The Hidden Tax of Bad FPGA Project Methodology

by Kamil Rudnicki, PhD

1 July 2026, FPGA Conference, Munich, Germany

What affects high efficiency at FPGA projects?

INTERNAL FACTORS:
- **poor methodology**
- poor tools (e.g. 15W CPU)
- poor communications
- poor project management
- poor practices
- ...

EXTERNAL FACTORS:
- broken tools and IPs
- mistakes in documentation
- ...

What is methodology?

A **methodology** is a system of

structured methods, principles, and rules

used to:

- conduct research,
- solve problems,
- or complete a project.

What happens?

Do it **right**.

- accept the delay
- understand it
- fix it (at the source) and forget

Take a **technical debt**.

- make a workaround

What happens?

Technical debt

Do it **right**.

- accept the delay
- understand it
- fix it (at the source)
  and forget

Why do we keep breaking it?

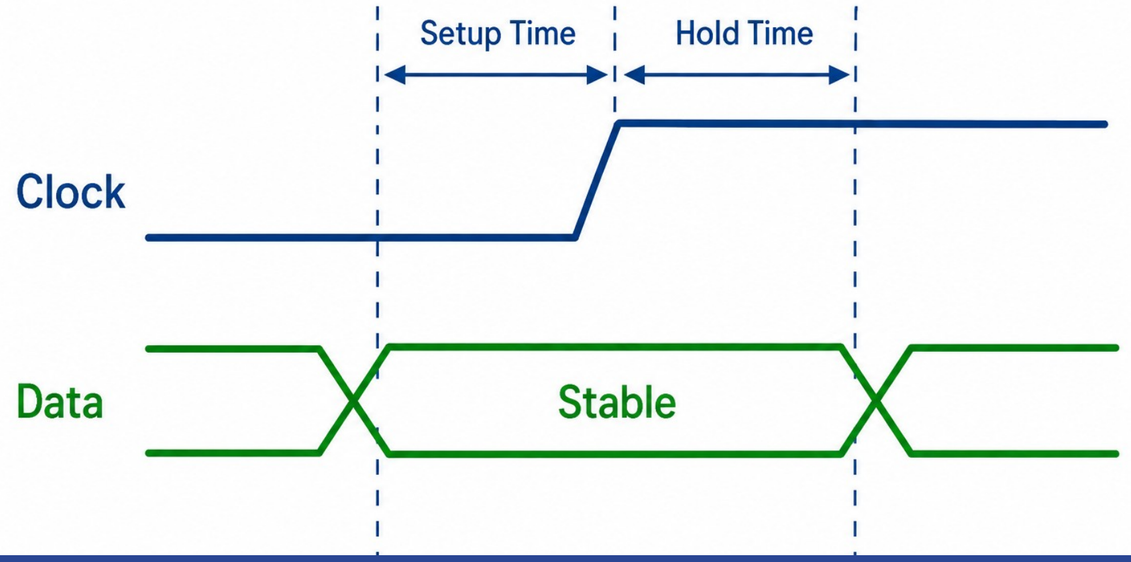
Full methodology works on any project.

~90% of projects use low-end FPGAs.

OVERSIMPLIFICATION

The higher the frequency, the more timing issues we get.
~~CLAIM: With low frequency there are no timing issues.~~

ALWAYS set input/output delays.
Use correct clocking structure.



What about documentation?

Example 1. Schematics designed. PCB ready. Need gateware.

Design with various interfaces:
10G Ethernet, ADCs (LVDS), USB, etc.

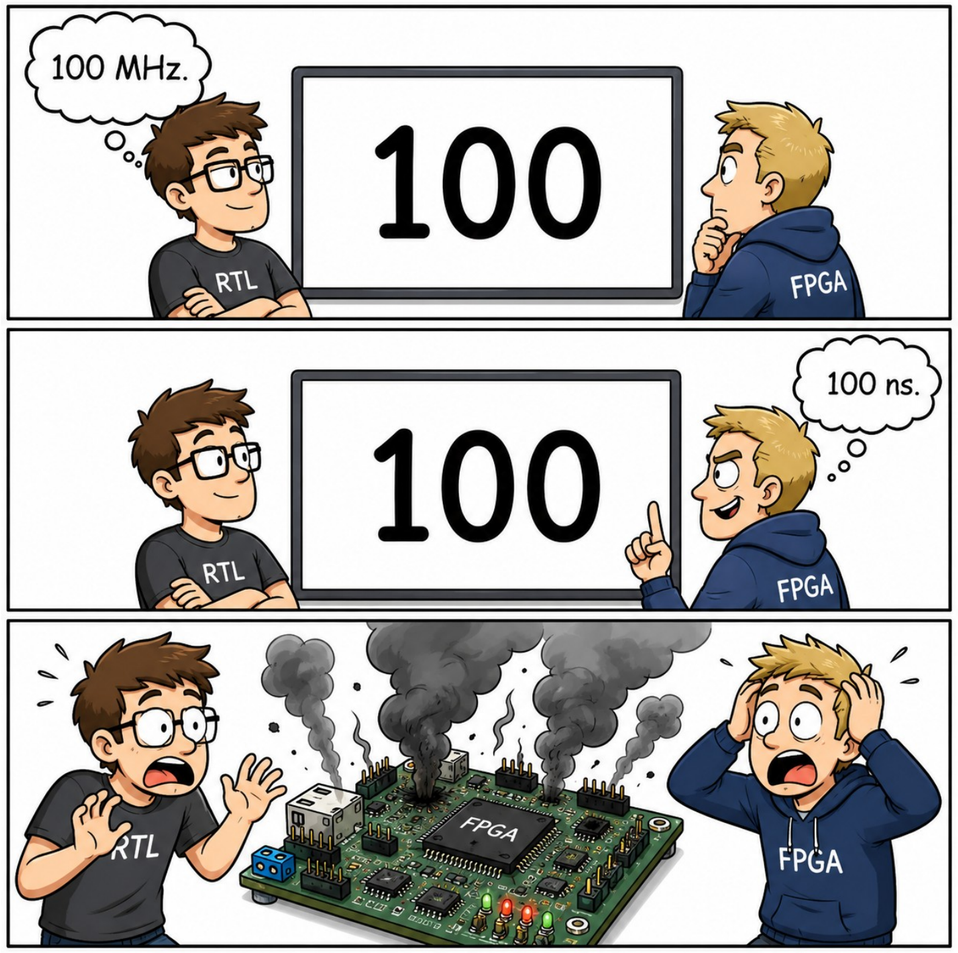
Started with 10G Ethernet. Observation:
After 60-90 seconds JTAG fails.

What actually happened?
No external DIFF_TERM for LVDS.
Lost 2 days worth of work.

Why it happened?
No documentation.

Assumptions?

Example 2. Assumptions.



8x 10G Ethernet... 8x 10G Ethernet...
8x 10G Ethernet... 8x 10G Ethernet...

It turned out to be 2x 40G Ethernet.
HW the same, but different GW.
Broken IP - 1 week of wasted time.

Freeze: tools, IPs,
requirements...

Why it happened?

No documentation.

What about documentation?

Example 3. Full gateware needed with client's IPs.

The reality of the IP:

- no documentation
- did not function correctly
- contained mistakes
- only after many messages some support started

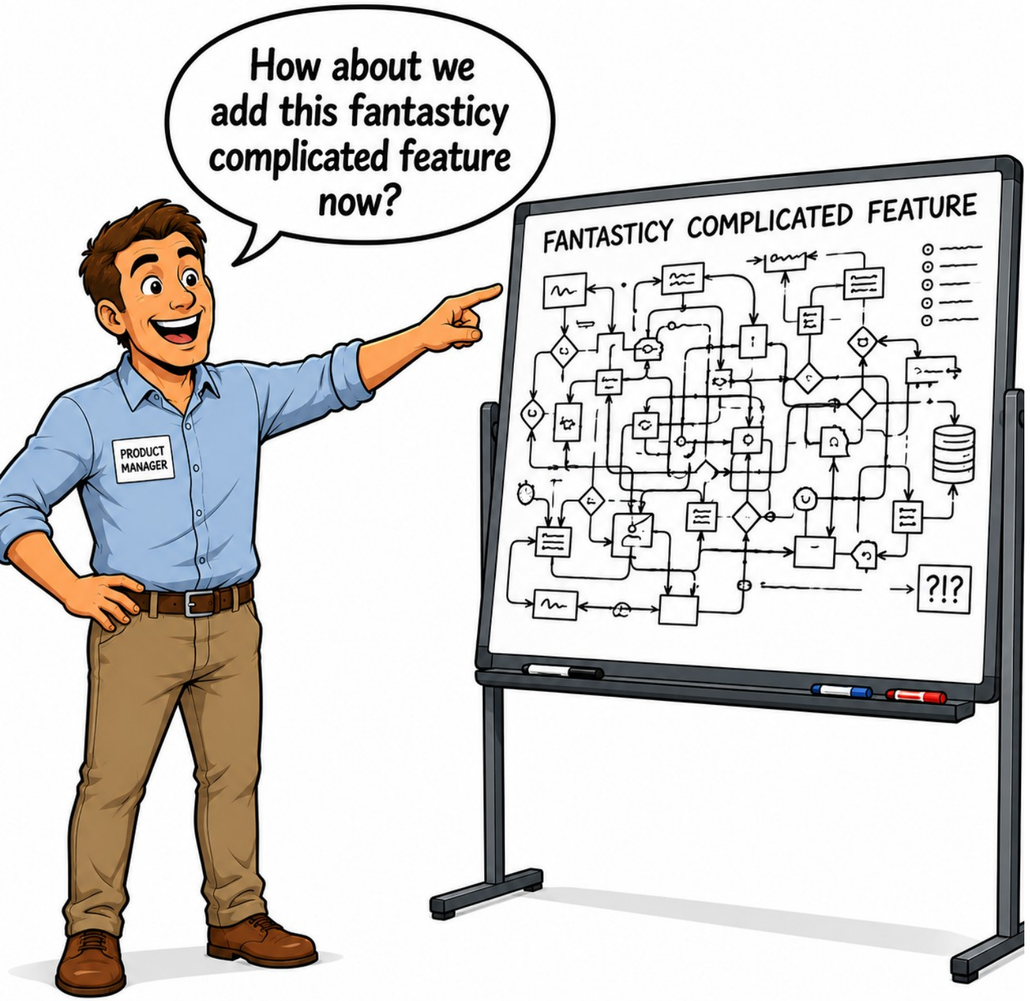
Why it happened?

No documentation.

No maintainance of the internal knowledge.

What about documentation?

Example 4. Unstable requirements.



Go back to:

- the architecture
- system engineering
- HW design
- RTL coding

Why it happened?

No strong vision of the product.
No freeze.
No documentation.

What about documentation?

Documentation – a golden reference point.

Documentation is **not optional**.

Documentation is for:

- planning and predicting potential issues BEFORE they occur
- communication between teams
- historical reference in the future (justify the project decisions)

How early do you verify synthesizability?

Example 5. The PCB production is done. There is a timing closure issue.

Guideline: Pin Placement for General Purpose High-Speed Signals

For general purpose high-speed signals faster than 200 MHz, follow these guidelines to ensure I/O timing closure.

- Avoid using HMC DQ pins as the input pin.
- Avoid using HMC DQ and command pins as the output pin.

I/O signals that use the hard memory controller pins are routed through the `HMCPHY_RE` routing elements. These routing elements have a higher routing delay compared to other I/O pins. To identify the hard memory controller pins for your Cyclone V device and package, refer to the relevant pin-out files.

Related Information

Cyclone V Device Pin-Out Files
Provides the pin-out files for each Cyclone V device package.

Yes, you are.

Preparing a skeleton project is a good engineering practice.

Brightelligence® whitepaper – Push FPGA boundaries

How quickly do you check?

Example 6. Long design cycles.

HW, SW, FW, GW, algorithms
all developed **independently.**

Never tested. Never verified.

Put together and… It didn't work.

Why it never worked?
Not enough feedback in the process.

How quickly do you check?

Example 6. Long design cycles.

How quickly do you check?

Example 6. Long design cycles.

Interface tester to verify pure hardware.

Testing, checking, verifying, simulating…

Proper integration phases:

- HW-GW
- GW-FW
- FW-SW

Testing, checking, verifying, simulating…

Result?

From years of no success,
to a product in 5 months.

Missing requirements?

1) MTBF is not specified.

How can I design it properly?

- How many FFs should I use in 1 bit synchronizer?
- Should I use a safe FSM?

2) Trace lengths (board skew)
3) Pin standards
4) Performance: latency, frequency, power, margins, resources…
5) Acceptance criteria

and many more…

Why it pays of to follow proper methodology?

Example 7. Comparison

Name	Project **X**	Project **V**
Use of client IP	Must	Can, but no
Documentation	None	None / Provided
Issue detection	Clock domains and parameters	Early, interface tester with stress tests
Support	None / Little (with inconclusive, misleading answers)	Optional
Technical debt	IPs (mistakes, not verified, non-intuitive)	No historical record
Complexity	Low	Fairly complex, lots of synchronized events, HSSI
Size of the FPGA	Small (AUS+)	Mid (KUS)
Time frame	10 months and counting…	5 months (completed)

What are the consequences?

- no scripts → no reproducibility
- no git → no traceability
- overcomplicated structures → no maintainability
- no architecture design → no clear vision of target
- we do things differently → understand that as broken methodology
- no documentation → tribal knowledge (difficult to support)
  time to market increases

- not carring about warnings → hidden issues (technical debt)
- not carring about timing issues → this will take a long time to finish
- not receiving requested info → the project cannot be completed

making the decision != methodology

Bright FPGA Academy

A transition from junior to mid-level FPGA engineer.

Long-term structured supervised hands-on course – **2-3 times faster results**

Full FPGA product develepment cycle.

Lectures, challenging homeworks (with technical feedback), consultations, exams, final project...

Own selection of platform and language.

Repeated cycle of
learning, applying, reviewing, improving, and documenting

ENGINEERING

Thank you

Brightelligence sp. z o.o.

ul. Switezianki 16
91-496, Lodz, Poland

https://www.linkedin.com/company/brightelligence/

Kamil Rudnicki

https://www.brightelligence.com
kamil.rudnicki@brightelligence.com
+48 888 888 351
https://www.linkedin.com/in/rudnickikamil/