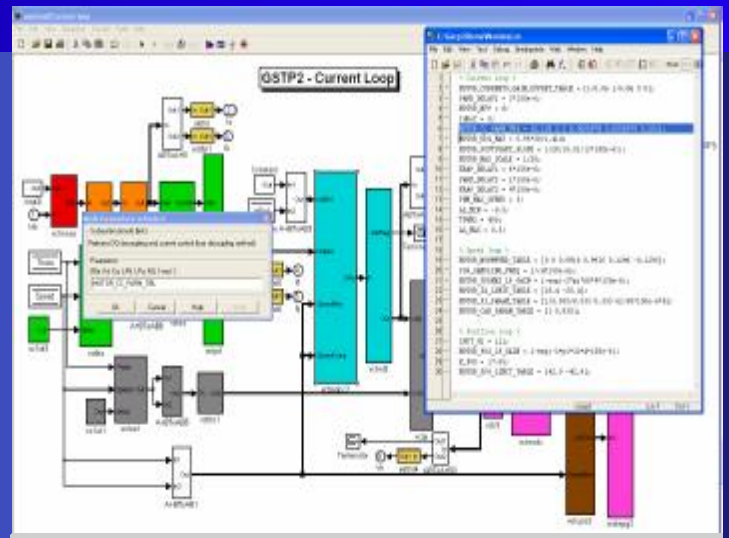
- The actuator modelling and tuning is performed with Simulink.
- First, a high-level model of the whole actuation system (control loop, motor, load, power electronics) is done.
- Afterwards, the control loop is re-written with blocks coming exclusively from the HBRISC2 library.
- Simulations will show that both models are equivalent.

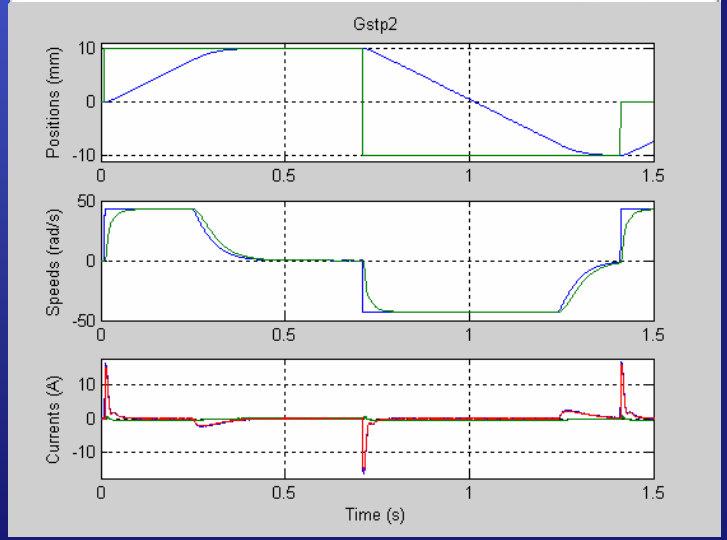




The synthesiser and the software libraries (2)

- The different blocks of the model have their own parameters.
- A matlab file called “val.m” contains all the parameters used in the control loops.
- Representative simulations can be performed – The same code is used for simulations and for synthesis.
- Simulation results can be stored and reused (for instance) in open-loop hardware tests.
- The synthesiser generates assembly code made exclusively of macro calls, file inclusions, and constant definitions (no assembly code as such).





Validation of the synthesiser

- The synthesiser is a single executable, but is divided in several modules:
  - A parser that reads the main input files and builds an equivalent object structure in memory
  - A sorter that orders the blocks and slots.
  - An assigner that gives units and registers to the macros, so the macros calls can be fulfilled.
  - A generator that produces the final code, and inserts intermediate macro calls to ensure integrity of the information.
- Specific test cases have been created to check that each of these modules worked correctly.
- A test case with the same complexity as the application has been synthesised and its generated code carefully checked.
- Integration and validation tests have used code generated by the synthesiser.

Macro library

Controller			Regulation		
VCTCutPowerIfStandby3			SC1Dimmer3 SC1Limiter2 SC1OverflowWithSaturation SC1PositionToSpeed3 SCLActuatorPosition VCTActuatorPosition SCLElectricalAngle3 SCLElectricalAngle SCLMultiplication VCTModulusLimiter3 VCTNormalizeFundamental2	SCLBackwardEulerLowPass3 SCLBackwardEulerPI2 SCLCorrectAfterRegulator2 SCLGain SCLRegulationError VCTMotorCreateDutyCycles VCTMotorDecoupledCurrentControl2 VCTMotorNeutralModulation VCTLvdtToPosition VCTSecondOrderFilter2	VCTClarkeDirect VCTClarkeInverse VCTGainOffset2 VCTParkDirect VCTParkInverse VCTRamp3 VCTFilter52
			Math		
			SCLOffsetGain2 SCLOneBy SCLOneBySqrt VCTCartesianToPolar	VCTArcTan VCTCosSin SCLAdd	
1553 stack			**Utility**		
	removed		ANYCopyRegister* ANYCopyRegisterA* ANYCopyRegisterB* ANYCopyRegisterSU* ANYLoadFromMem INTRepeatLoop3* ANYLoadImmediate INTDualInputOr UNDCondFalseJumpTo* UNDCondTrueJumpTo*	SCLDualInputSelector SCLQuantizer3 SGLSaveImmediateToMem SGLSingleSaveToMem SCLDiffTrueJumpTo* SCLDiffFalseJumpTo*	UNDGoBank* UNDGoNextBank* UNDGoPreviousBank* UNDJumpTo* SC1IsGreater3 SC1IsGreaterEqual3 SC1IsLower3 SC1IsLowerEqual3 INTUniToggle3 SC1UniToggle3

HW				
ADC SCLFormatData12bitsAsym SCLFormatData12bitsSym SGLADCInit* ANYReadADCSample* SGLSelectADCMux* SGLSelectADCMux3* VCTMuxSwitchAdcSample VCTMuxSwitchAdcSample3	**IO Interface** SGLIOIfaceInit*	**PWM** SC1BriscPwmSymOutput3 VCTBriscPwmSymOutput3 SGLPWMInit* UNDStartPwmTimer* UNDStartMainTimer* VCTBriscSymPwmGen3 VCTSavePwmIO* SC1SavePwmIO*	**Excitation** SGLExcitationInit* UNDExcitationTicSelect*	**Serial Link** SGLTransmitFSL INTTransmitFSLA INTTransmitFSLB UNDSerialInit*
Sofware timer SGLSoftTimerInit* UNDStartSoftTimer* UNDWaitSynchro* UNDWaitSynchroDelayed*	**SPI** SCLReadSPI INTWriteSPIA INTWriteSPIB INTReadSPI			**RIO Access** ANYLoadIO3 SGLSaveIO3 SGLSaveIO*

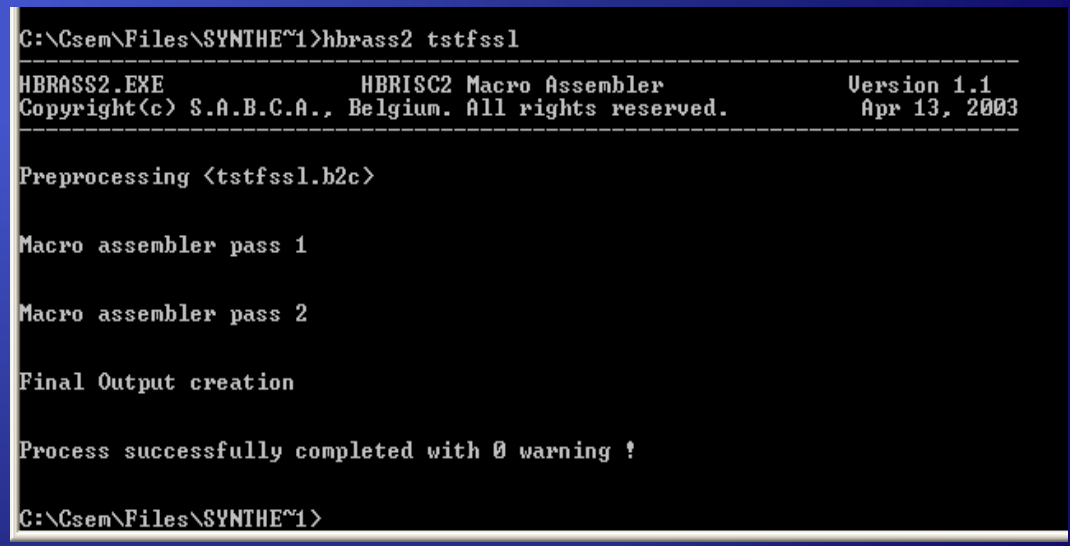
The macro library

- The macros are classified by category.
- The macros respect macro assembler coding standards.
- The macros follow a template, giving them an homogenous aspect, increasing their readability.
- The 3 first letters of the macro give information concerning their applicability (gives the units where they can be executed).

```
;SCLAdd
;Description:                General Scalar Function
;- Compute the addition of two floating registers (a or b or
      ab).
;-------------------------------------------------------------------------
;Equivalent code:                                        reference
      number
;
;    Unit.routnorm = normalise(Unit.InReg1 + Unit.InReg2);1
;      Unit.OutReg = Unit.routnorm                        ;2
;-------------------------------------------------------------------------
;Work. Units:          a, b or ab
;-------------------------------------------------------------------------
;History:
;  1.0                              By MC - Jun 19, 2001
;                                   Initial revision
;-------------------------------------------------------------------------
.MAC SCLAdd Unit InReg1 InReg2 OutReg

fadd Unit InReg1 InReg2          nop          ;1
fnorm Unit routadd               nop          ;1
Mov Unit routnorm OutReg         nop          ;2

.ENDMAC

;-------------------------------------------------------------------------
;Max Cycles:          6
;-------------------------------------------------------------------------
;End of scla.bh
```

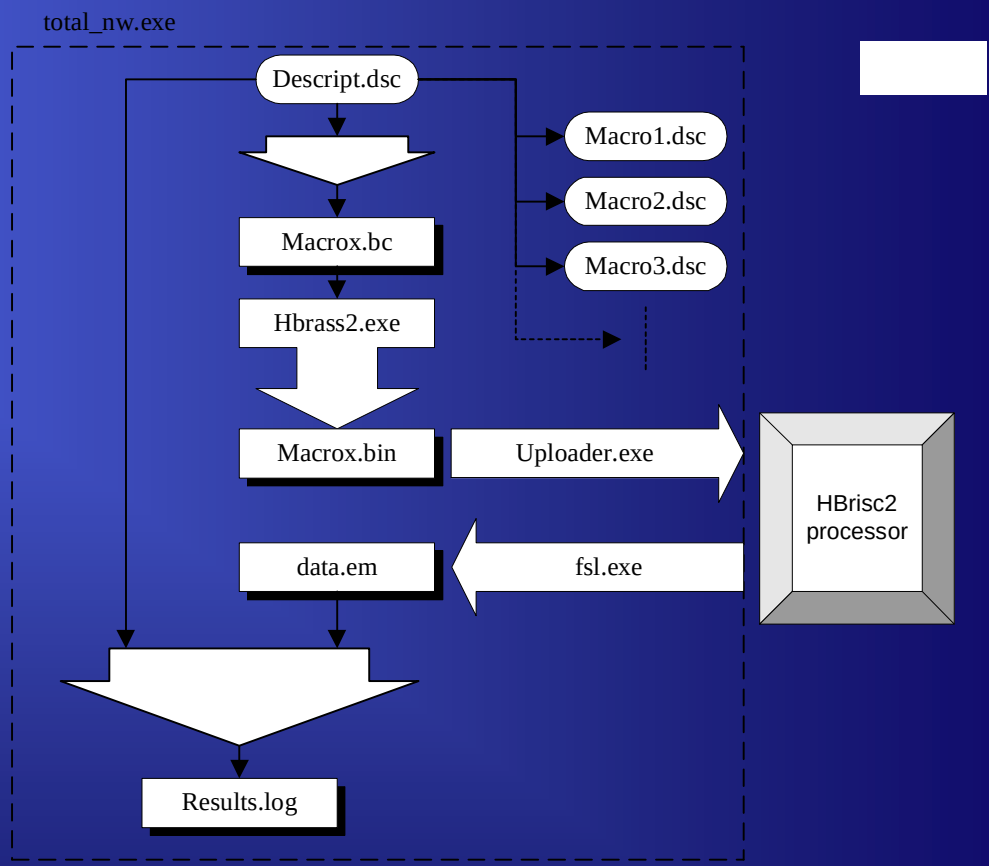
The macro assembler

- This macro assembler supports:
  - the mnemonics for the full hbrisc2 instruction set
  - nestable include files
  - nestable macros
  - forward reference resolution (two pass assembler)
  - hbrisc2 data format oriented directives
  - double instructions per line to fit to hbrisc2 memory access philosophy
  - checking for specific hbrisc2 restrictions
  - minimal size limitations due to dynamic memory allocation
  - mathematical expressions evaluation



The unit test tool

- Total_nw is a tool that allows a large set of macros to be tested in a sequential and automatic way, over a large set of input values.
- It can generate variations over the input variables (registers) or contants (constant values passed directly).
- The variations can be linear (most of the cases) or not (step, sinus, …)
- A single macro can be tested by several test cases.
- A hierarchical structure ensures unicity of all input files.
- All results are stored together in a single file, easing the post analysis work.

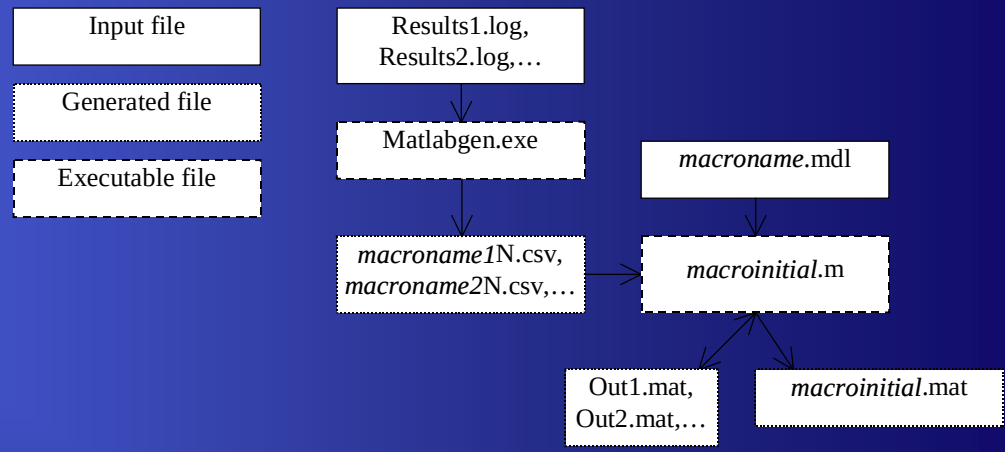


Validation of the macros

- All macros except interface macros have been tested with total_nw.
- Interface macros have been simplified as much as possible (simple copy of one register to another for instance) to limit the risk of error insertion.
- Interface macros have been tested at integration level.
- Special care has been taken to ensure that all macros branches were covered by the different test cases.
- The macros results have been compared to those of an equivalent C++ routine.
- The acceptable error limit has been set to 1% relative for floating point results, and the maximum between 1 and 1% for integer values. Exceptions have been set for filters which were tested with sinusoidal curves, and for which numerous passages by zero and long accumulation of data generated temporarily high relative errors. Special error studies have been done in theses cases.

Validation of the Simulink library

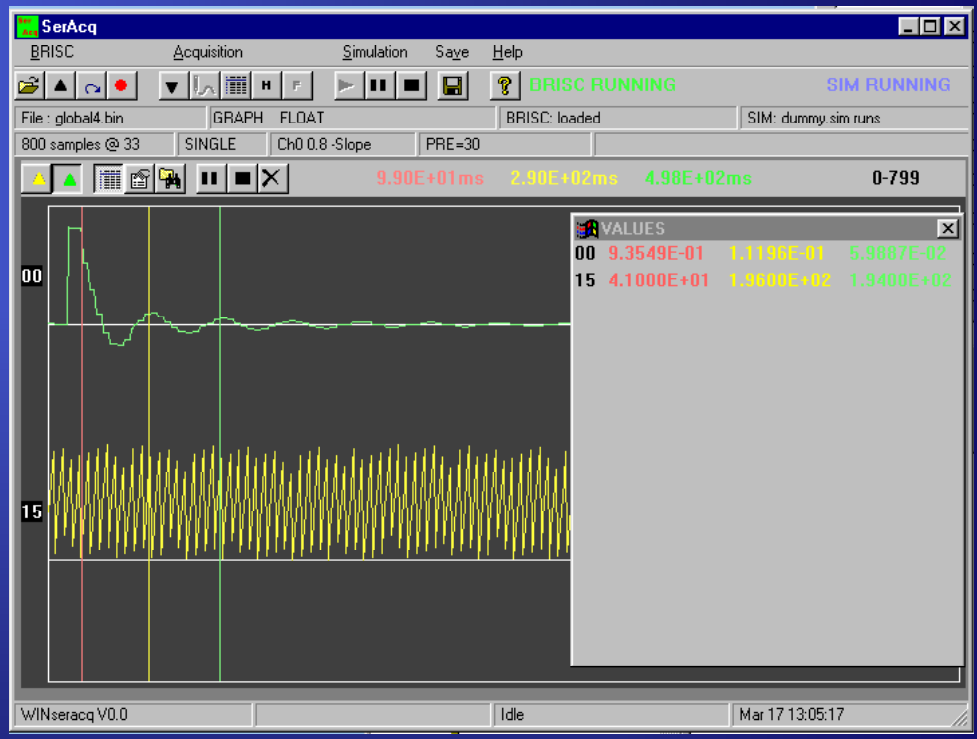
- Simulink blocks have been tested with the test vectors from macros unit tests.
- The results have been compared to the results obtained for the macros.
- A small utility has been created to extract data from the result files, and create csv files readable by Matlab.
- Simulink models have been created that allow rapid testing of the different Simulink blocks.



All the tests of both libraries were successful.

The Win-Seracq Tool

- **The Win-Seracq tool is a Win32 application that allows:**
  - To upload a binary code into the HBRISC2 processor and start the processor on it.
  - To send scripted commands to the processor through the SPI link, in order to generate a given behaviour.
  - To acquire the data transmitted by the processor through the HSSL link.
  - To display the acquired data the same way as an oscilloscope: conditional triggering, signal measurement, …
  - To store the acquired data on disk for future analysis.



IMCM Demonstration Application

- The actuator performances are tested when mounted on a test rig simulating its actual mechanical load when installed in the launcher stage. The main load characteristics are:
  - Constant effort of 4490 N
  - Inertia of 1562 Kg

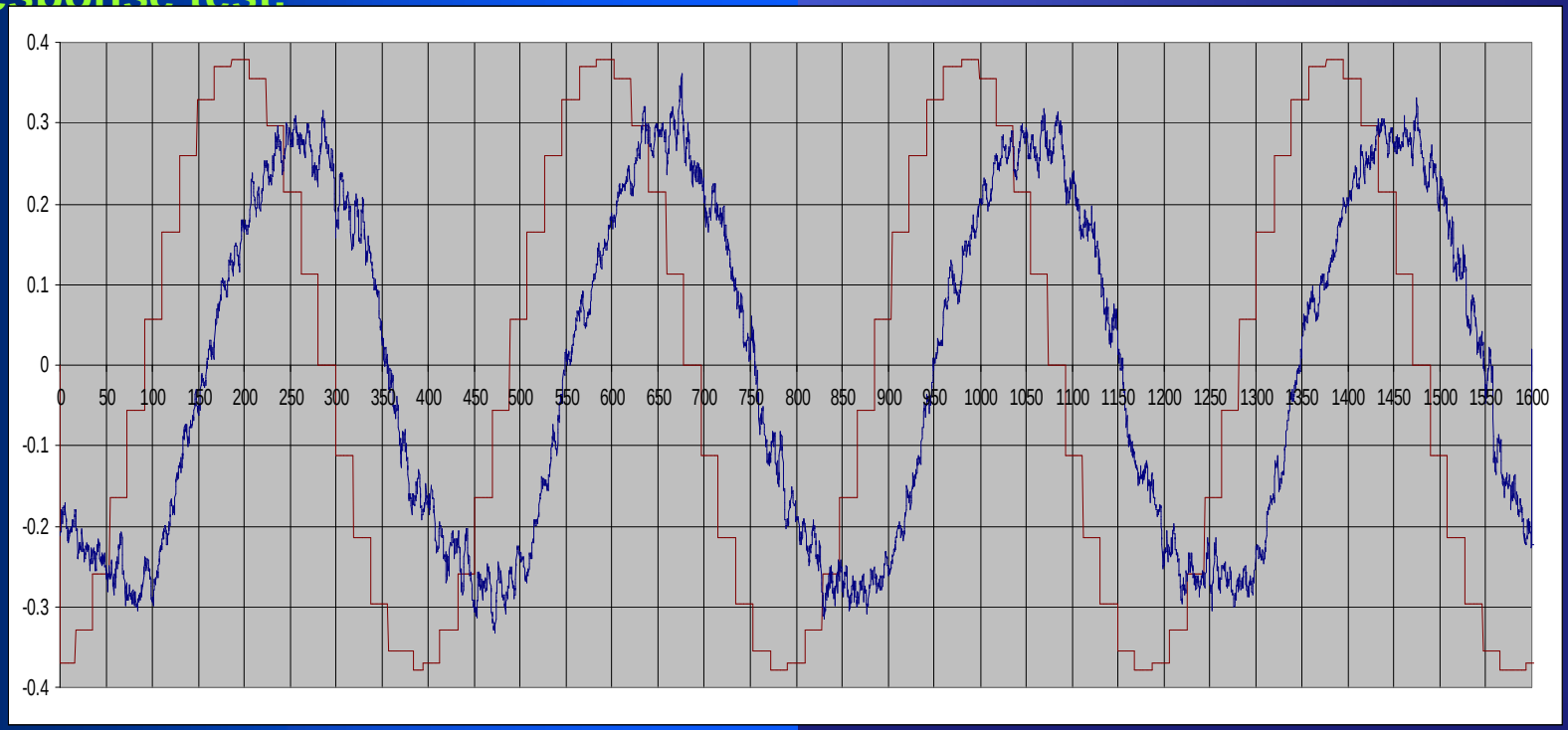
IMCM Demonstration Application

- The following table give the principal achieved performances under load

	Measurement	Criteria
Saturation speed [mm/s]	38.64 mm/s	≥ 34 mm/s
Time to reach 34 mm/s [ms]	66.6 mm/s	< 80 ms
Frequency response @ 2.5 Hz [°; dB]	-30 °; 2.35 dB	> -45 °; < 7.5 dB
Frequency response @ 3.7 Hz [°; dB]	-47 °; 5.2 dB	> -90 °; < 7.5 dB

IMCM Demonstration Application

- The following figure illustrates the actuator position order and achieved position, as measured by the embedded lvdt, during a 2.5 Hz frequency response test.

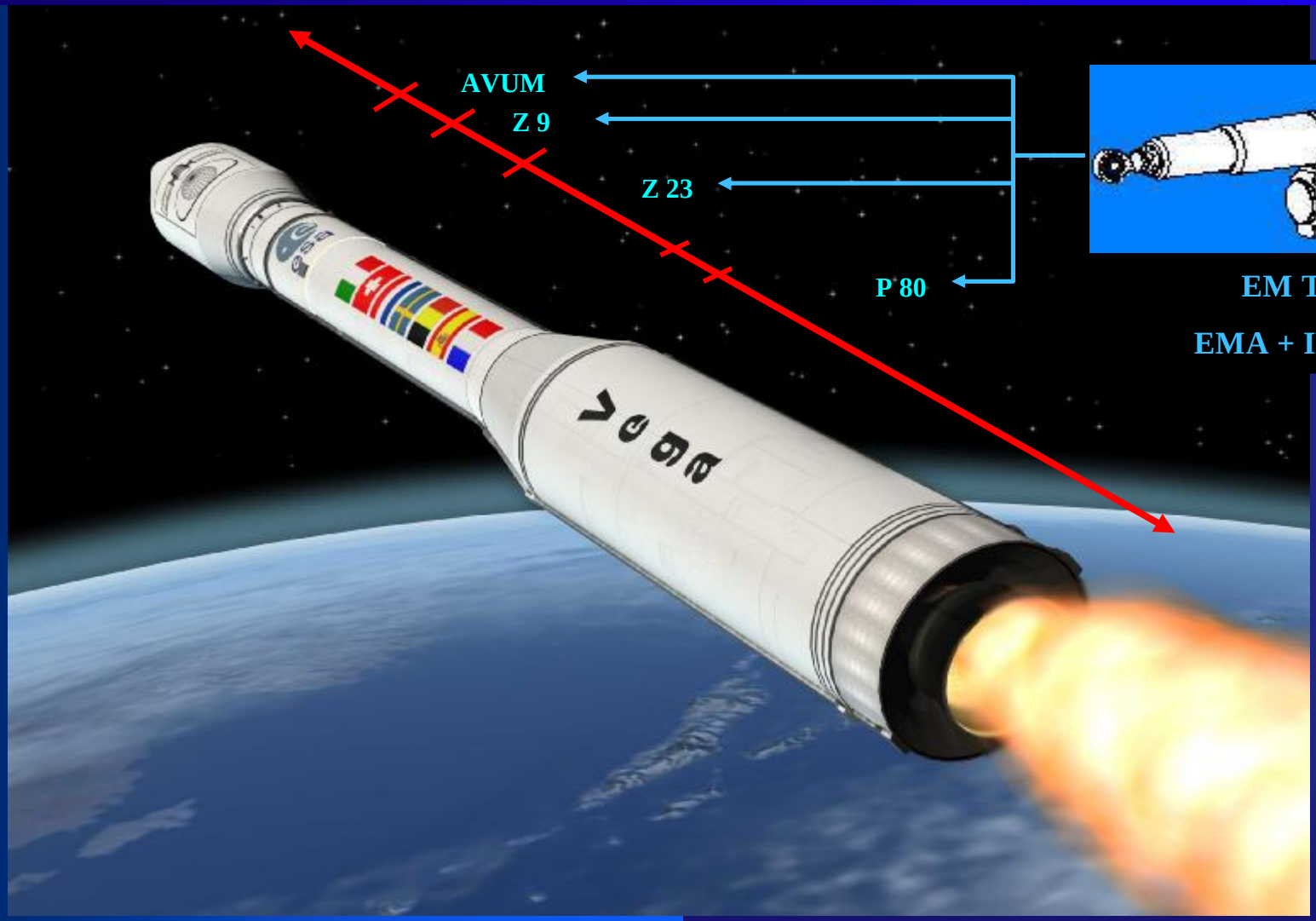


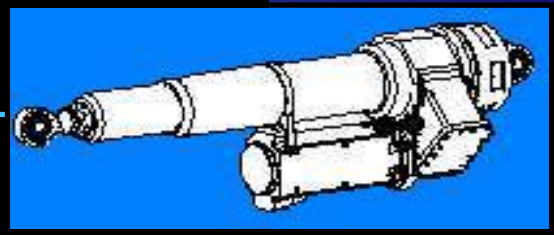
Time in ms; Amplitudes in mm

HBRISC2 Perspectives

- *Current and coming applications*
  - *VEGA*
  - *Tip-tilt mechanism*
  - *GSTP3 HPEA*
  - *GSTP4 Reaction Sphere*
- *Development environment*
  - *Towards HBRISC2 Application+ Tool*

VEGA TVC IPDU



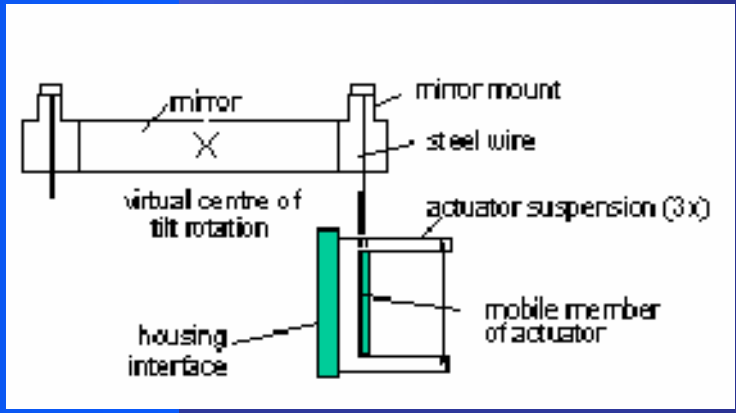


EM TVC is

EMA + IPDU +BS

Tip-Tilt Mechanism, a CSEM Application

- *3-axis Tip-Tilt Mechanism demonstrator supporting a mirror and based on CSEM innovative flexure structure technology (FLEXTEC)*



- *3-axis control is implemented with one single HBRISC-2*
- *First SABCA activity/experience as "HBRISC-2 provider"*
  - *Evaluate fine pointing potential of HBRISC-2*
  - *Have external feed-back on HBRISC-2 circuit and developing environment*

GSTP3 HPEA & GSTP4 Reaction Sphere, a CSEM Application

- ***GSTP-3 HPEA: High Power Electrical Actuation for Solid Rocket Boosters***
  - ***Goal: validate technologies that could be used in a Solid Rocket Booster high Power Electro Mechanical Thrust Vector Control (EMTVC)***
  - ***Premise of ARIANE 2010***
  - ***HBRISC2 is part of the 50kW motor drive demonstrator:***
    - *Direct drive*
    - ***Power sharing** between axes*
    - ***Redundant controller***
- ***GSTP-4 Reaction Sphere – under negotiation –***
  - ***Support to BRISC***
  - ***Expertise in asynchronous motor available (VFC application)***
  - ***Possible companion activities: see next slides on development tools potential upgrades***

Towards HBRISC2 *Application*+ Tool

- **First real applications (VEGA TVC & Tip-Tilt) have highlighted that a continuous evolution of HBRISC2 tools is required:**
  - **To cover wider application ranges**
  - **To develop faster**
- **Potential evolutions include:**
  - **HBRISC2 debugger**
  - **Synthesizer+ (“single-model/multi-binary” architecture)**
  - **Enhanced Win32-Seracq and HSSL-SPI I/F**

IMCM Conclusion

- HBRISC2 and its development tools, Synthesizer and libraries were successfully developed and validated in the frame of this GSTP2 IMCM project
- The circuit and its tools are already part of several applications:
  - The flight critical VEGA launcher TVC, on all four stages
  - Complex demonstration applications with CSEM
  - More contacts are on-going
- Evolution towards even more efficient development tools are already planned