# VERIFICATION OF LAST INCH ARCHITECTURES FOR CORRECT-BY-CONSTRUCTION TIMING

Calnex

John C. Eidson, et.al.

ITSF-2015


www.calnexsol.com

# Acknowledgements

Co-authors:

- Aviral Shrivastava, Arizona State University
- Hugo Andrade, National Instruments
- Patricia Derler, National Instruments
- Ya Shian Baboud, NIST Gaithersburg
- Marc Weiss, NIST Time and Frequency
- Kevin Stanton, Intel Corporation

# Overview

Calnex

- “Correct-by-construction” temporal semantics
- Reminder of an existence proof
- What will it take?
- Testbed proposal
- Conclusions

# What is meant by timing that is "correct-by-construction"?

Calnex

Designers of embedded systems, especially distributed embedded systems, should be able to design, simulate, and code generate for multiple targets with guaranteed timing!

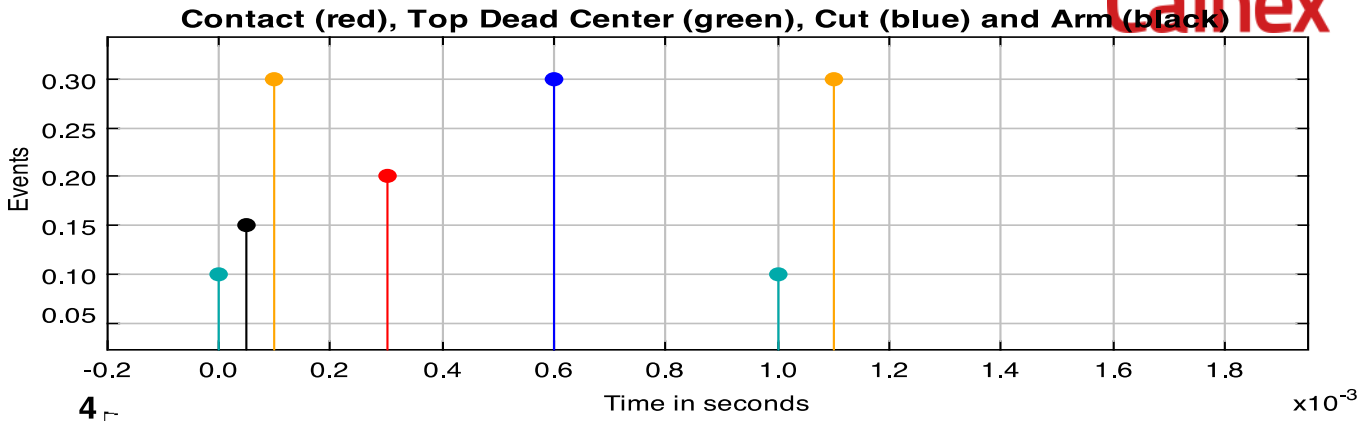
# How is timing achieved today?

Calnex

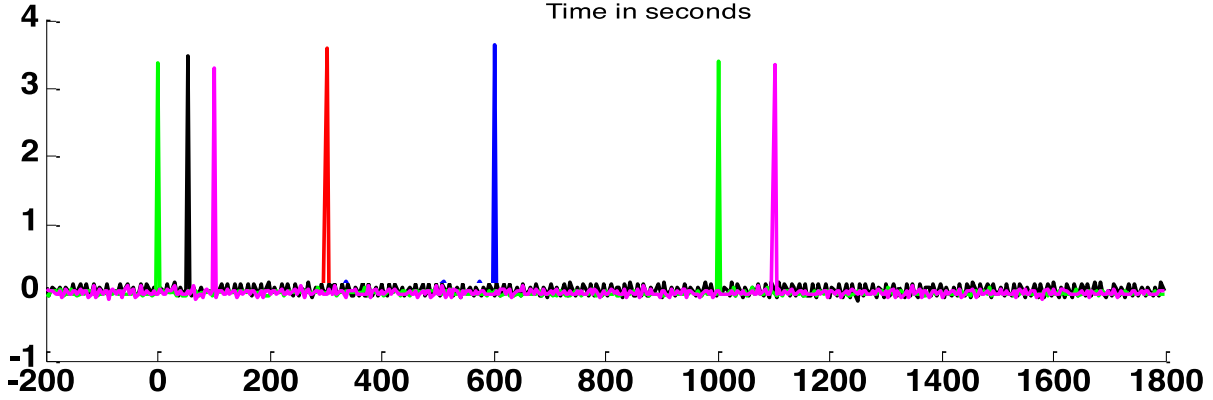
- One current method is carefully constructed bounded-WCET code plus extensive testing followed by frozen design and implementation
- All timing in hardware e.g. FPGA
- Potential “correct-by-construction” techniques:
  - Time triggered architectures e.g. PROFINET
  - PTIDES-like architectures (see next two slides)

# EXISTENCE PROOF: Renesas vs. XMOS: I/O timing (from ISPCS 2013 report on work at UC Berkeley)

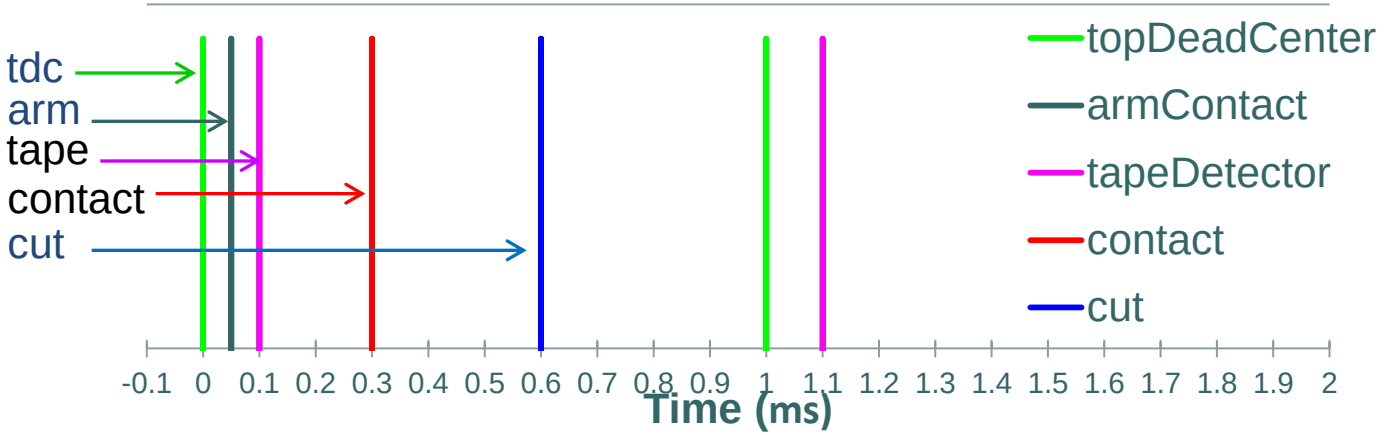
Simulation




Renesas

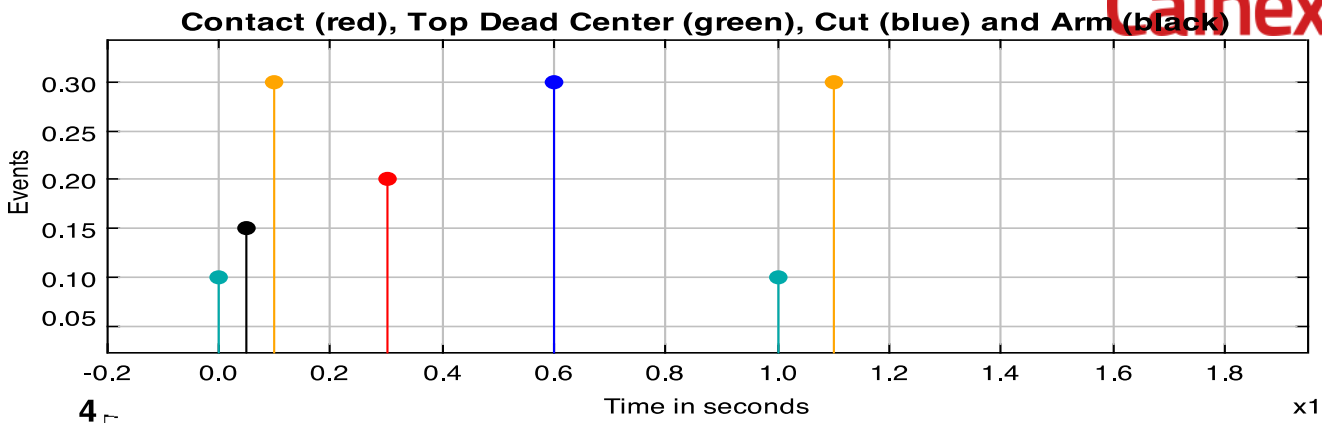



XMOS

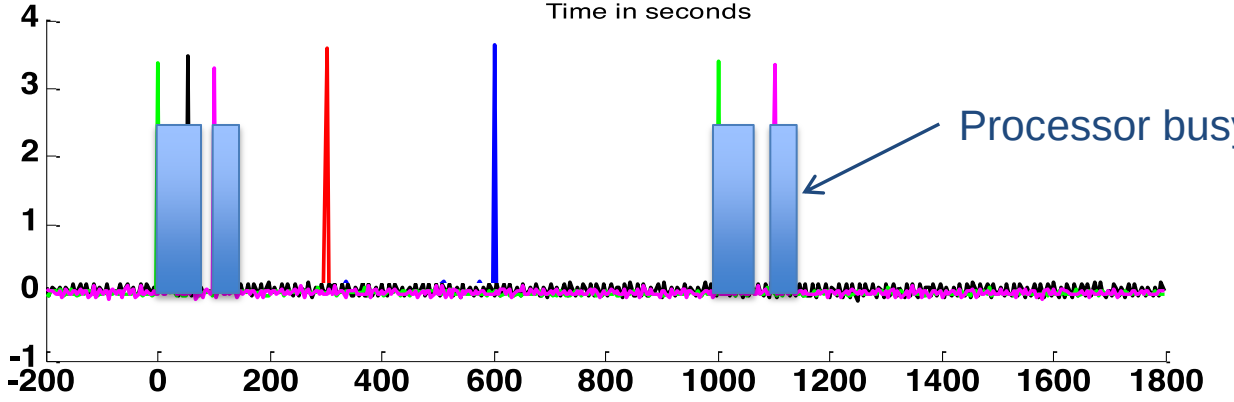


# Renesas vs. XMOS: Busy vs. Idle Time

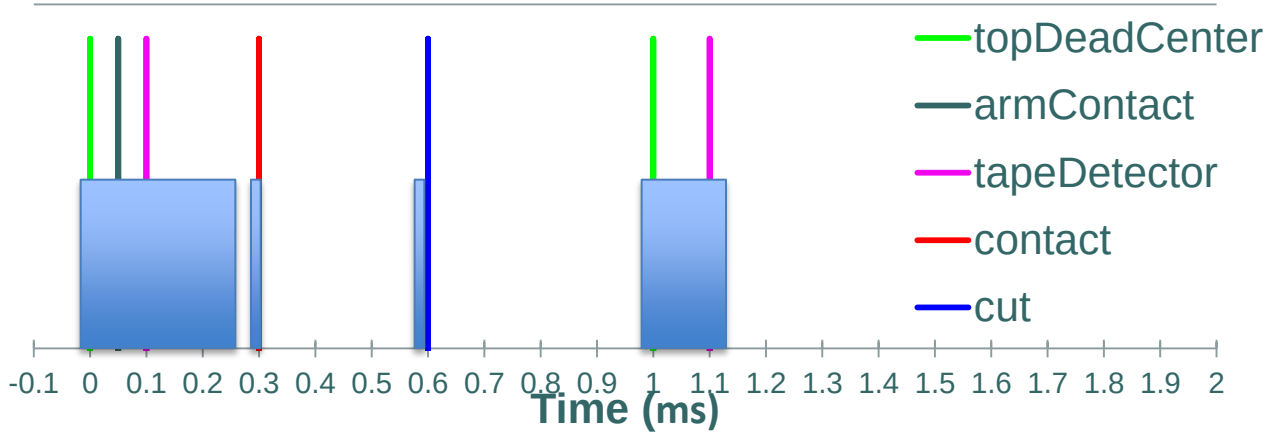
Simulation




Renesas




XMOS
(single core
execution)

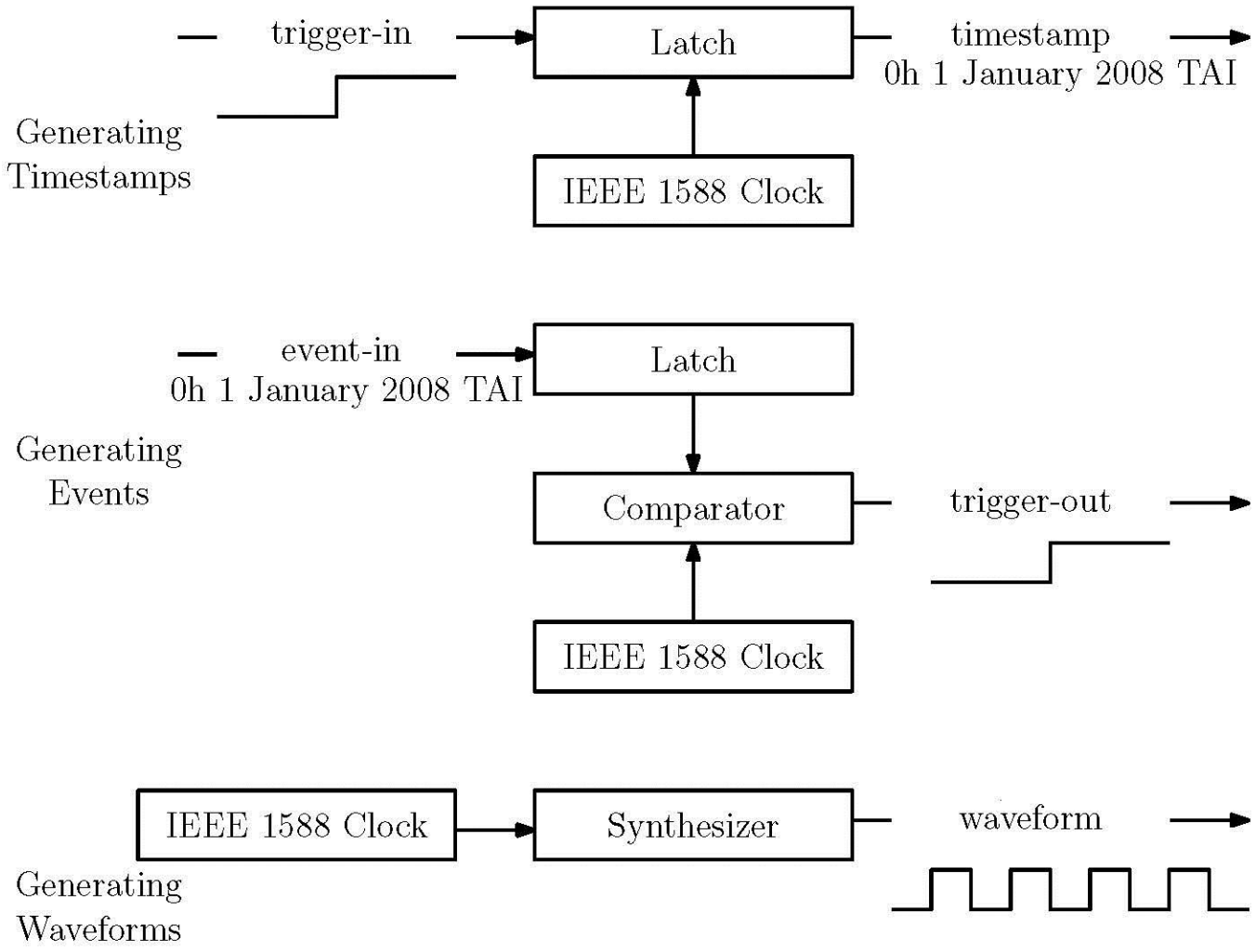


# “Correct-by-construction” restrictions

Calnex

- Requires bounded temporal density of I/O and network traffic (closed world)
- Requires bounded WCET and network latency
- Timing enforcement is done in hardware timing primitives.
- Code timing is not strict except for bounded WCET.

# Timing Primitives



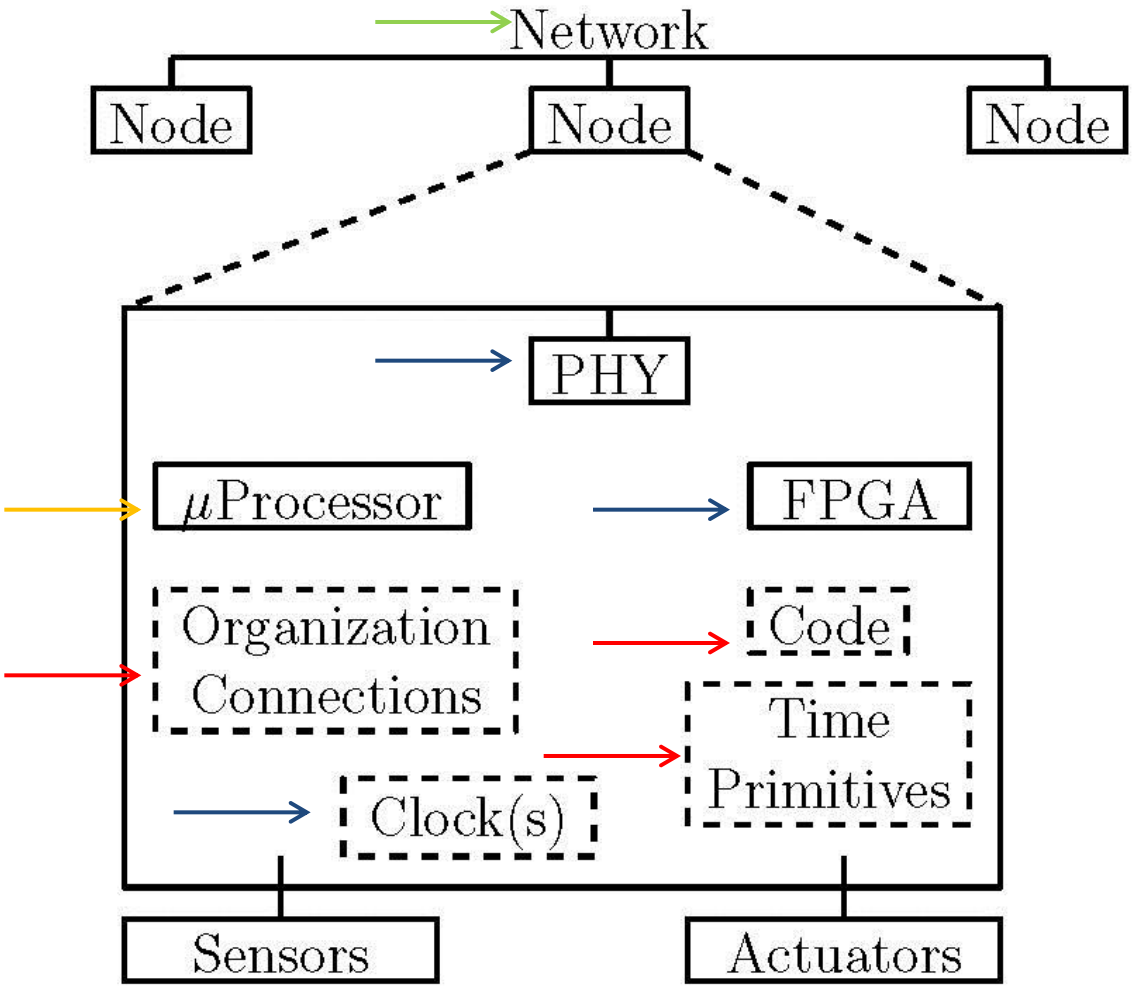
# To achieve a correct-by-construction design environment, we must sort out this mess!

~OK →

Improving →

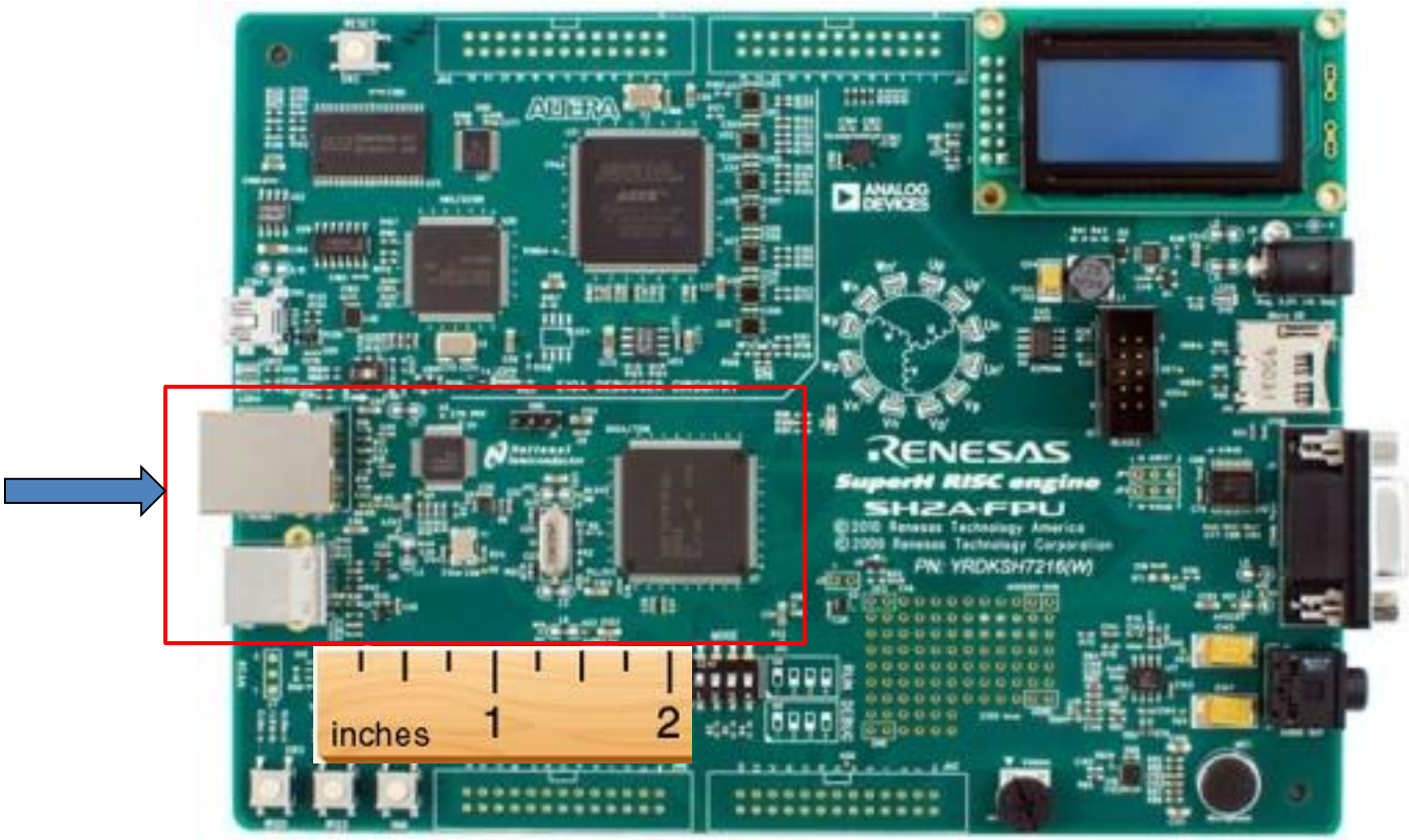
Glimmer of hope →

Lots to be done →



# And on a real board this stuff looks like this

This is the stuff of the previous slide



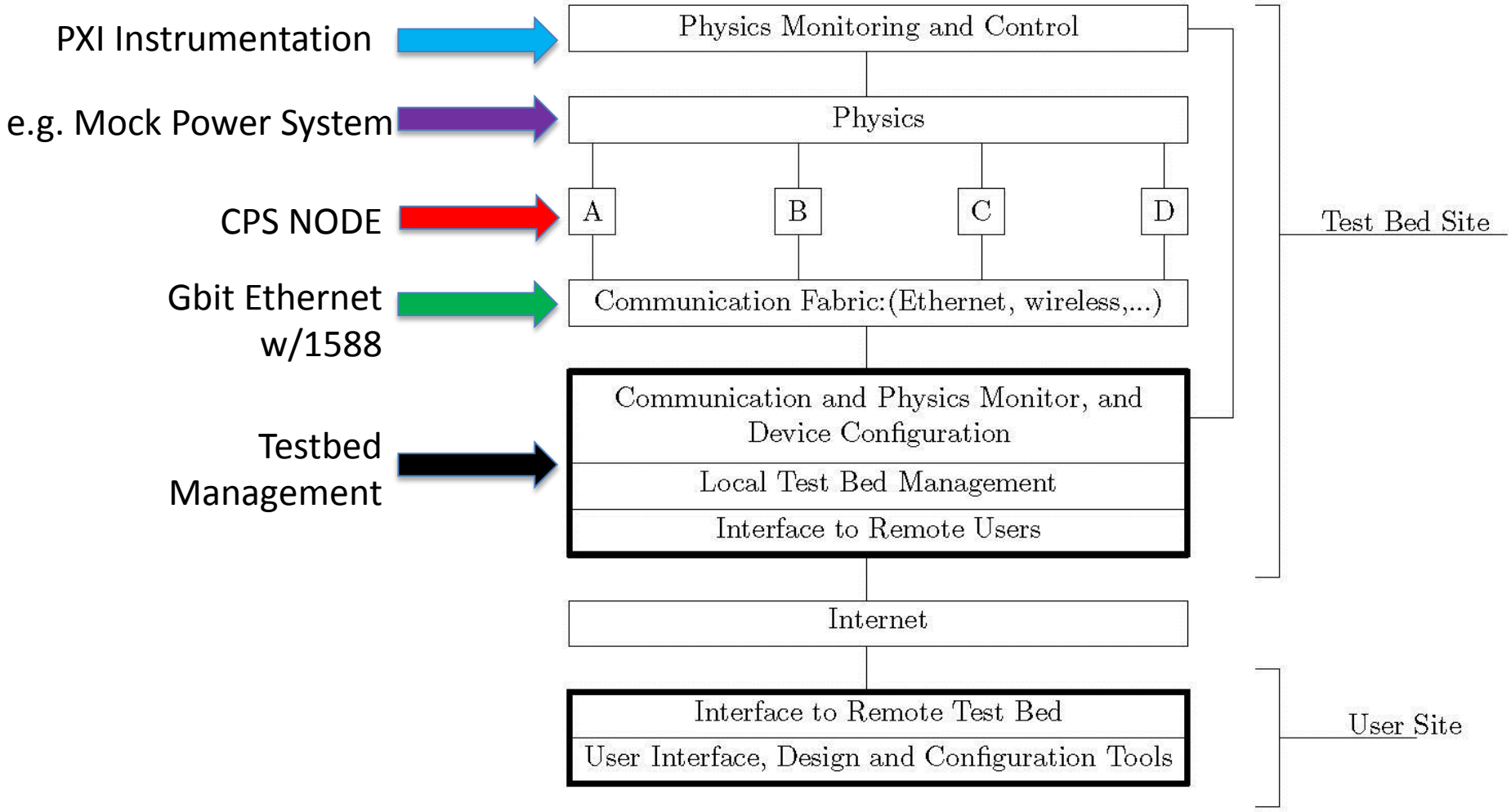
# The Testbed

Calnex

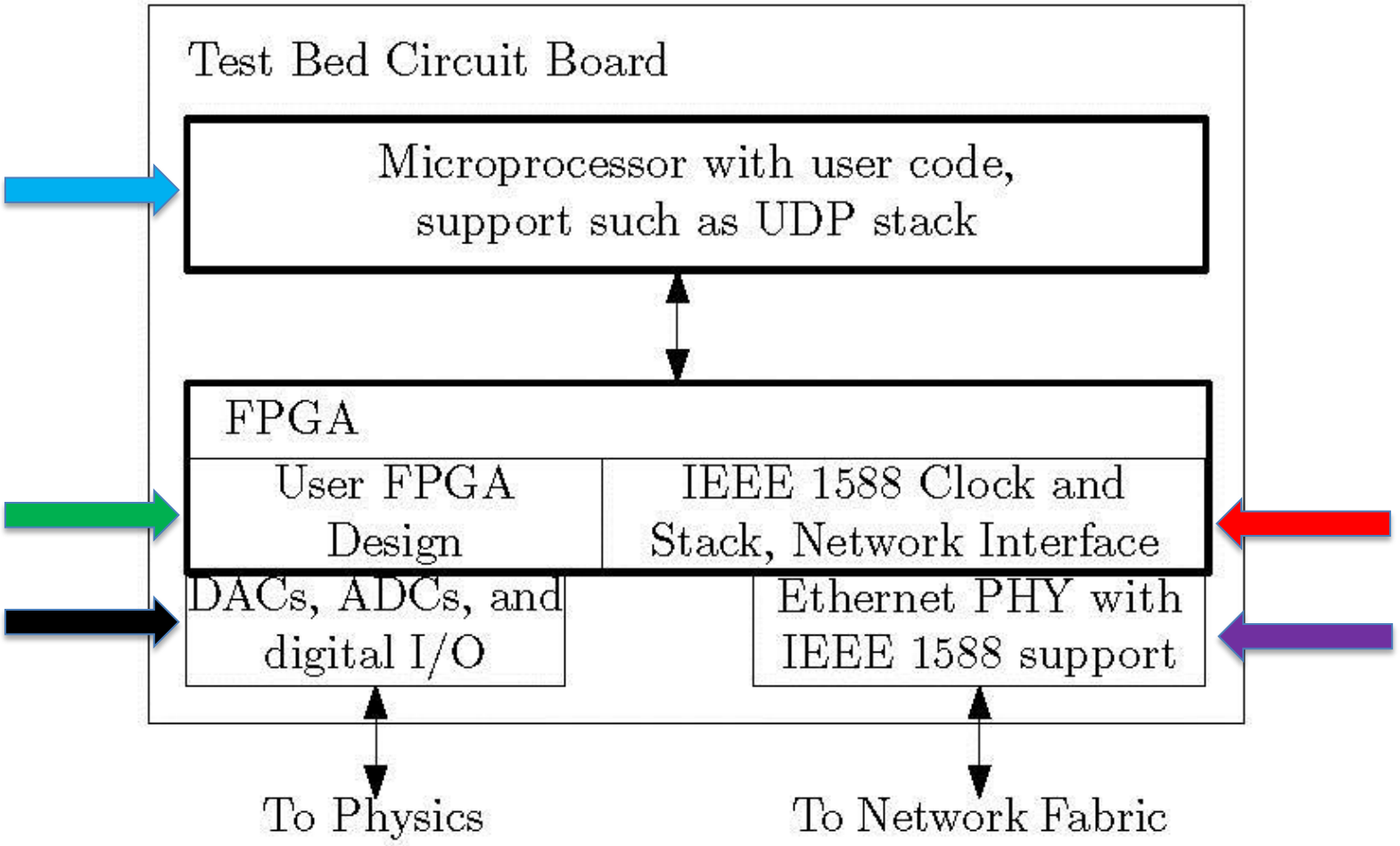
The purpose of the “correct-by-construction” timing testbed is to:

- Facilitate R&D, proof of concept, and collaboration in timing methodologies and design environments
- Allow comparison and verification of design alternatives.
- Explore metrics and test methods

# Testbed Architecture



# Testbed CPS Node Architecture



# The Testbed "Hello World" examples

Calnex

To shorten the learning curve in using the testbed we propose two hello world examples:

- A time triggered implementation
- A Ptides implementation

Both will implement the same application-tentatively measuring and adjusting the phases of two legs of a mock power grid prior to interconnection.

# Conclusions

Calnex

- There is still a lot of work to do before we have correct-by-construction timing
- The proposed testbed will enable more rapid progress particularly in the areas of:
  - Designs for hardware support of explicit time,
  - Designs for true real-time operating systems,
  - Languages, compilers, and other software development infrastructure,
  - Techniques for exploiting explicit time in applications.

Calnex

# Thanks for your attention!