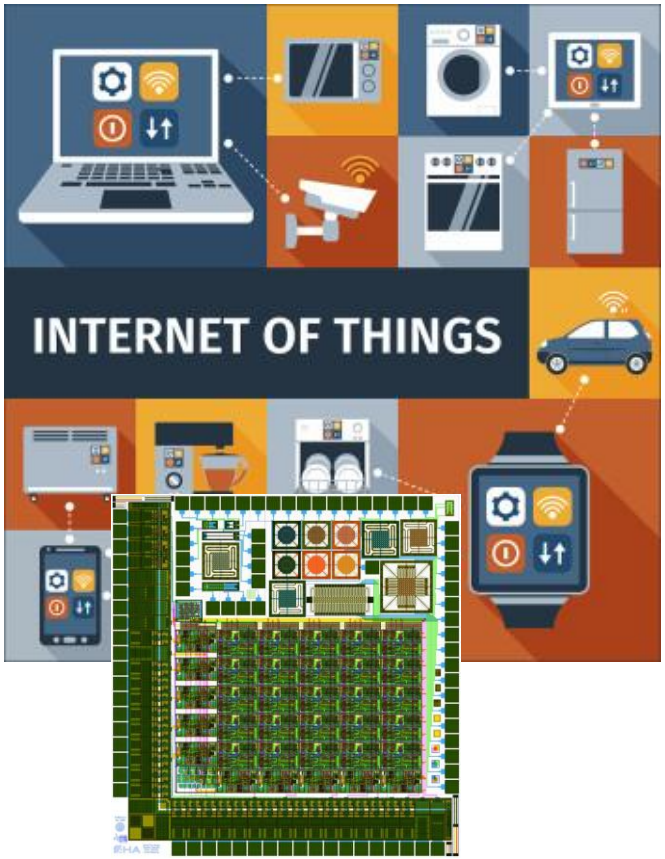
Go2UVM Package introduction

http://www.go2uvm.org

EDA start-up from India



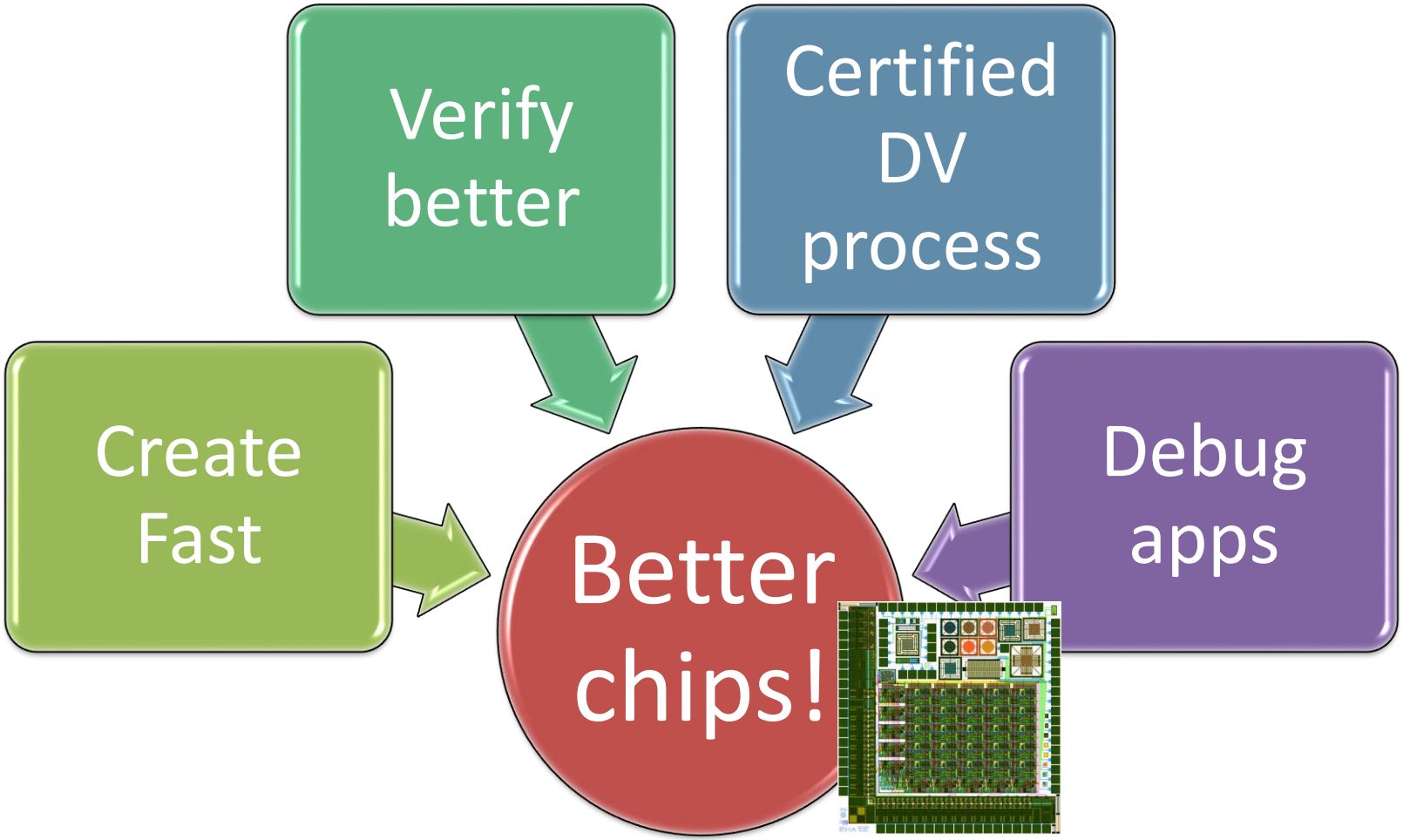
Problem?

Not enough DV expertise

Tech challenges - Low Power, Quality of DV

Business challenges - Small teams, short TTM

Our Value Proposition



Products at a glance

DVCreate

- SV Interface
- UVM, Go2UVM Package
- Low Power

DVRules

- UVM
- SVA
- SVTB

RTGen

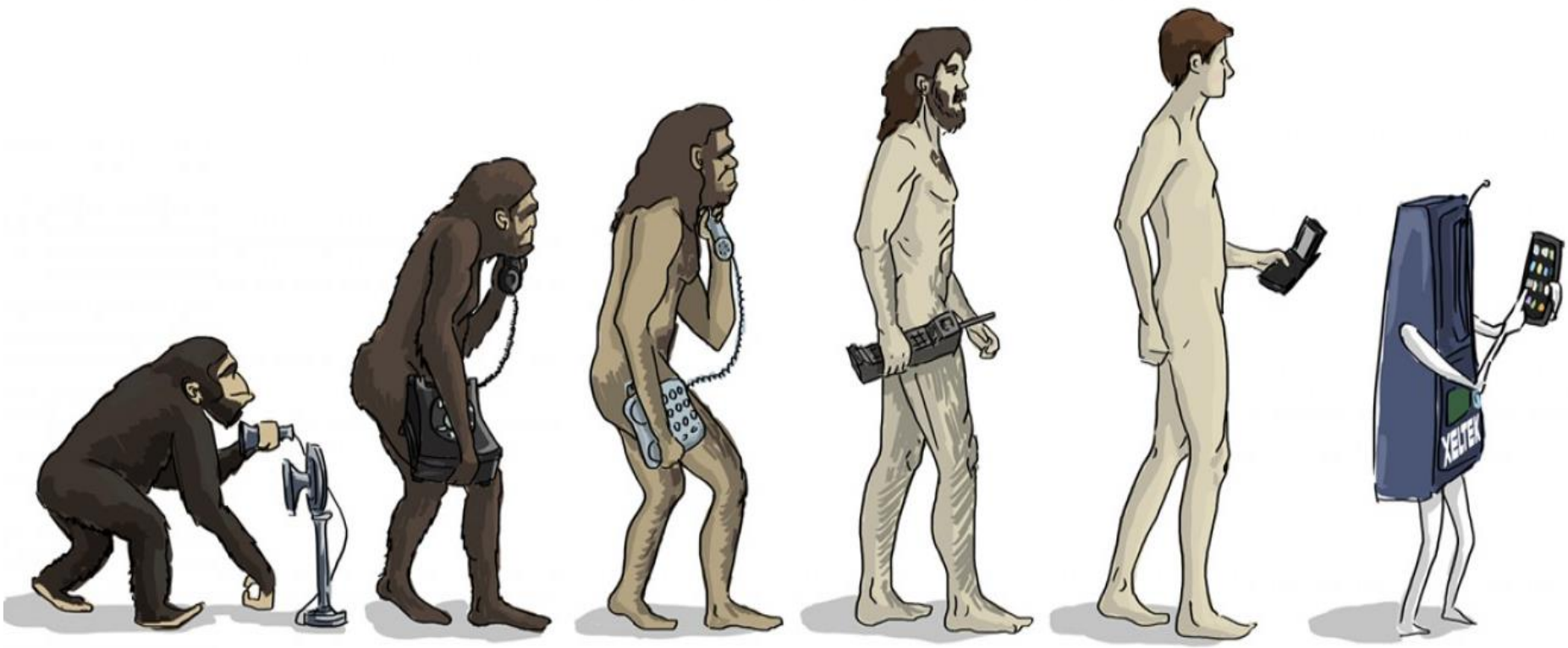
- SystemVerilog
- UVM
- Formal-to-Sim adapter

DVEnc

- Verilog
- SystemVerilog
- UVM

Go2UVM Package introduction

Mobile Phone evolution



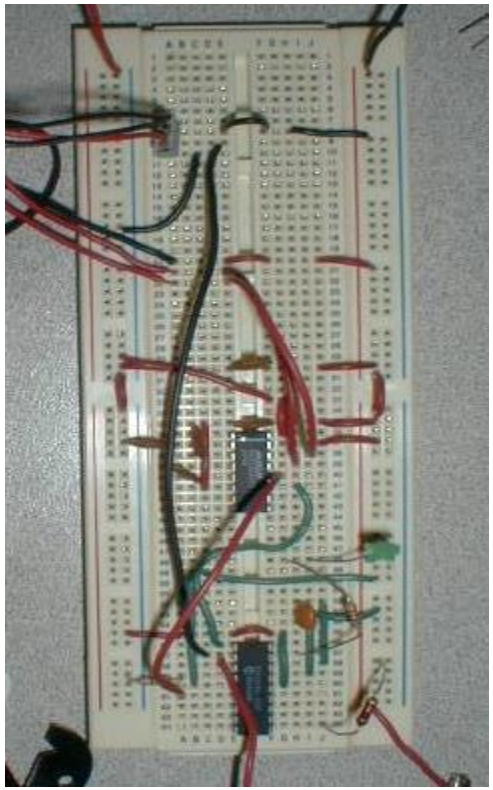
Quotes from Martin Cooper

- Well, we knew that someday everybody would have a [cell] phone
- Phones have gotten so complicated, so hard to use, that you wonder if this is designed for real people or for engineers.

UVM for verification similar?

Origins of “testbench”

Signal Generator + Supply etc.

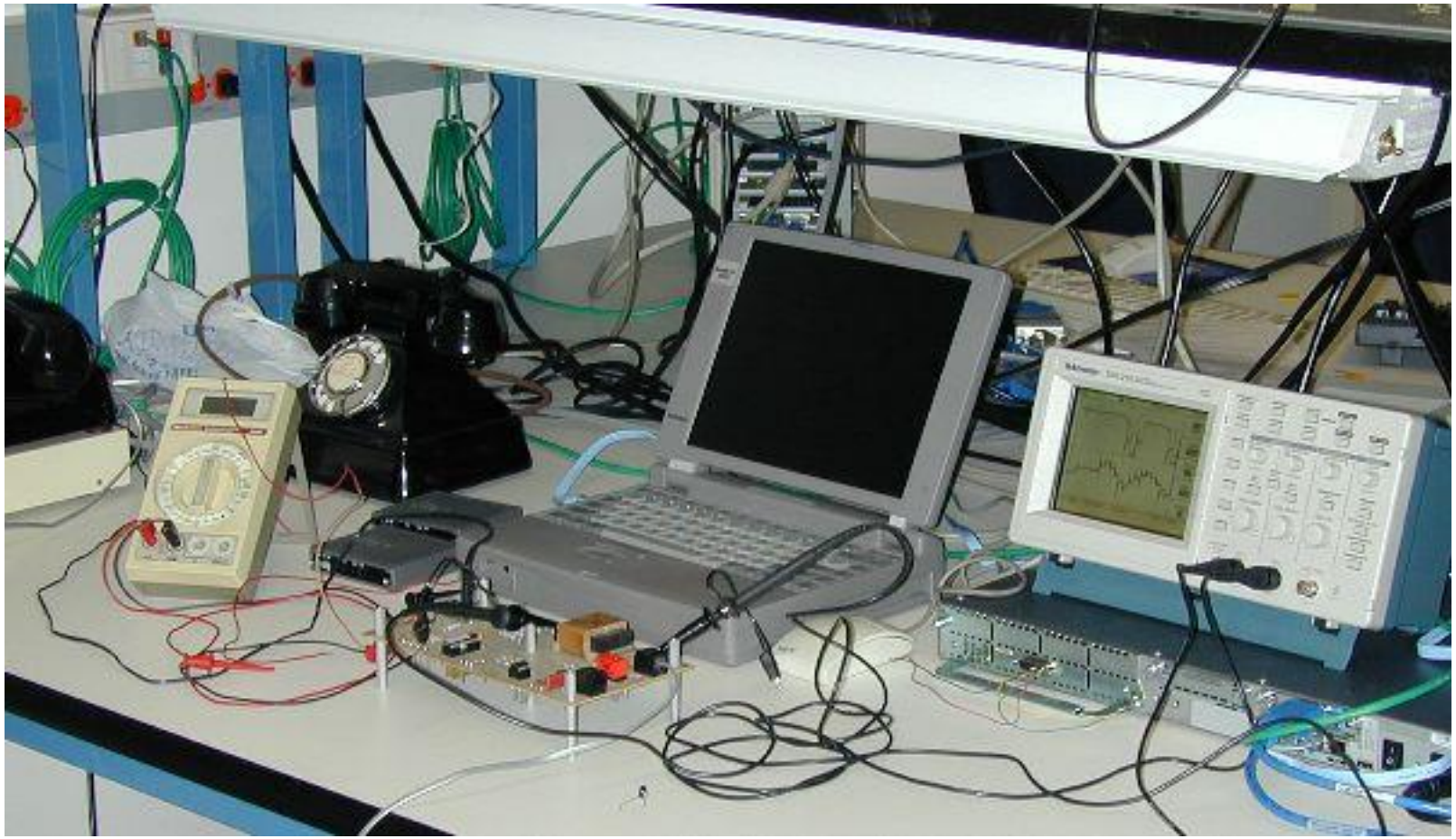


Design in Bread-board

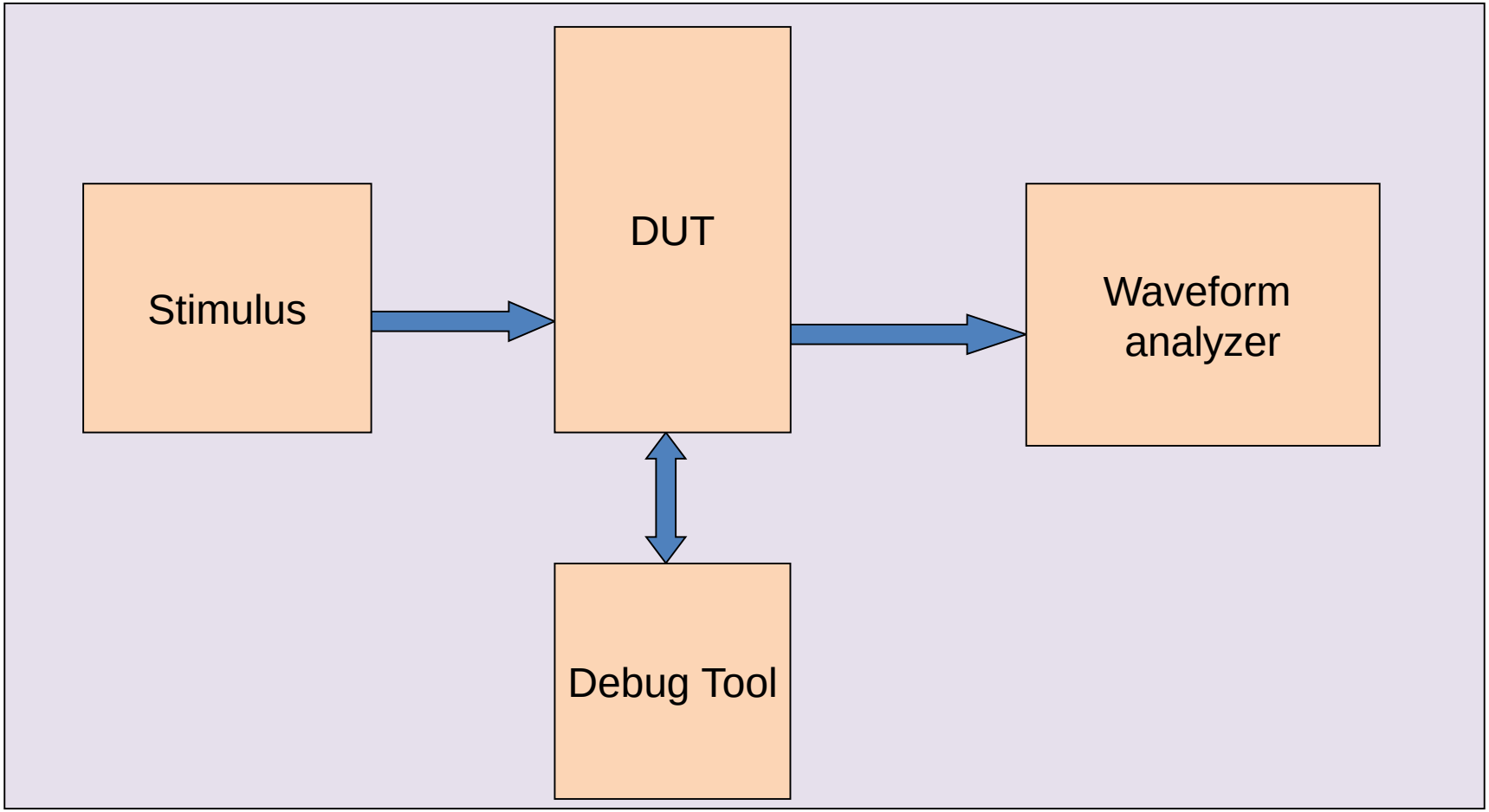
Oscilloscope

Multimeter

“Testbench”



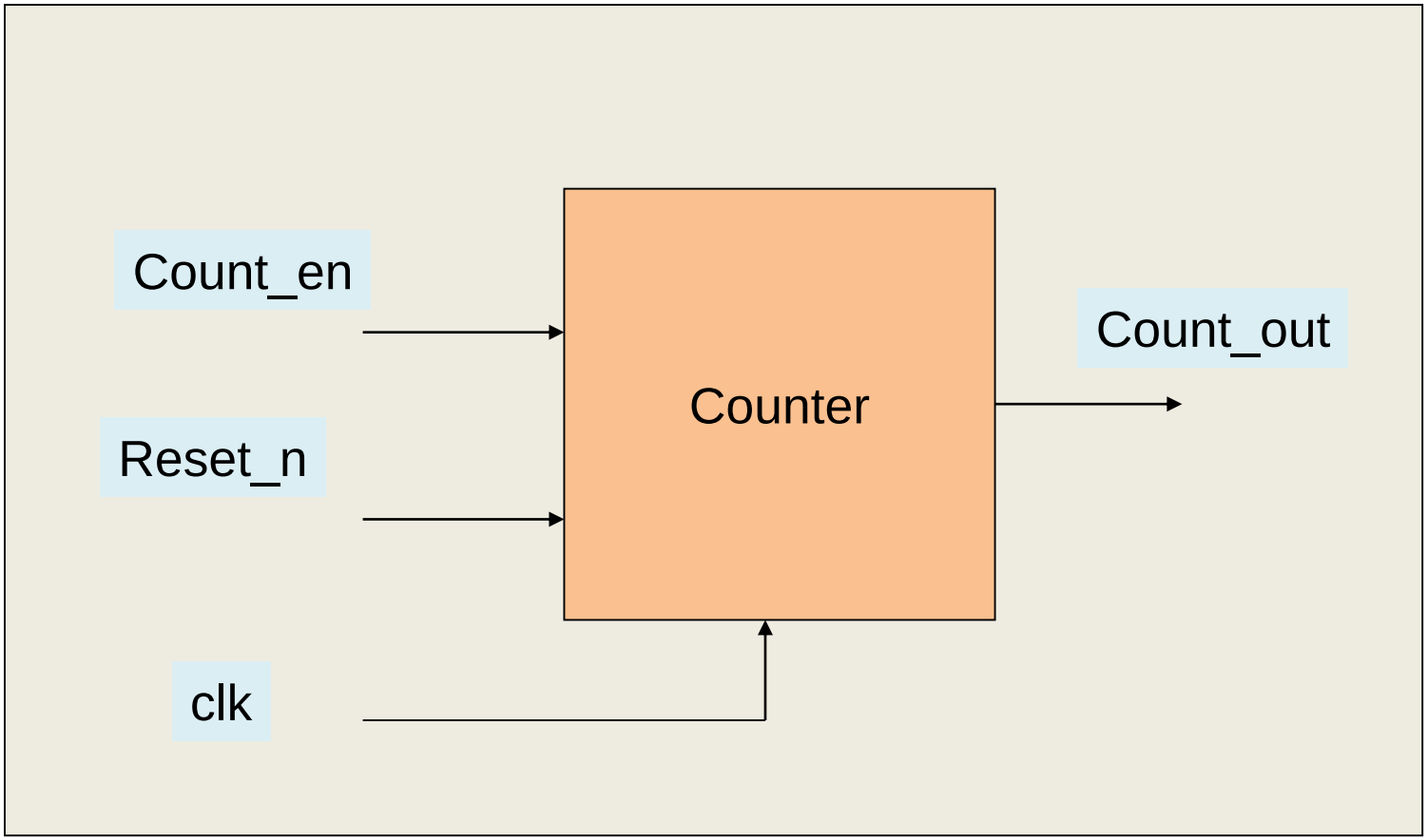
Testbench in Simulation world



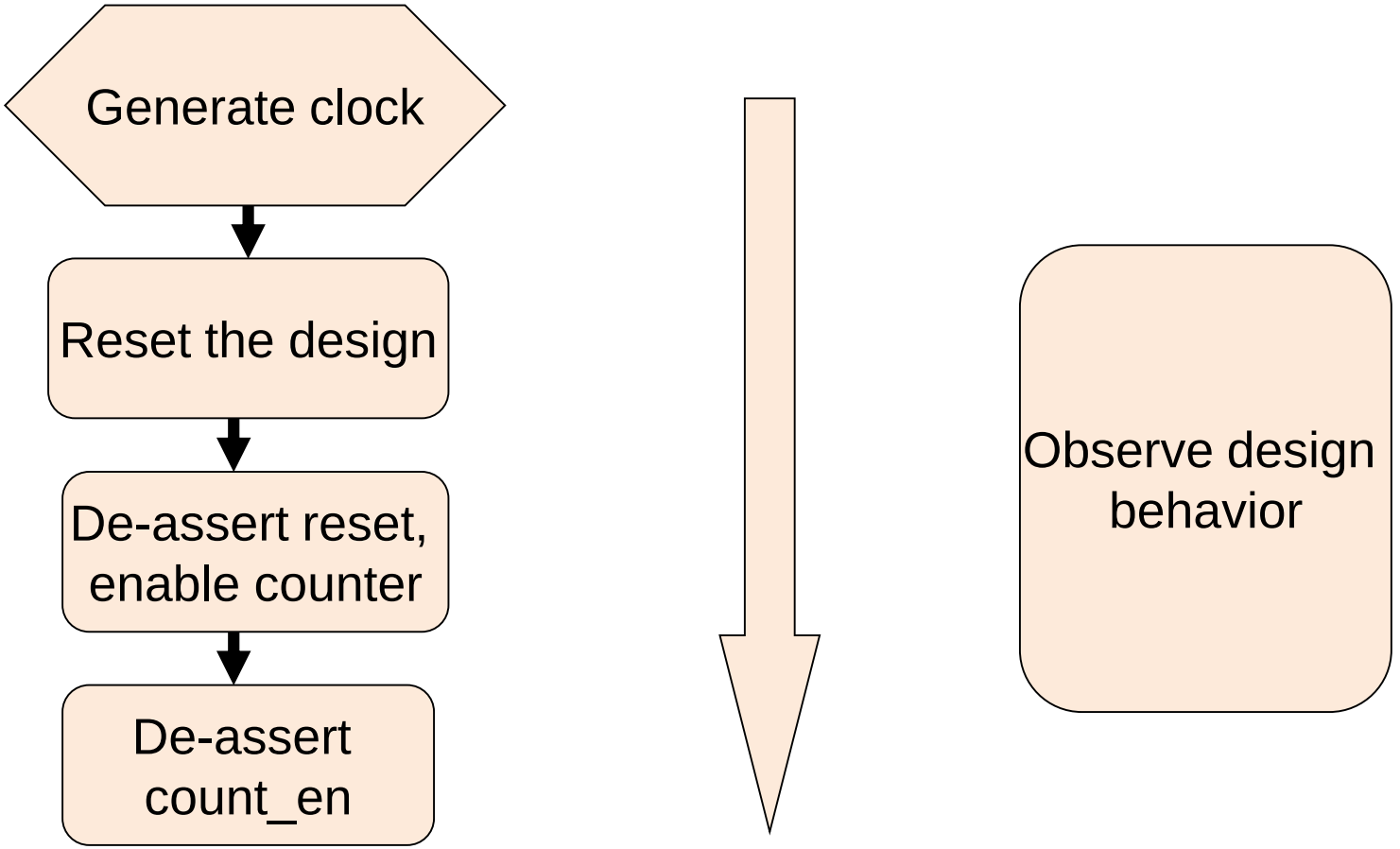
Types of Testbenches

- No single-size fits all!
- Depends on many things:
  - Complexity of the design
  - Is the design new or a derivative?
  - Application domain
  - Availability of tools, technologies
  - Skill set of team members

Counter DUT



Simplified Verification flow for the counter

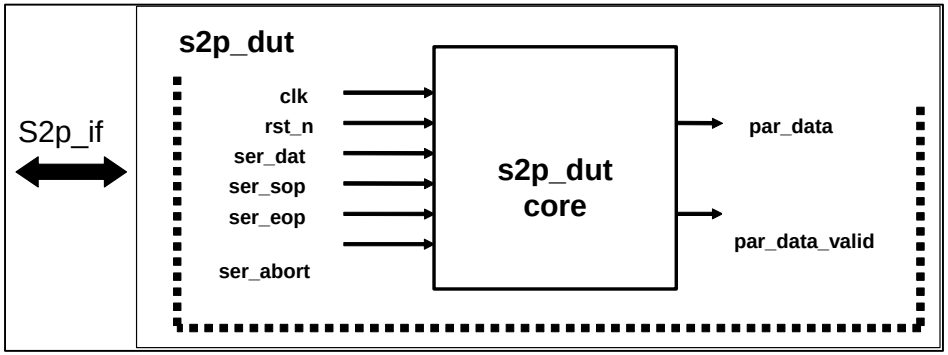


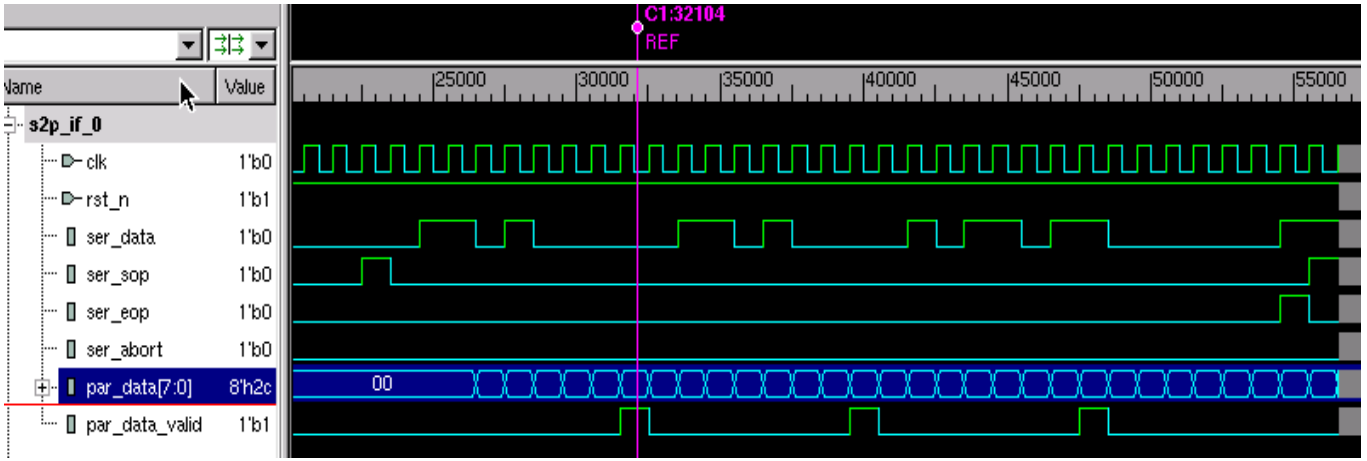
Linear Testbench

- First step into a proper Testbench
- Typically written in a HDL – Verilog/VHDL
- A simple, linear sequence of events over a period of several clocks
- Stimulus describes what happens in a series of assignments that are spread over time

```
initial begin : stim
  #10 reset = 1'b1;
  #100 reset = 1'b0;
  #10 inp_1 = 16'haabb;
        inp_2 = 4'b0010;
  #10 if (out_1 != inp_1 && inp_2) $display ("Error");
.....
end : stim
```

S2P Interface and Timing

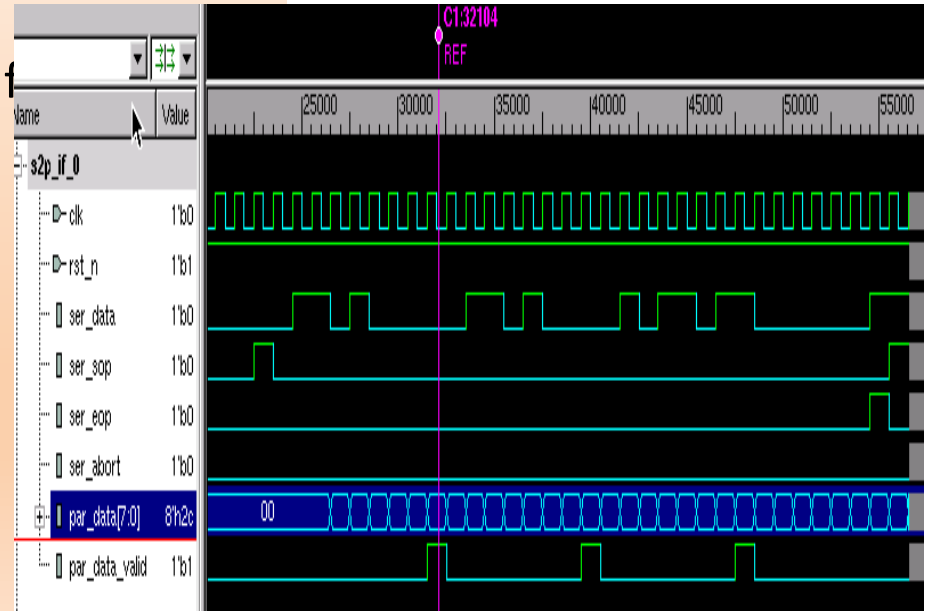




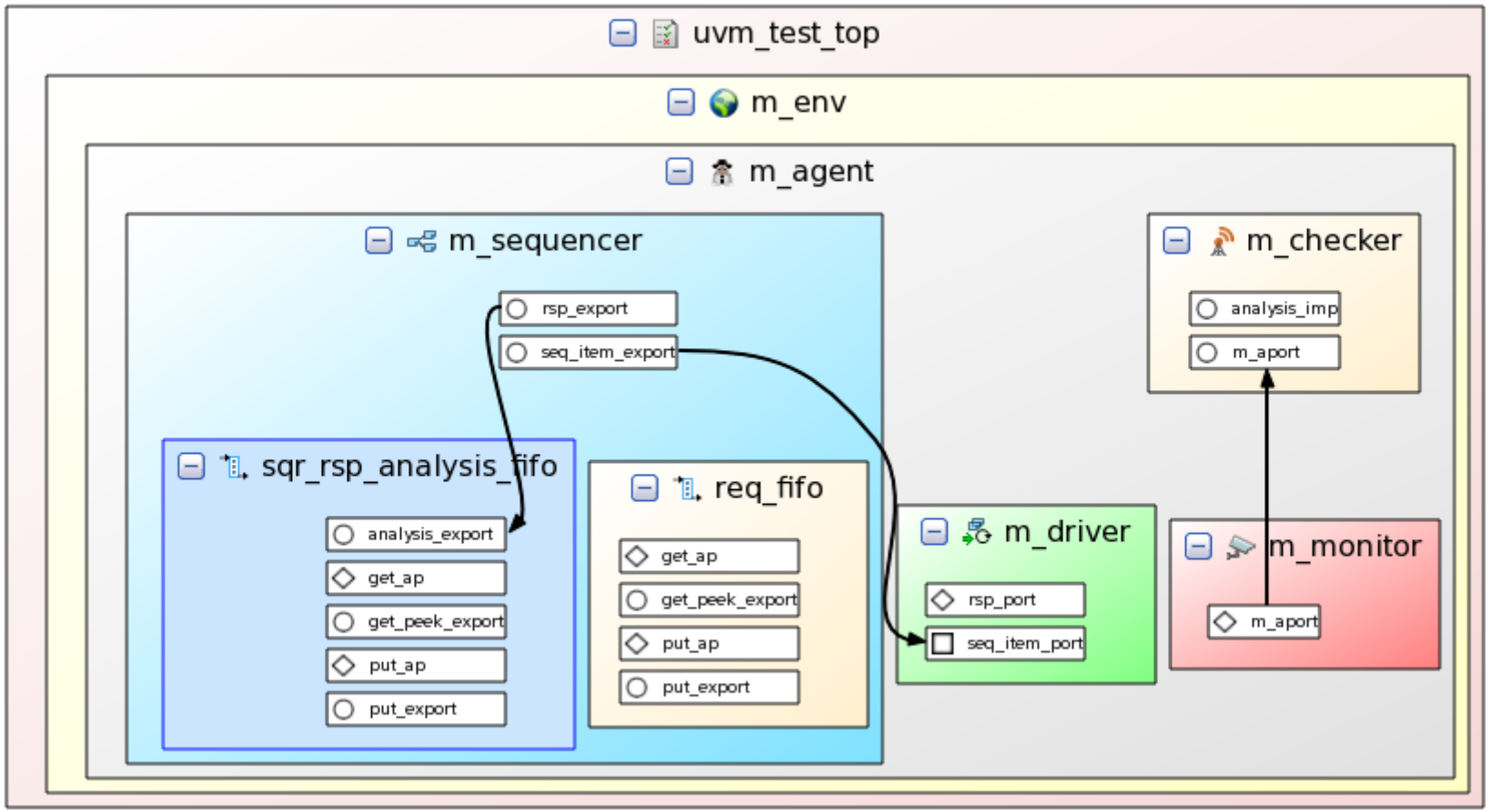
Procedural Testbench – S2P example

BFM model

```
task send_ser_pkt (logic [31:0] data);
   ser_sop <= 1'b1;
   ser_eop <= 1'b0;
   ser_abort <= 1'b0;
   ser_data <= data[0];
   @ (posedge clk);
   ser_sop <= 1'b0;
   for (int i=0; i <= 30; i=i+1) begin : f
     ser_data <= data[i];
     @ (posedge clk);
   end : for_0
      ser_data <= data[31];
   ser_eop <= 1'b1;
   @ (posedge clk);
   ser_eop <= 1'b0;
endtask
```



Typical UVM architecture

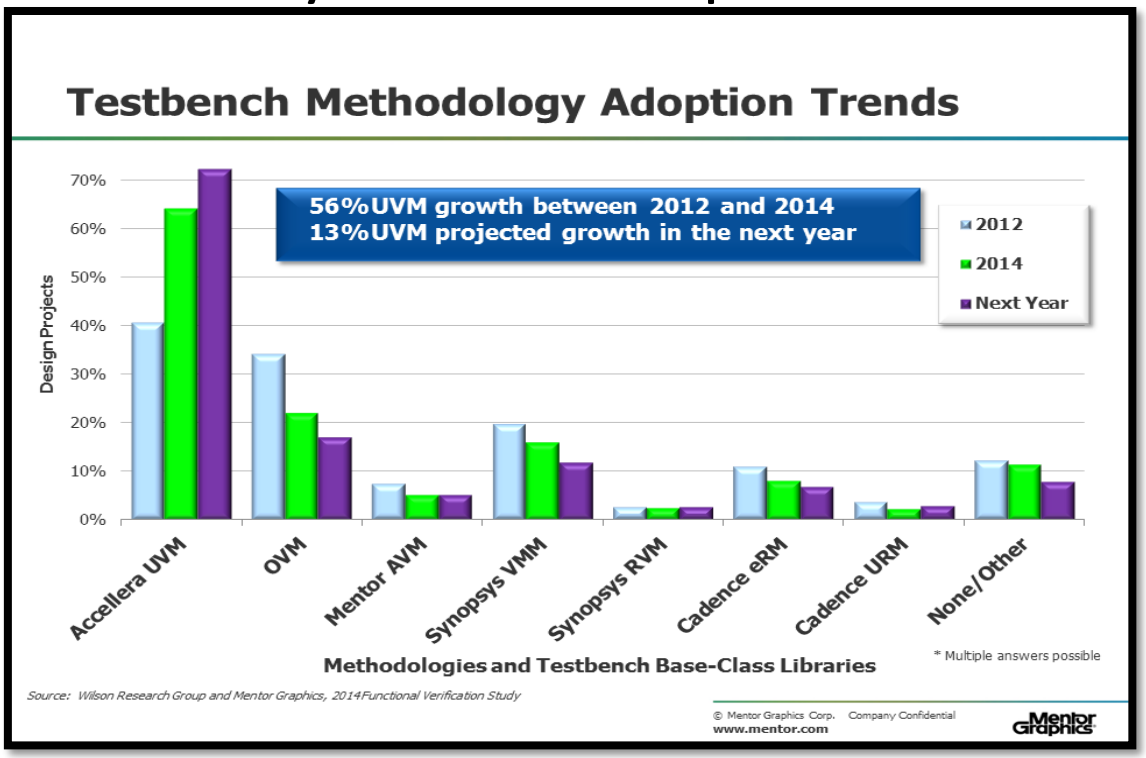


UVM journey so far

- UVM has come a long way
- Great framework
- Good block level stimulus features
- Well documented, proven to work
- Summary of decades of BKMs – Best Known Methods
- Continues to expand

UVM – fastest growing methodology

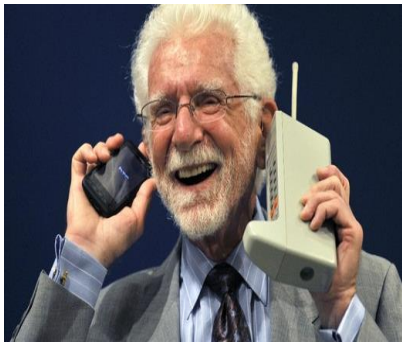
- Source: Independent survey by Wilson group
  - Sponsored by Mentor Graphics



Near future

- New markets, users for UVM
  - FPGA teams
  - RTL designers
  - Debug engineers
  - Assertion Developers, Formal Verification (counter examples)
  - Waveform-2-Stimulus tools
  - DFT vectors in simulation

Phone - Who is it for?



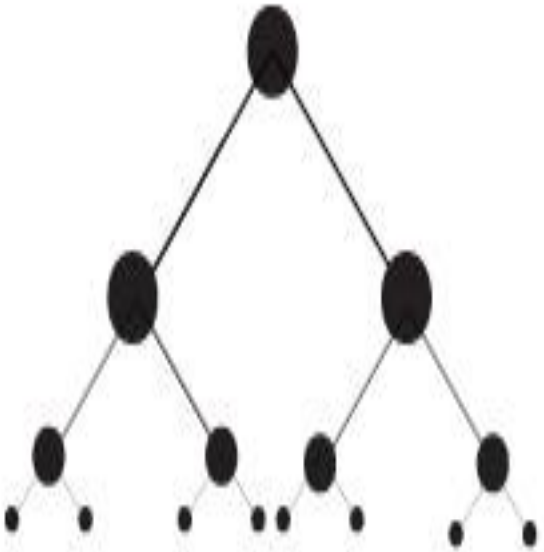
- Phones have gotten so complicated, so **hard to use**, that you wonder if this is designed for real people or for engineers.

Is UVM going the same route?

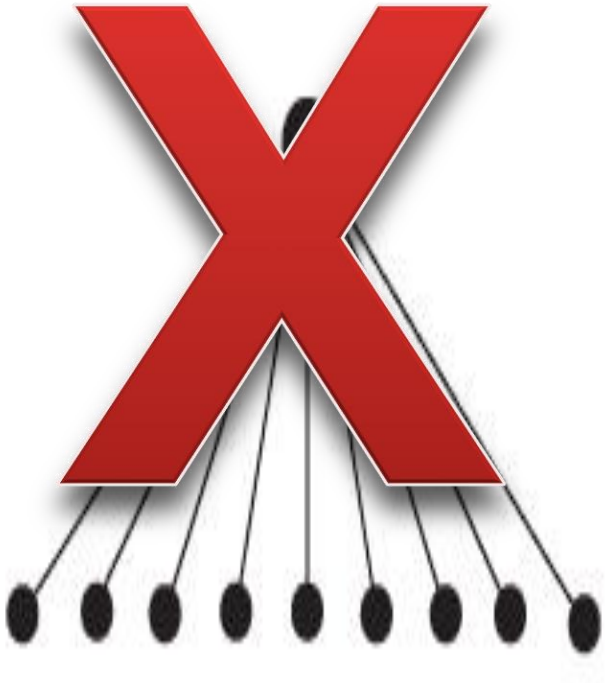
- UVM – **Universal Verification** Methodology
- Slowly becoming "only" for OOP savy, SW friendly Verification engineers
- Need to make it easy for RTL, FPGA, automation etc.

UVM is Complex, but..

COMPLEX



COMPLICATED



- Need not be complicated

VW's Go2UVM Package

- Our goals:
  - Provide simple framework for writing high quality simulation traces
  - Not to deviate from UVM approach
  - Introduce UVM to Verilog users
  - Make them future ready for more powerful UVM
- Simple package on top of standard UVM, released in open-source via http://www.go2uvm.org

VW_Go2UVM_pkg - Verilog vs. UVM stimulus

Verilog

up_down_cunter.sv + (~/Deskto

Edit Tools Syntax Buffers Win

```
task reset();
  $display("Start of reset");
  rst_n <= 1'b0;
  repeat(10) @(posedge clk);
  rst_n <= 1'b1;
  @(posedge clk);
  $display(" End of reset");
endtask : reset
```

UVM

tb_up_down_counter.sv + (~/tools/VWorks/VW_Go

Edit Tools Syntax Buffers Window Help

```
import uvm_pkg::*;
  `include "uvm_macros.svh"

import vw_go2uvm_pkg::*;


class go2uvm_count_test extends go2uvm_base_test;
  virtual count_if vif;

  task reset ();
    `uvm_info(log_id, "Start of reset", UVM_MEDIUM)
    vif.cb.rst_n <= 1'b0;
    repeat(10) @ (vif.cb);
    vif.cb.rst_n <= 1'b1;
    @ (vif.cb);
    `uvm_info(log_id, "End of reset", UVM_MEDIUM)
  endtask : reset
```

VW_Go2UVM_pkg - Verilog vs. UVM stimulus

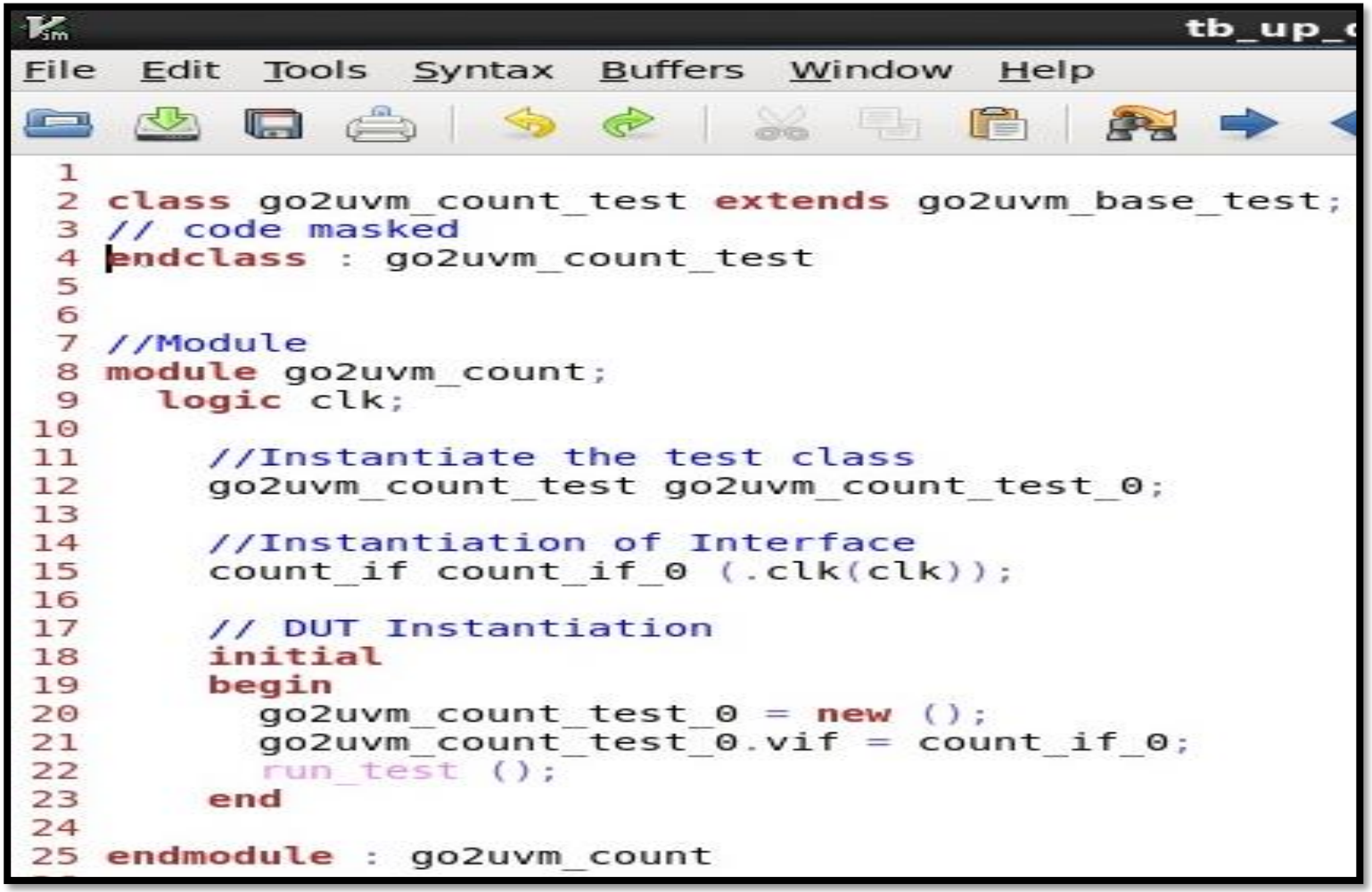
Verilog

```
task drive();
   int i;
  $display("Start of Test");
  @(posedge clk);
  load  <= 1'b1;

  repeat(2) @(posedge clk);
  load  <= 1'b1;
  data  <= 8'h78;

  repeat(2) @(posedge clk);
  load  <= 1'b1;
  cen   <= 1'b1;
  up_dn <= 1'b1;

  repeat(3) @(posedge clk);
  load  <= 1'b1;
  cen   <= 1'b1;
  up_dn <= 1'b0;
```

UVM

```
   `uvm_info(log_id, "End of reset", UVM_MEDIUM)
endtask : reset

task main();
   int i;
  `uvm_info(log_id, "Start of Test", UVM_MEDIUM)
   @ (vif.cb);
  vif.cb.rst_n <= 1'b1;
  vif.cb.load <= 1'b1;

  repeat(2) @ (vif.cb);
   vif.cb.rst_n <= 1'b1;
   vif.cb.load <= 1'b0;
   vif.cb.data <= 8'h78;

  repeat(2) @ (vif.cb);
   vif.cb.load <= 1'b1;
   vif.cb.cen <= 1'b1;
   vif.cb.up_dn <= 1'b1;

  repeat(3) @ (vif.cb);
   vif.cb.load <= 1'b1;
   vif.cb.cen <= 1'b1;
   vif.cb.up_dn <= 1'b0;
```

VW_Go2UVM_pkg - finishing touch..



Go2UVM sample log

- Same as standard UVM log format

```
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(217) @ 0: report
er [Questa UVM]  questa_uvm::init(+struct)
# UVM_INFO @ 0.00 ns: Go2UVM [Go2UVM] RELNOTES
#
# ----------------------------------------------------------------
# (C) 2004-2016 Verifworks a venture of CVC Pvt Ltd.
#  VW_Go2UVM Version 1.1d.5 (Verifworks http://www.verifworks.com )
#
#   ************       IMPORTANT RELEASE NOTES         *************
#
#   You are using a version of the Go2UVM Package from VerifWorks
#   a venture of CVC Pvt Ltd http://www.cvcblr.com
#   See http://www.verifworks.com for more details
#
# ----------------------------------------------------------------
#
# UVM_INFO @ 0.00 ns: reporter [RNTST] Running test ...
# UVM_INFO ../tb/tb_up_down_counter.sv(29) @ 0.00 ns: Go2UVM [Go2UVM] Start of r
eset
# UVM_INFO ../tb/tb_up_down_counter.sv(34) @ 105.00 ns: Go2UVM [Go2UVM] End of r
eset
# UVM_INFO @ 105.00 ns: Go2UVM [Go2UVM] Driving stimulus via UVM
# UVM_INFO ../tb/tb_up_down_counter.sv(39) @ 105.00 ns: Go2UVM [Go2UVM] Start of
 Test
```

How to use Go2UVM Pkg?

- Very simple use model
- Include 1 extra file to your list of files → Done!
  - vw_go2uvm_pkg.sv
- We ship examples and sample scripts
- Use env variable: $VW_GO2UVM_HOME
  - Pkg is under: $VW_GO2UVM_HOME/src
  - Examples: $VW_GO2UVM_HOME/examples
  - Docs: $VW_GO2UVM_HOME/docs

What do I get by using Go2UVM Pkg?

- