A Hardware Refinement Method in SystemC

Jérôme Cornet
THALES Communications France
Institut Supérieur d'Informatique, de Modélisation et de leurs Applications

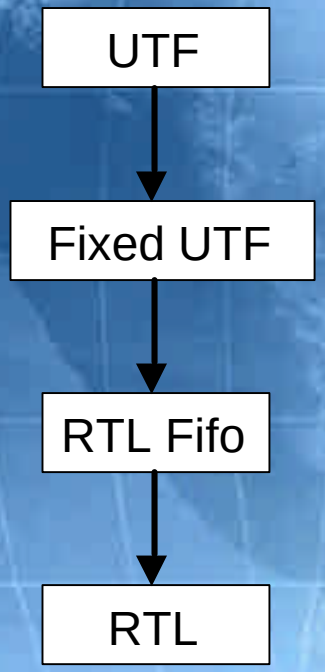
Agenda

- Motivations
- Starting Point
- Refining to Fixed-point UTF
- Hardware Refinement
- Synthesis Results
- Problems
- Conclusion
- Q & A

THALES

ISIMA

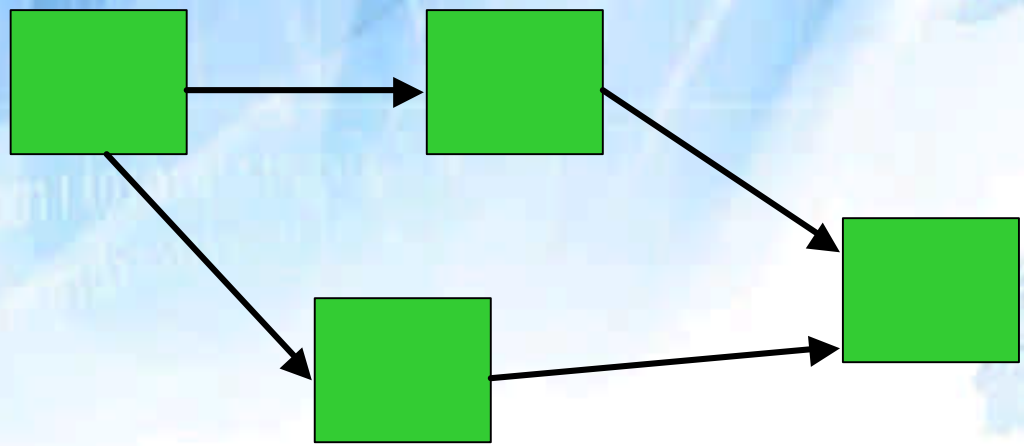
Motivations



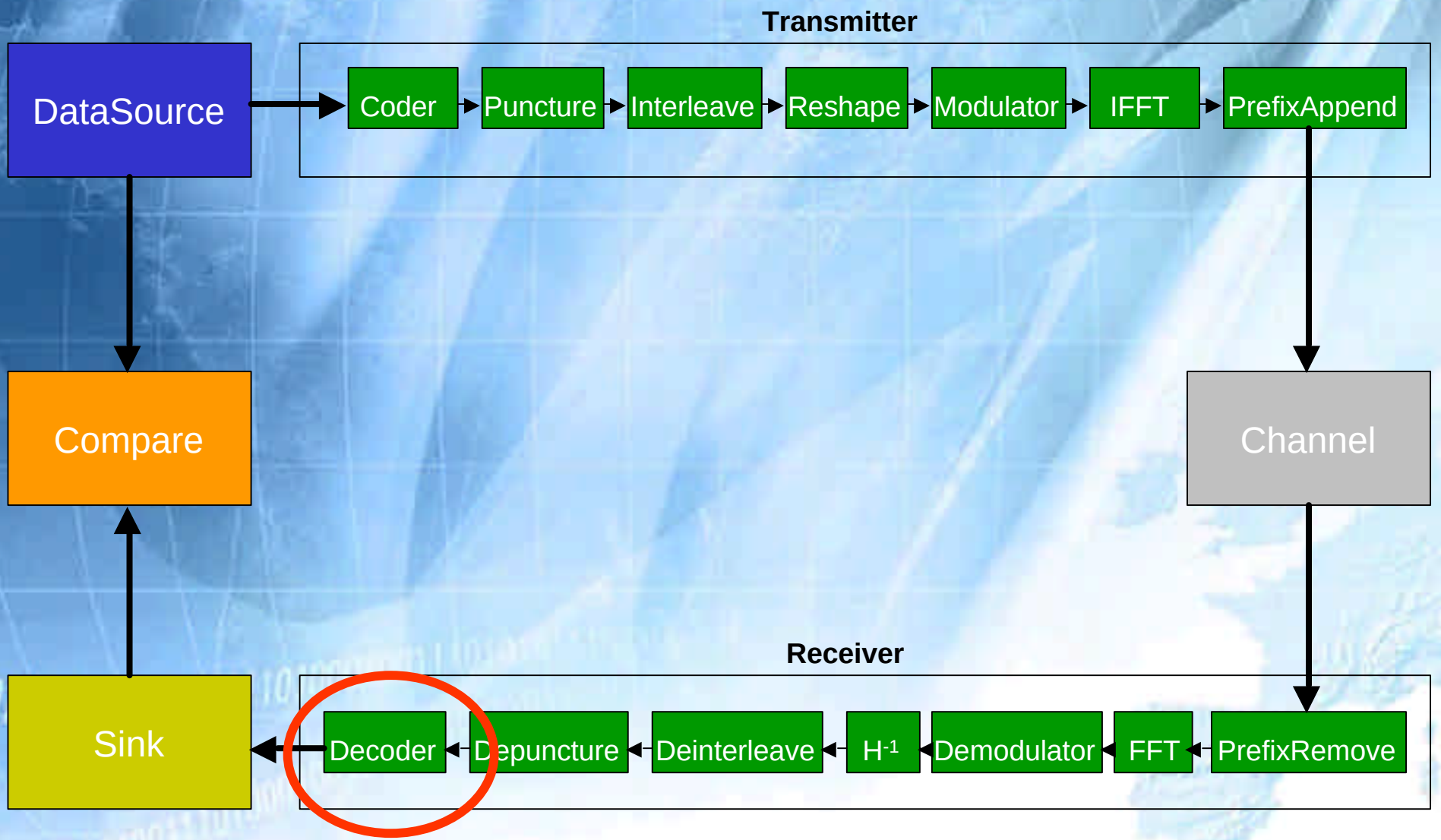
- Exploring and evaluating a descending design approach
  - **Defining new abstraction level(s)**
  - **Transitions between different models**
  - **Designing a hardware refinement methodology**
- Detailing Hardware Refinement on a concrete application
  - **Digital Signal Processing, Datastream Design**
  - **Viterbi Decoder (rate 1/2)**
  - **Implementation on FPGA Target (Xilinx VirtexE 400)**

Starting Point : UTF Model

- Split between different functions
- Usual communications by fifos (sc_fifo)
- Complex datatypes or Static Data Flow on simple types
  - **C++ custom type “ Matrix ”**
  - **double + SDF**

