On the effectiveness of Assertion-Based Verification in an Industrial Context

L.Pierre, F.Pancher, R.Suescun, J.Quévremont

TIMA Laboratory, Grenoble, France

Dolphin Integration, Meylan, France

Thales Communications & Security, Gennevilliers, France

Introduction

- What can be the benefits of Assertion-Based Verification - and of synthesizable assertion checkers - in an industrial context?

- Outline:
  - What is Assertion-Based Verification?
  - A few words about the synthesizable checkers
  - Formalization and verification of requirements for an actual *HDLC controller IP*
    - Return of experience, benefits of the formalization steps, properties violations
    - Characteristics of the generated hardware assertion checkers

Introduction - ABV ?

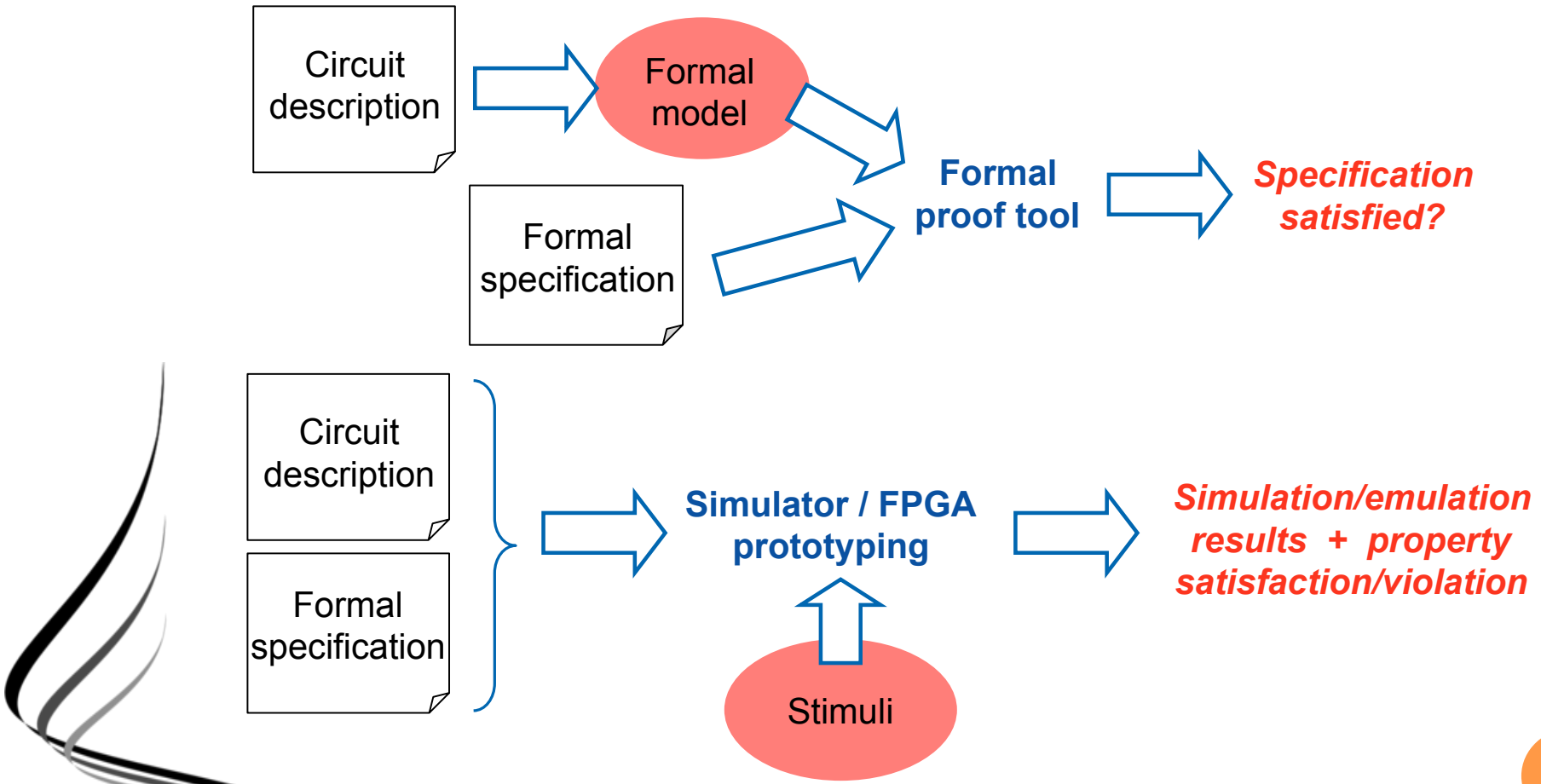
- *Assertion*: statement about the *intended behaviour* or a *requirement* of the design
  - Temporal logics: CTL, LTL,…
  - Specification languages: SVA (IEEE Std 1800), PSL (IEEE Std 1850)
- *Assertion-Based Verification*: does the design obey these temporal assertions?
  - Static analysis (model checking)
  - Dynamic verification (during simulation)

INTRODUCTION - ABV ?

- *Assertion*: statement about the *intended behaviour* or a *requirement* of the design
  - Temporal logics: CTL, LTL,…
  - Specification languages: SVA (IEEE Std 1800), PSL (IEEE Std 1850) ←
- *Assertion-Based Verification*: does the design obey these temporal assertions?
  - Static analysis (model checking)
  - Dynamic verification (during simulation) ←

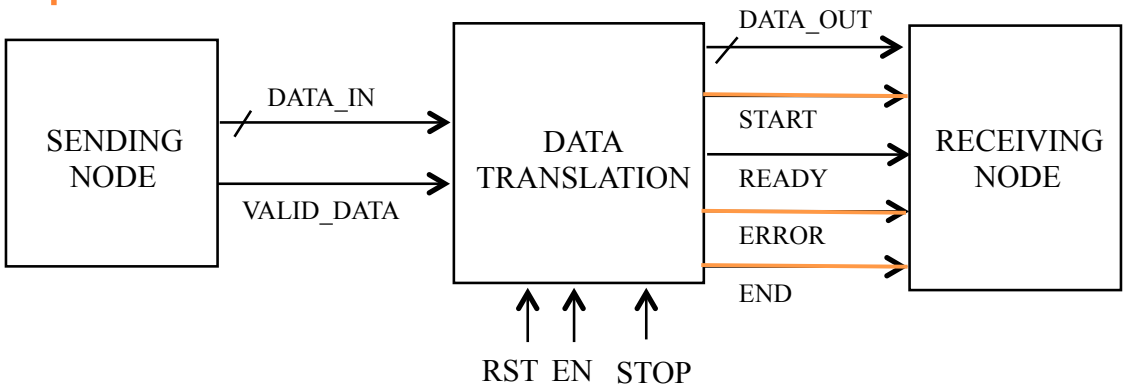
Introduction - ABV ?

- Static vs dynamic verification

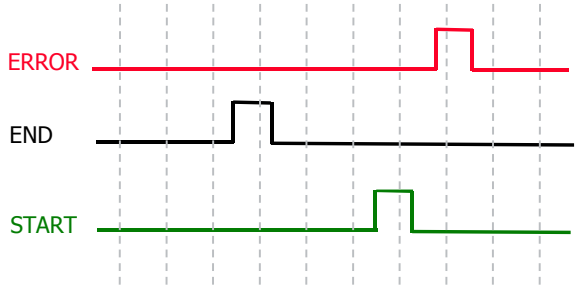


ASSERTION CHECKERS

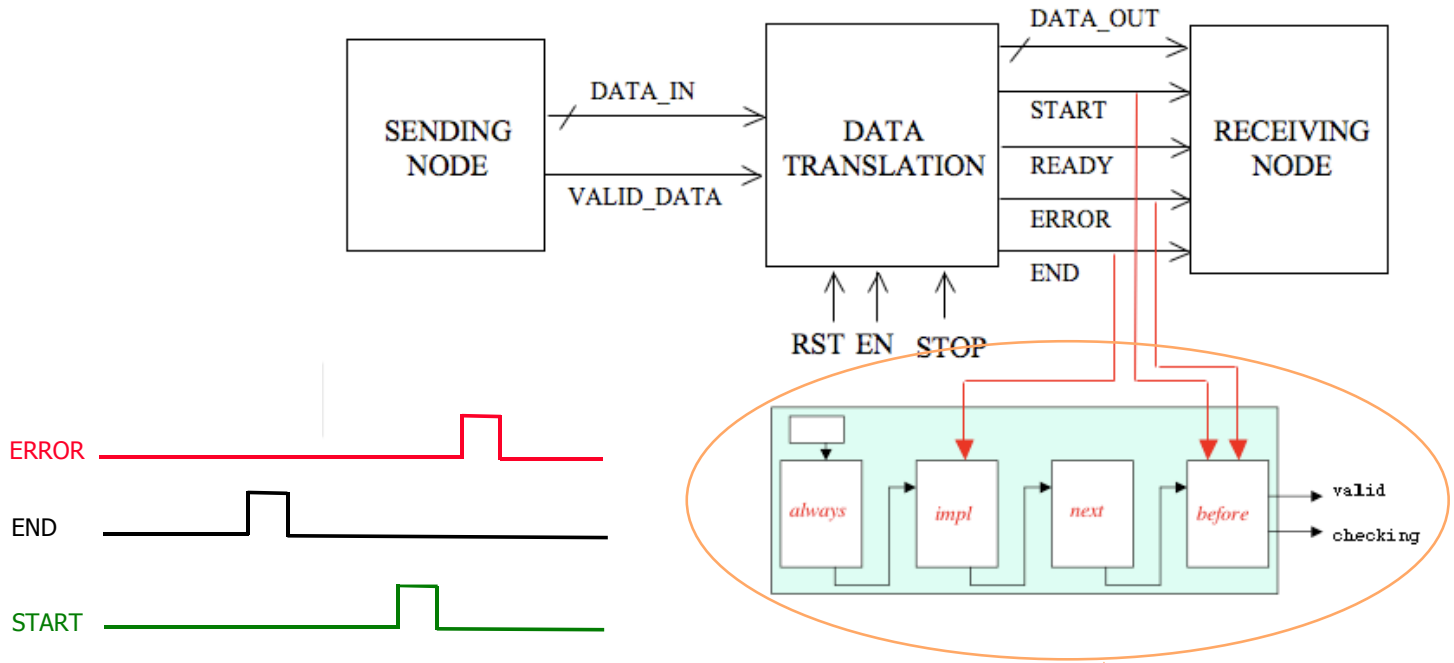
- PSL associated with VHDL RTL descriptions
  - Logic and temporal properties on the signals of the design
  - Example:



default clock = (posedge clk);
always(END ->
next (START before ERROR))



ASSERTION CHECKERS



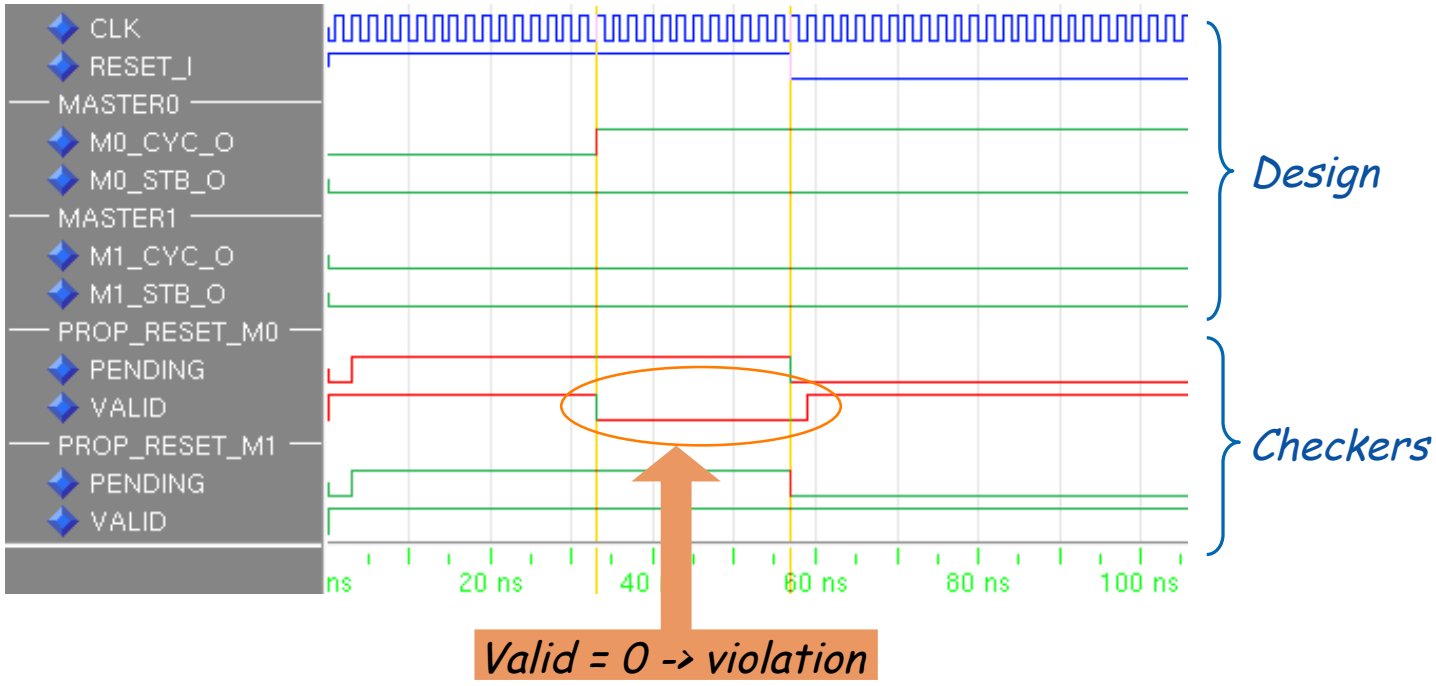
always(END -> next (START before ERROR))

Checker

PSL assertion

Assertion checkers

- Simulation of the design under verification connected to the assertion checkers:



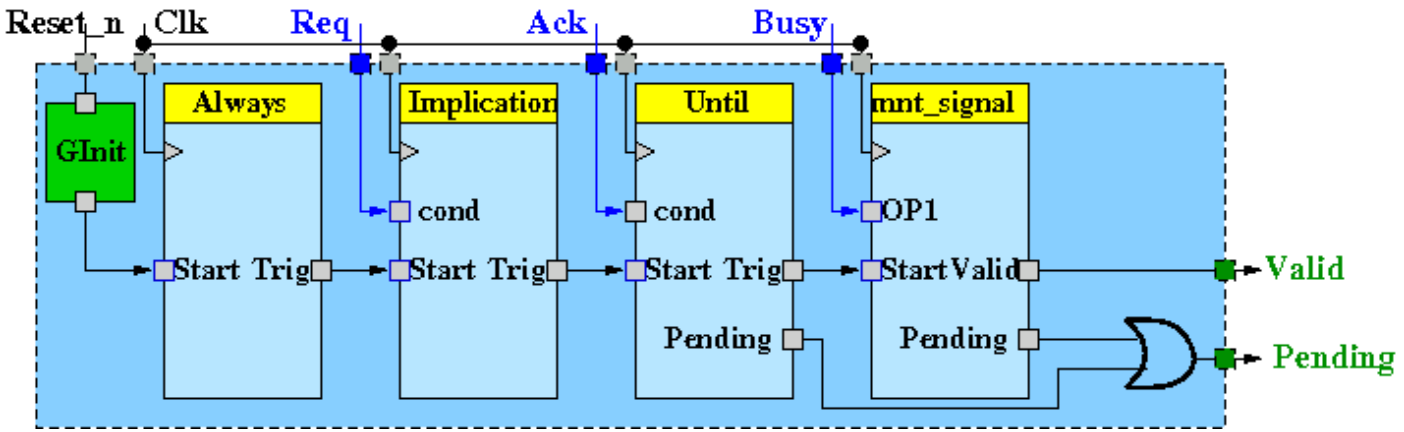
- FPGA prototyping with the checkers

ASSERTION CHECKERS

Formally proven

- Compositional construction
  - Example:

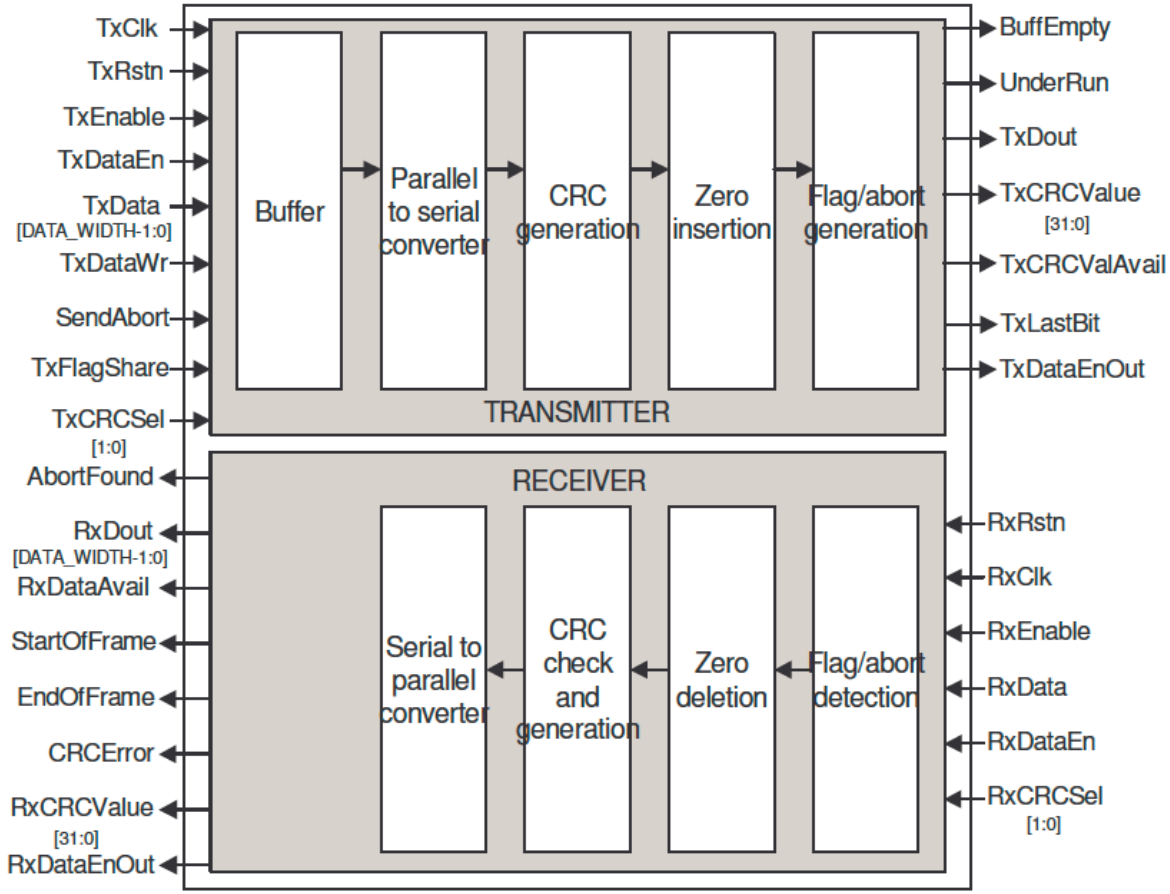
always (Req -> (Busy until! Ack))



http://tima.imag.fr/vds/Horus/
http://www.dolphin.fr/medal/products/smash/options/smash_SVA.php

Case Study

- HDLC (High-level Datalink Control) controller IP

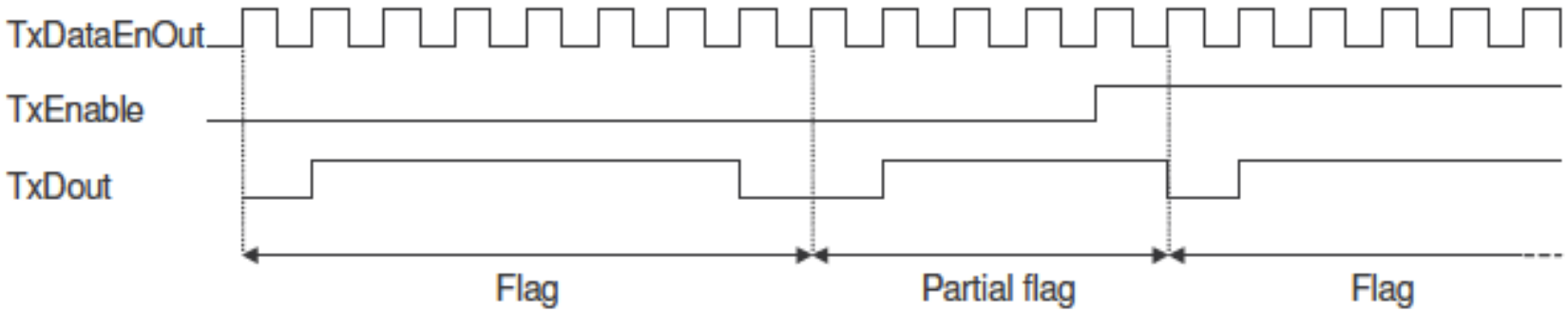


Case Study

- Documents
  - VHDL source files and user guide
  - Requirements specification
- About 20 requirements
  - Nominal behaviour
    - Start and end of frames - Example: *The receiver shall activate the signal EndOfFrame when it is outputting the last byte of a frame*
    - Transparency
  - Idle mode - Example: *While disabled (idle mode), the transmitter shall send flags or partial flags*
  - Abort
  - ...

Case Study

- Example: while disabled (idle mode), the transmitter shall send flags or partial flags




```
property HDLC_240:
  always({!TXLASTBIT && !TXDATAWR; TXLASTBIT && !TXDATAWR}
       |=> {{!TXDOUT && !TXENABLE; (TXDOUT && !TXENABLE)[*6];
             !TXDOUT && !TXENABLE}[*];
            {{TXENABLE} |
             {!TXDOUT; (TXDOUT && !TXENABLE)[*0..6]; TXENABLE}}});
```

Case Study

- Informal vs formal requirements
  - The signal AbortFound shall be set high when the reception of an abort sequence is detected. *When exactly?*
  - When SendAbort gets active, the transmitter shall abort the frame by transmitting at least seven consecutive ’1’ bits. It shall be followed by flags before the transmission of new words. *Violations, due to an implicit hypothesis about the sampling clock for output data.*
  - If disabled while sending a character, the transmitter shall finish sending that character, followed by CRC and close flag before coming back to idle mode. *Violations, due to an implicit hypothesis (no reception of data while disabled).*

Case Study

- Informal vs formal requirements
  - The transmitter shall activate the UnderRun signal and end the frame automatically if no new valid byte has been written in the input buffer on the seventh rising edge of the transmitter clock following the emission of the buffer empty signal. *Violations, due to an implicit hypothesis (latency for the emission of UnderRun allowed).*
  - The receiver shall activate the signal EndOfFrame when it is outputting the last byte of a frame. *Violations, but the unexpected behaviour is worked around by the including components.*

Case Study

- Characteristics of the generated monitors

Properties	Area overhead	Leakage power overhead	Detection of violations
Transmitter			
HDLC200	3.81 %	3.61 %	✘
HDLC240	3.40 %	3.27 %	
HDLC250(1)	14.70 %	13.55 %	✘
HDLC250(2)	16.35 %	14.96 %	✘
HDLC300	2.19 %	1.92 %	✘
HDLC160	7.21 %	6.30 %	
Receiver			
HDLC210	3.41 %	2.98 %	
HDLC260	3.58 %	3.02 %	
HDLC310	7.48 %	6.70 %	
HDLC320	20.80 %	19.28 %	✘

Synthesis performed with Synopsys Design Compiler. The target cell library is Dolphin's SESAMEeHSvHD_TSMC_0.18um.

Conclusions

- Overhead - Summary:
  - Area: 80%
  - Leakage power: 72% - Dynamic power: 10%
  - No impact on frequency
- Thales's return of experience
  - More bugs detected than with classical simulation
  - Careful (natural language) expression of the requirements
  - PSL can be considered as a way to make safer designs
    - In FPGA-based prototyping, to track and detect more bugs more quickly
    - If embedded monitors are targeted: tradeoff between design criticality and resource overhead

Thanks… Questions?