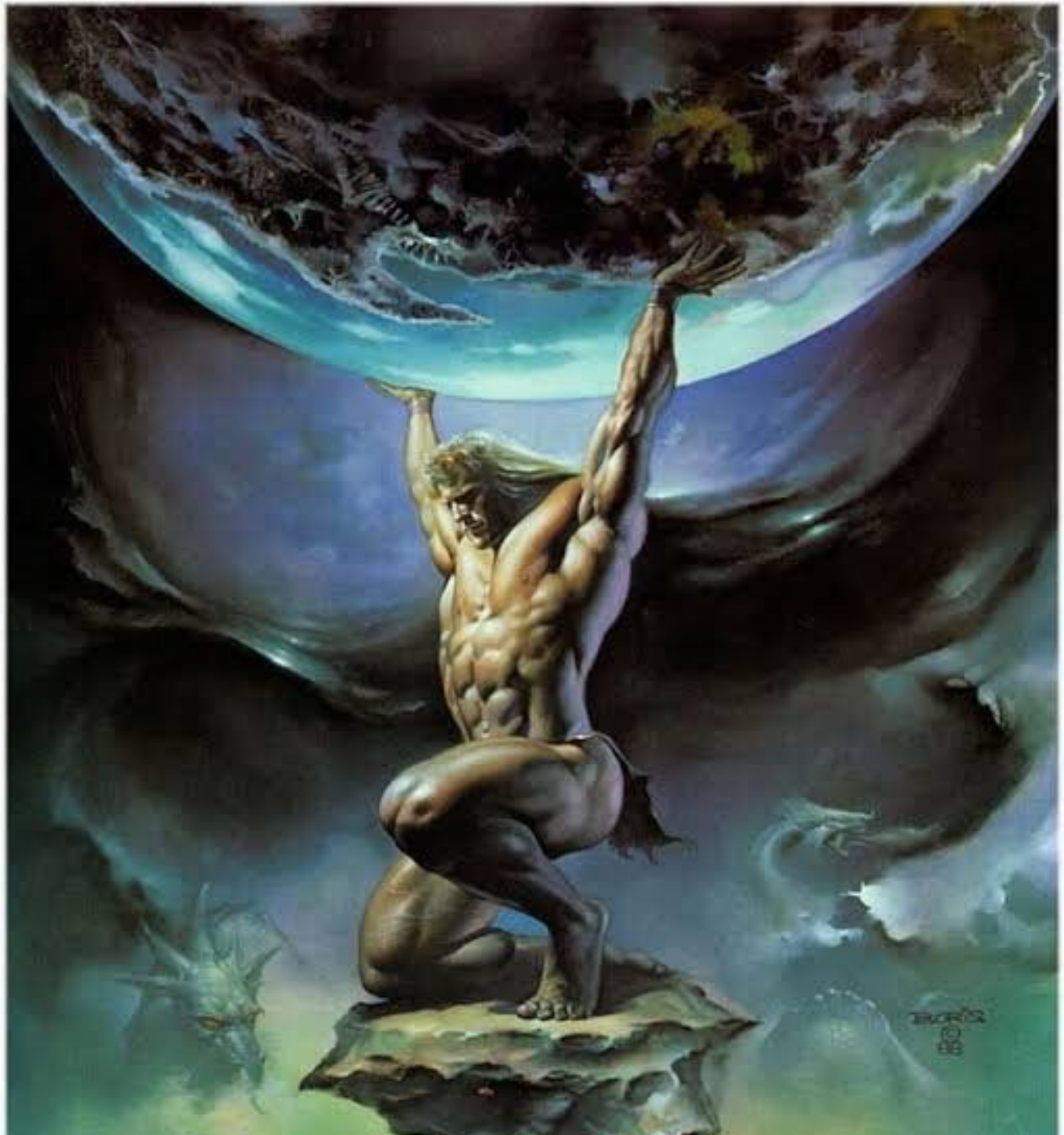
wat: hardware edition

Emil J / widlarizer

TCL



TCL

```
emil@blahaj ~> tclsh
% set x 1
1
% set x $x+1
```

TCL

```
1  emil@blahaj ~> tclsh
2  % set x 1
3  1
4  % set x $x+1
5  1+1
```

TCL

```
1  emil@blahaj ~> tclsh
2  % set x 1
3  1
4  % incr x
5  2
```

TCL

> … there is no syntax, or it's so simple you have to do everything outside the syntax.

- Guido van Rossum, *1999*

> It was designed to be a "scripting language", on the assumption that a "scripting language" need not try to be a real programming language. … It lacks arrays; … It fakes having numbers, which works, but has to be slow.

- GNU Jesus, 1994

> There are people at the office who have been dealing with this crap for 20 years and still can't tell me off the top of their head the order of parameters to basic functionality functions.

- some guy on orange site

Verilog

Verilog

_1_ and _2_ compute different values. Credit: @whitequark

```
 1  module top(a, b, c);
 2      input a;
 3      input [2:0] b;
 4      input [3:0] c;
 5      wire [3:0] _0_;
 6      wire [3:0] _1_;
 7      wire [3:0] _2_;
 8      assign _0_ = + $signed(b);
 9      assign _1_ = a ? _0_ : c;
10      assign _2_ = a ? (+ $signed(b)) : c;
11  endmodule
```

Verilog

> +m Unary plus m (same as m)

- IEEE 1364 Verilog HDL

> For the arithmetic operators, if any operand bit value is the unknown value x or the high-impedance value z, then the entire result value shall be x.

- also IEEE 1364 Verilog HDL

Table 17-5—Mnemonics for strength levels

Mnemonic	Strength name	Strength level
Su	Supply drive	7
St	Strong drive	6
Pu	Pull drive	5
La	Large capacitor	4
We	Weak drive	3
Me	Medium capacitor	2
Sm	Small capacitor	1

Verilog

`wire` vs `reg`

???

Verilog

- one (1) good parser
- others so bad they literally crash your GUI
- “safe subset”
- IR?!
- every sufficiently complex HW design contains a verilog codegen
- non-synthesizable Verilog

EDA

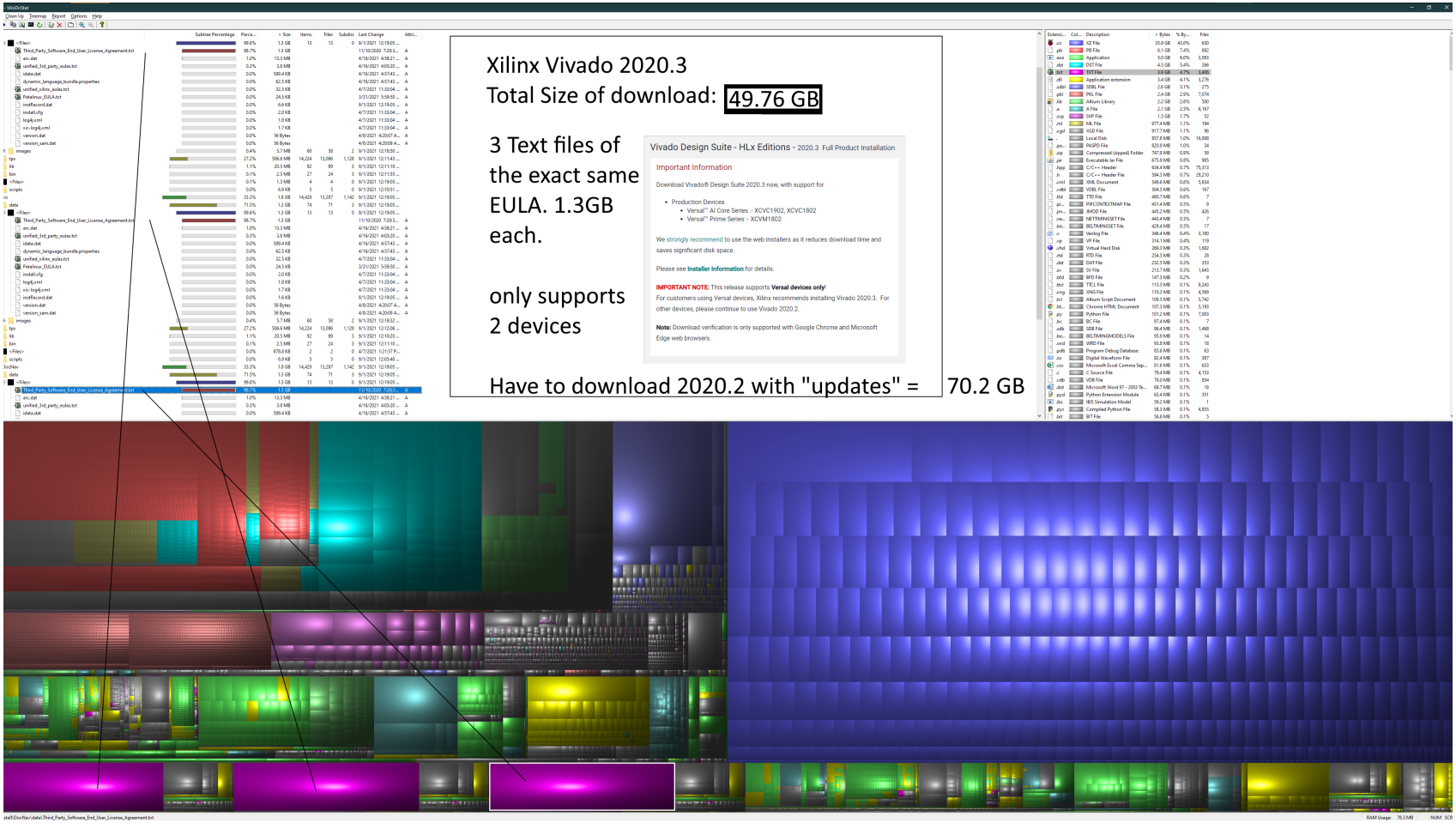


Figure 3: Xilinx Vivado, 3k USD

EDA

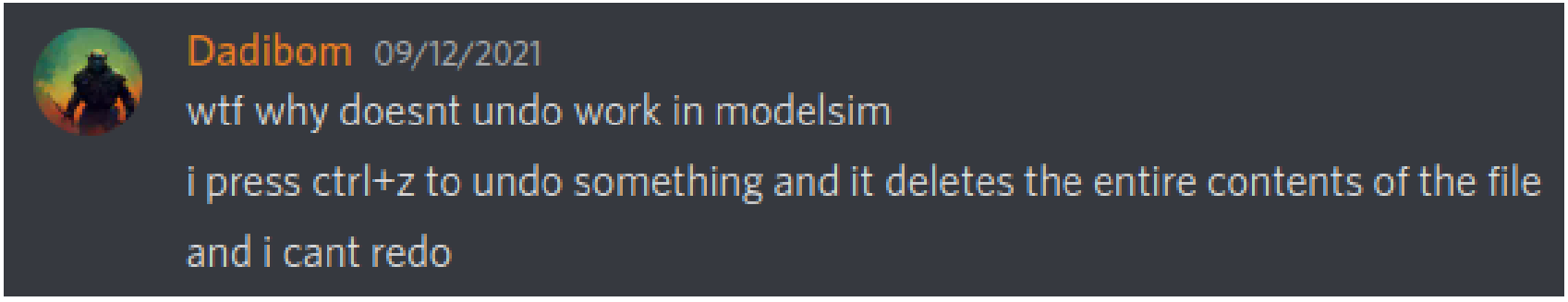


Figure 4: Modelsim

- nevermind, TCL is fine actually

EDA

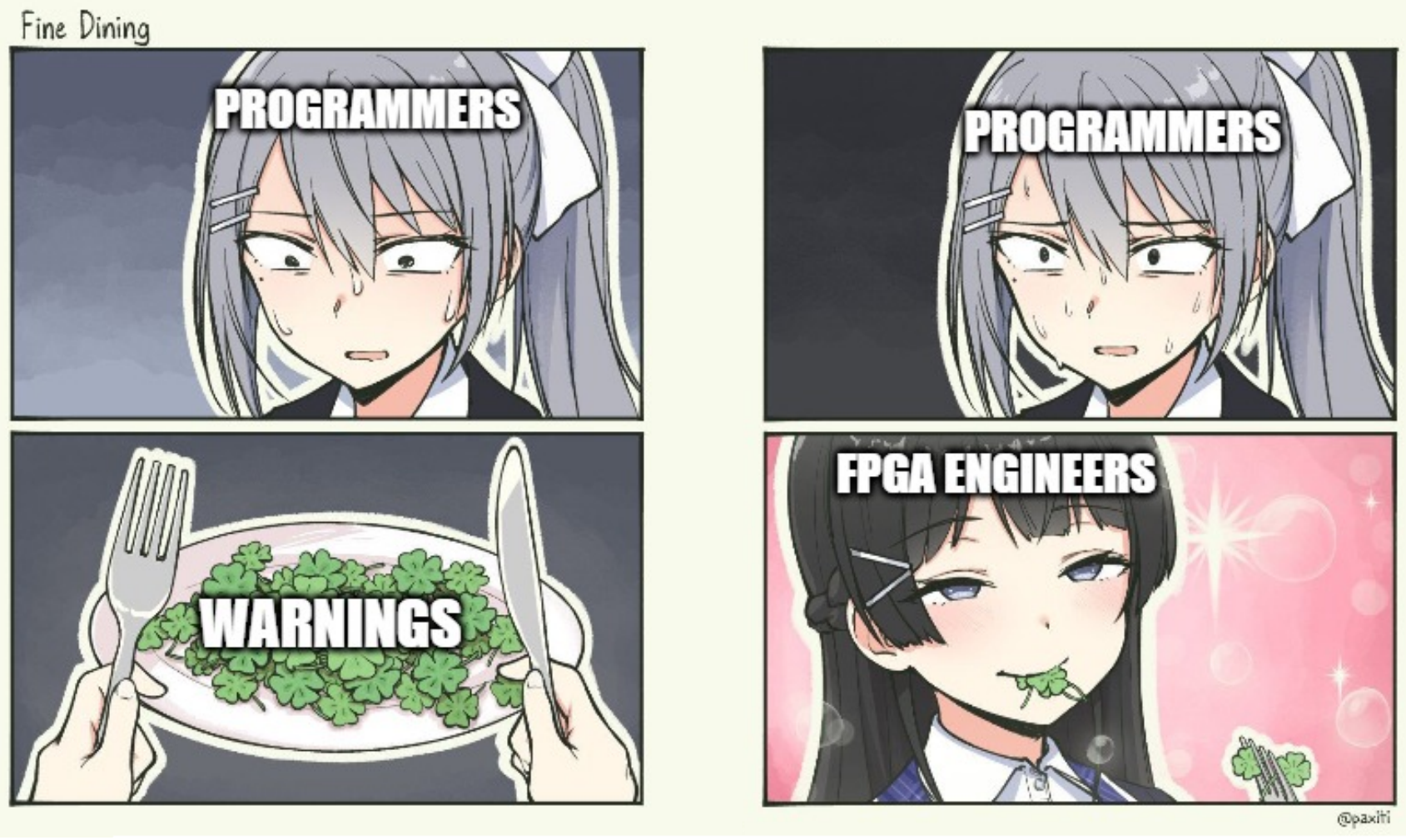


Figure 5: warnings

EDA

```
13221   set_property src_info {type:SCOPED_XDC file:87 line:13 export:INPU
13222   # Xilinx, and to the maximum extent permitted by applicable
13223   set_property src_info {type:SCOPED_XDC file:87 line:14 export:INPU
13224   # law: (1) THESE MATERIALS ARE MADE AVAILABLE "AS IS" AND
13225   set_property src_info {type:SCOPED_XDC file:87 line:15 export:INPU
13226   # WITH ALL FAULTS, AND XILINX HEREBY DISCLAIMS ALL WARRANTIES
13227   set_property src_info {type:SCOPED_XDC file:87 line:16 export:INPU
13228   # AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING
13229   set_property src_info {type:SCOPED_XDC file:87 line:17 export:INPU
13230   # BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-
13231   set_property src_info {type:SCOPED_XDC file:87 line:18 export:INPU
13232   # INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and
13233   set_property src_info {type:SCOPED_XDC file:87 line:19 export:INPU
13234   # (2) Xilinx shall not be liable (whether in contract or tort,
13235   set_property src_info {type:SCOPED_XDC file:87 line:20 export:INPU
13236   # including negligence, or under any other theory of
13237   set_property src_info {type:SCOPED_XDC file:87 line:21 export:INPU
13238   # liability) for any loss or damage of any kind or nature
13239   set_property src_info {type:SCOPED_XDC file:87 line:22 export:INPU
13240   # related to, arising under or in connection with these
13241   set_property src_info {type:SCOPED_XDC file:87 line:23 export:INPU
```

Figure 6: codegen wat

Hardware

You want to see side effects? I'll show you side effects

Hardware

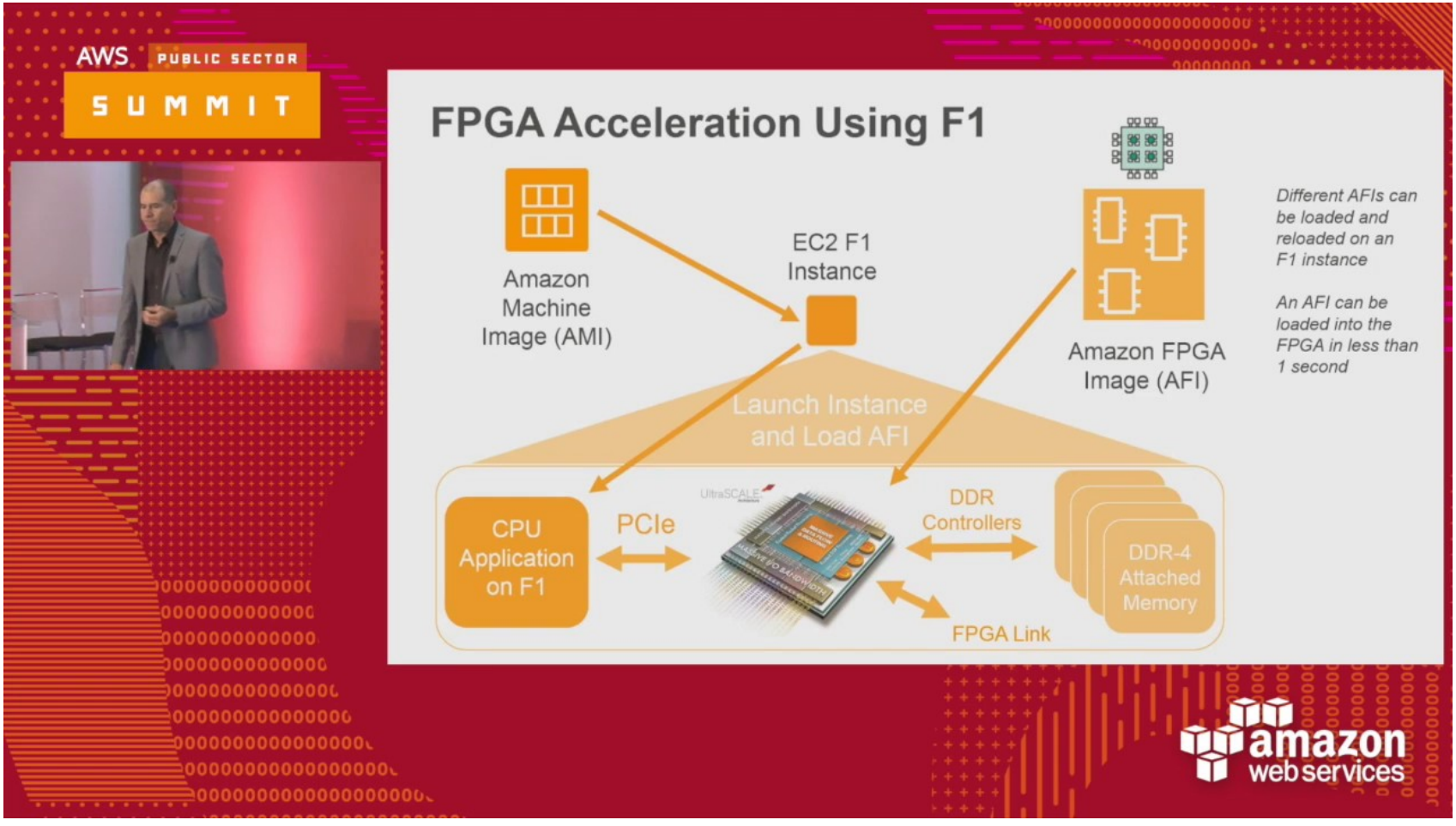


Figure 8: AWS F1

Hardware

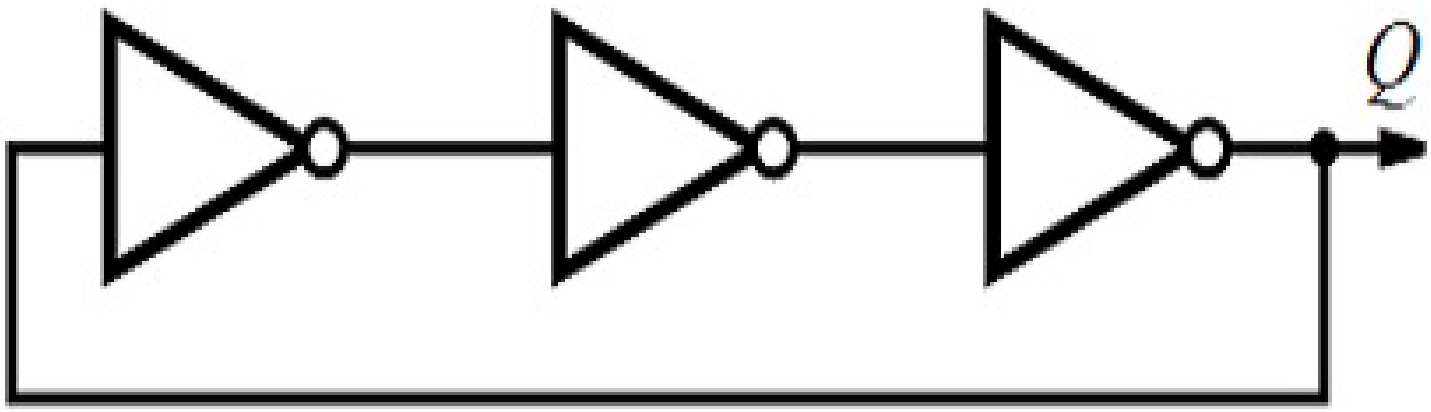
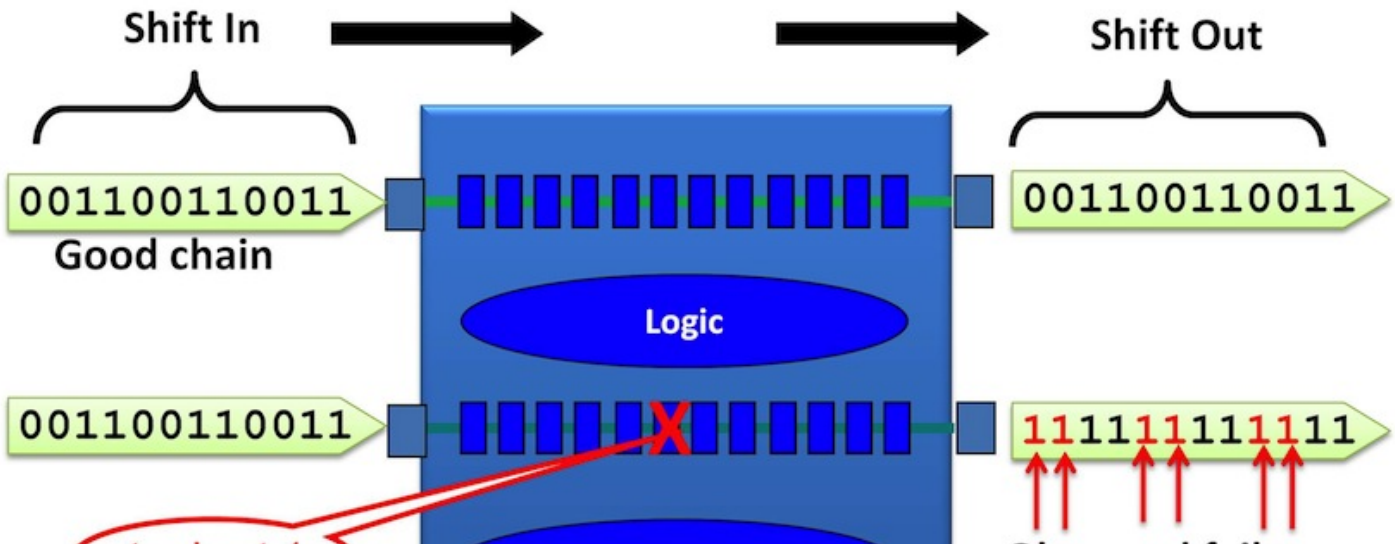


Figure 9: Ring oscillator

Hardware

> When the response to a test vector is captured by state elements in scan based tests, the switching activity of the circuit may be large resulting in abnormal power dissipation and supply current demand.

- 10.1109/TEST.2006.297694



Hardware bugs

Always read the errata.

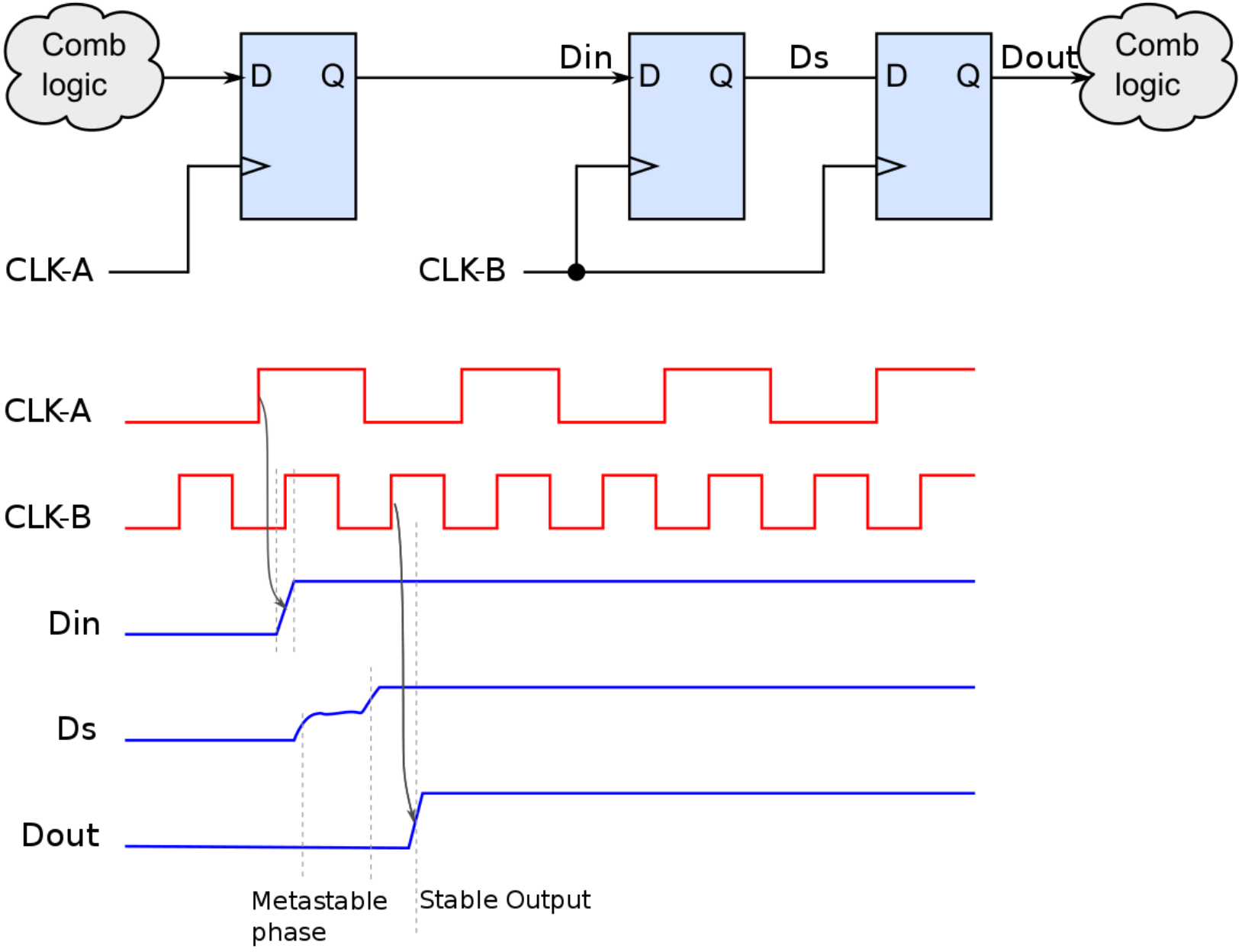
> I2C: In master receive mode, data remains latched in data register until new data is received.

> Workaround: When slave is configured to transmit data on an irregular basis, it should not sent 2 consecutive bytes with the same data

- STM SPC582Bz automotive qualified certified microcontroller errata

…how do I submit a pull request against a piece of melted sand?

Hardware bugs



What now?

- better languages! Clash, Chisel, Amaranth
- tooling? Yosys, simulators, place-and-route, ASIC fab stuff, formal verification
- industry-standard isn't all there is
- hardware *can* be fun
- small world with tough problems