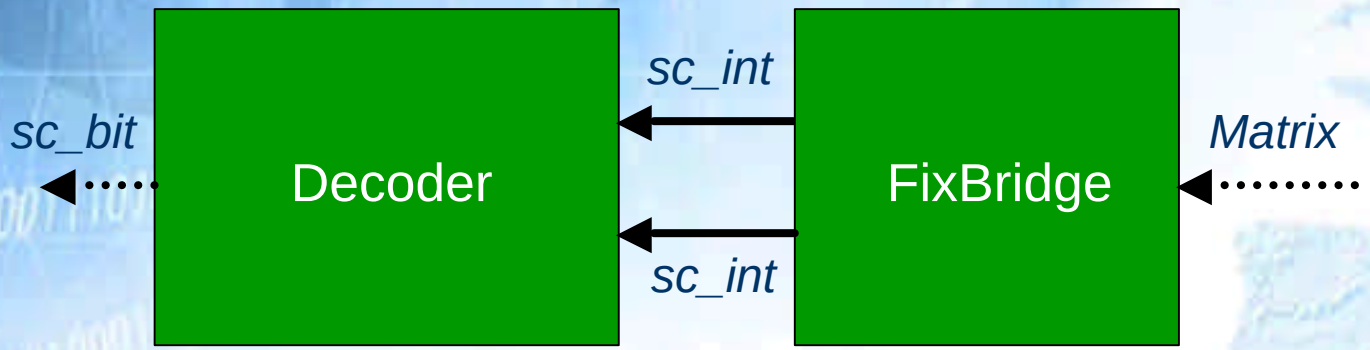


802.16a Modem Example



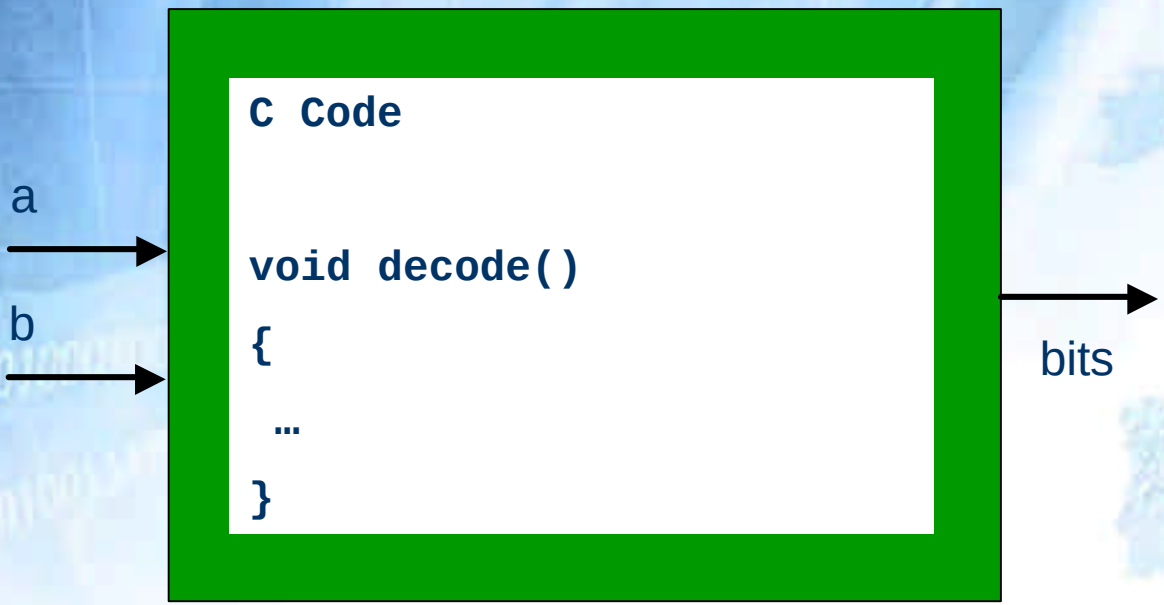
Refining to Fixed UTF

- Traditional fixed-point transition
  - **… but in hardware context**
  - **Use of sc_fixed as an early help ® fixed-point datatypes**
  - **Final state : sc_int, sc_uint® fixed-precision datatypes**
- Adapter module: “ FixBridge ”
- **<u>Validated:</u>** performances in fixed point/fixed precision



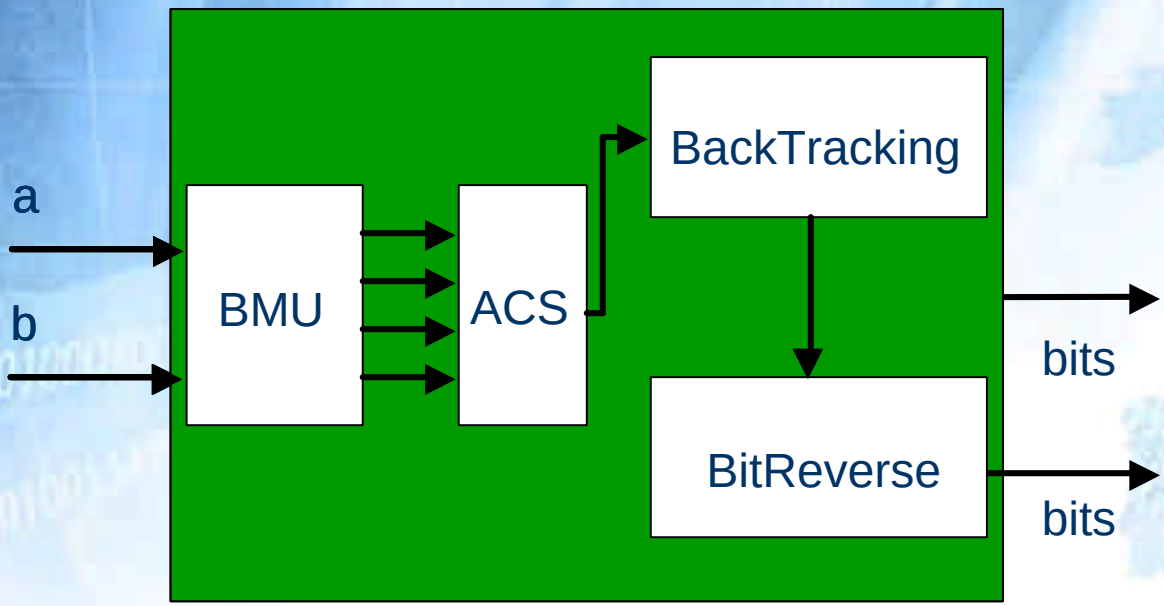
Hardware Refinement

- Starting Point:
  - **Same datatypes as in RT Level for I/O**
  - **sc_fifos**
  - **C Code describing Viterbi Decoding**



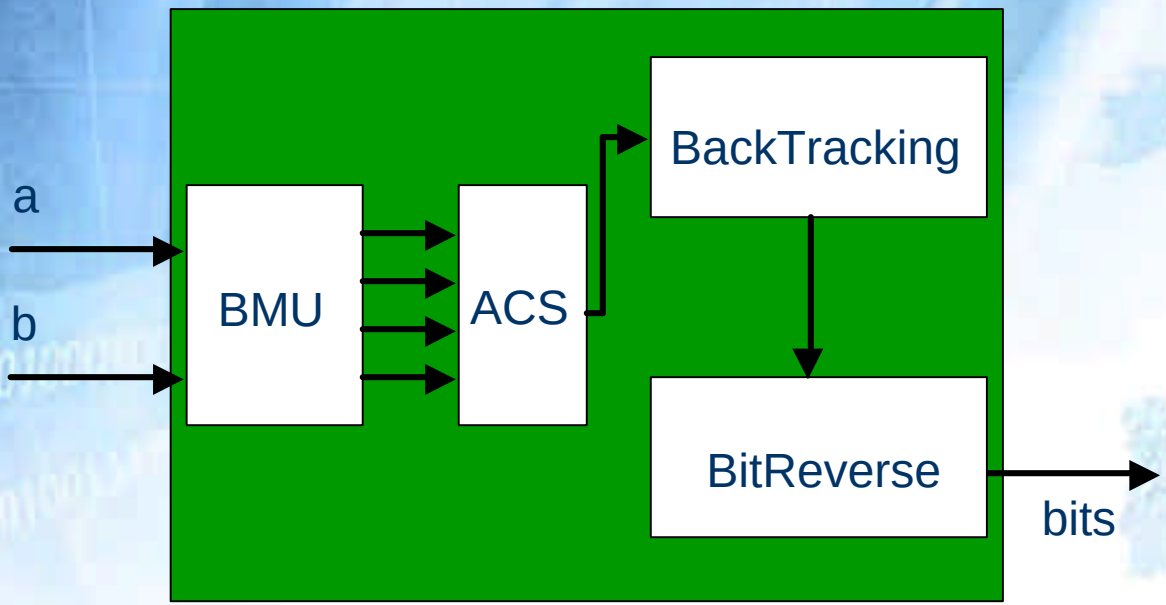
First level of structural refinement

- Spliting processing into different sub-modules
- Refining communications
- Static Data Flow Processing



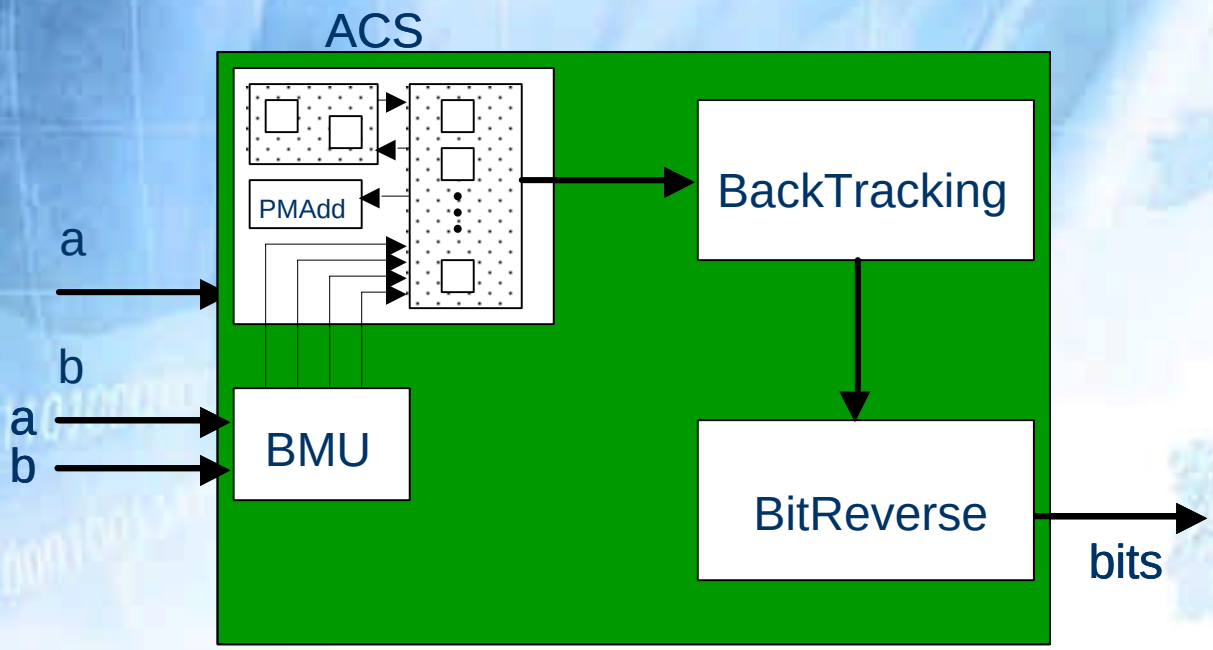
Second Step

- Doing “token by token” processing
- Some submodules still need SDF processing
- **Validated:** Split between submodules, storage needs



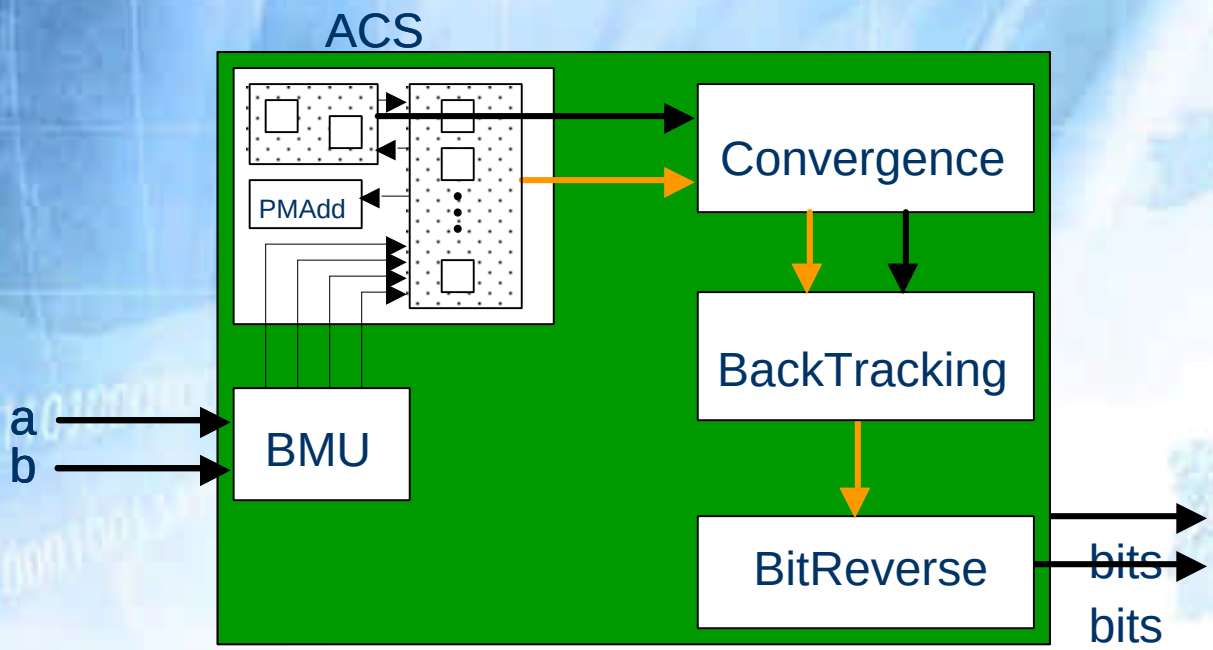
Further levels of structural refinement

- For “token by token” processing
- Operators pipelining, optimisations
- **Validated:** Internal operators “ RTL ” description



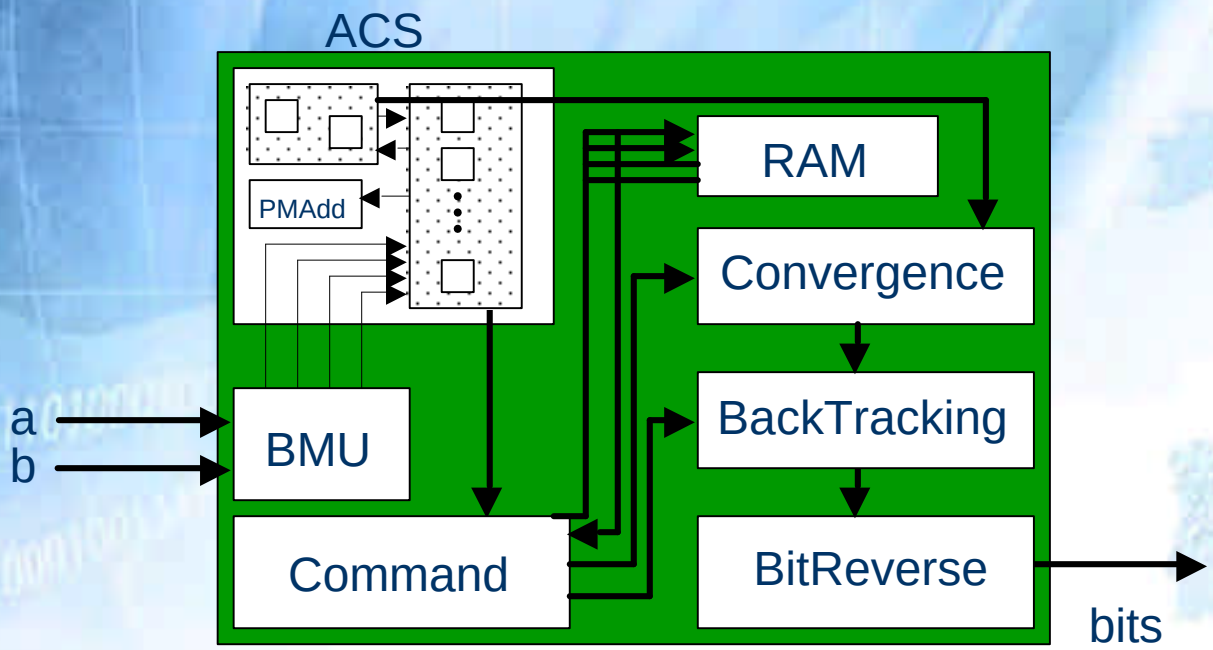
Reducing storage need

- Algorithm changes
- Sequencing within operators
- Storage described by arrays
- **Validated:** performances



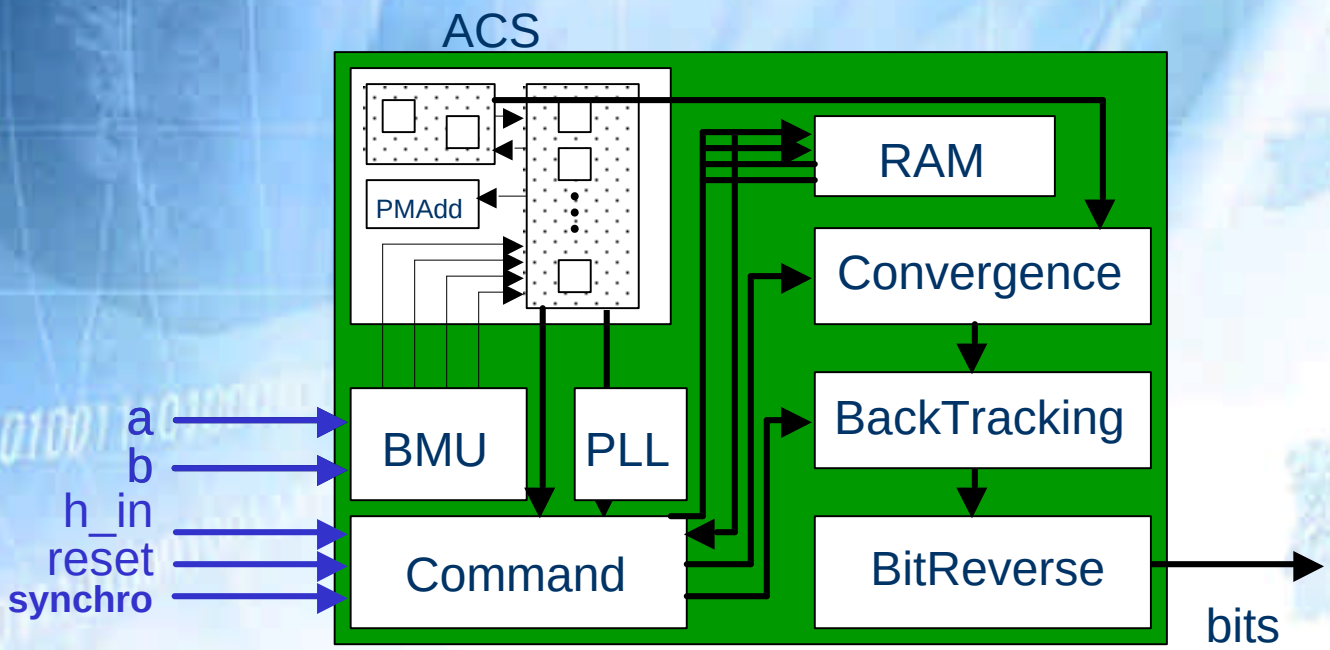
Storage refinement

- High-level model of FPGA RAM (fifos i/o)
- Sequencing block
- Data multiplexing
- **Validated:** Architecture with RAM, *RTL Fifo level*



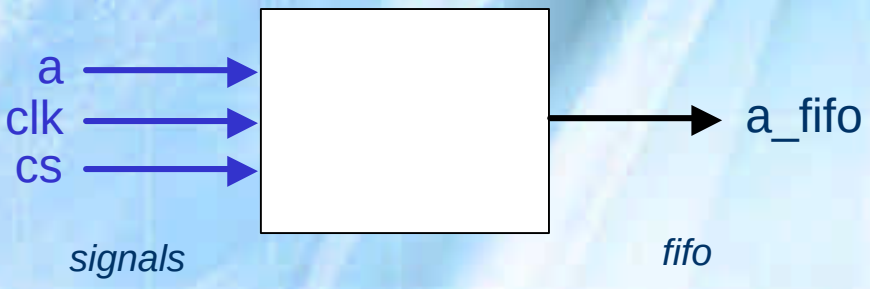
Sequencing

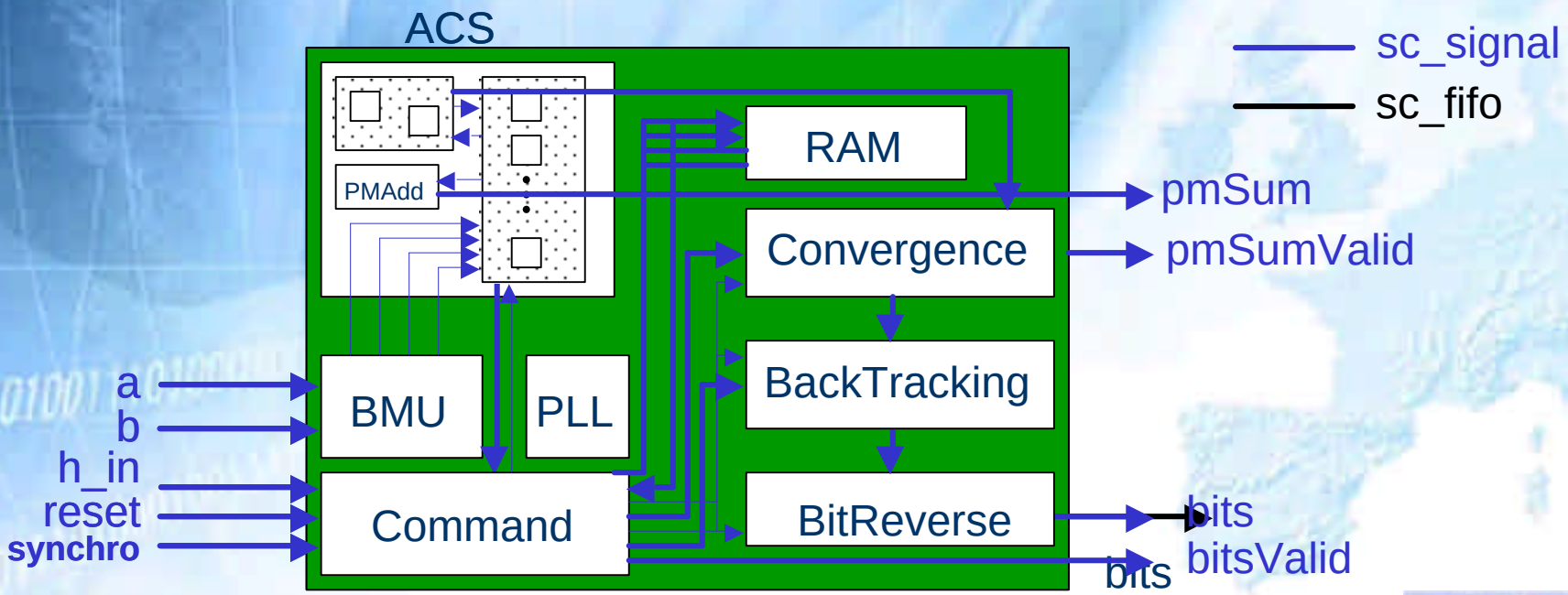
- Converting fifos into signals
- Clock Component: PLL
- Clock/Reset/Validation inputs
- RTL Adapter module



Sequencing : tools

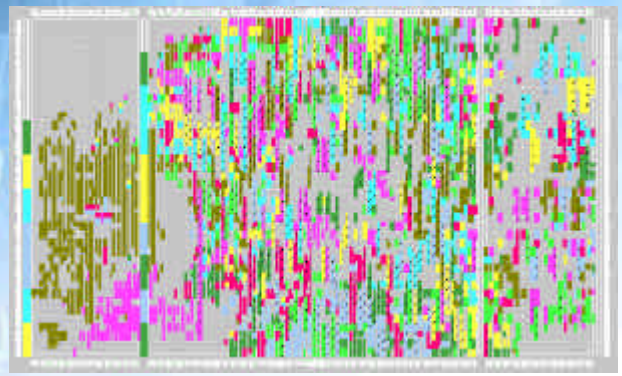
- Mixing RTL Fifo and RTL
- Wrapper
- State machines
- **Validated:** Temporal side

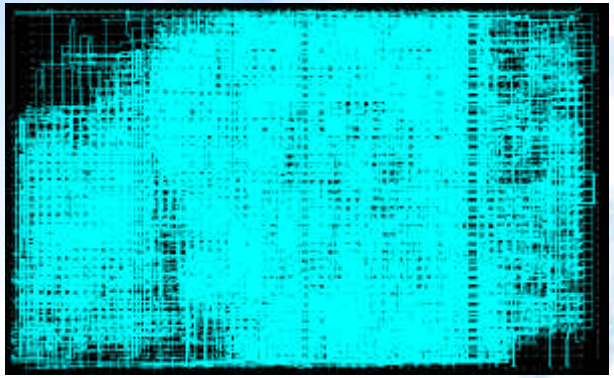




Synthesis Results

- Use of Nepsys SystemC to VHDL Compiler (Prosilog)
- Post place & route validation by SystemC/VHDL cosimulation (Nepsys/ModelSim)
- Performances compared with traditionnal VHDL :
  - **+ 1,6 % Area (2480 slices on Xilinx VirtexE 400)**
  - **About 48 MHz instead of 60 MHz (25 MHz initially)**





Problems

- No RAM/PLL models available in SystemC
  - **Hand-written models**
  - **Direct core generation in SystemC?**
  - **VHDL Models Translation to SystemC?**
  - **Cosimulation?**
- Type choices in SystemC RTL
  - **bool/sc_logic, sc_uint/sc_lv**
- RTL Adapter
  - **Non synchronous on active edge of the clock**

Conclusion

- Refinement method from UTF to RTL with SystemC
  - **Convenient choice to apply the method**
  - **Ability to integrate various levels of abstraction (UTF/RTL in the global system, RTL Fifo/RTL in the Decoder)**
- Incremental validation
  - **One error type at each step**
  - **Easy to build functionally correct RTL**
  - **Testbench reuse**
- Method well suited to data-driven designs

Q & A

Contacts:

Jérôme Cornet (Jerome.Cornet@isima.fr)

Anne-Marie Fouilliart (Anne-Marie.Fouilliart@fr.thalesgroup.com)

Pierre Wodey (Pierre.Wodey@isima.fr)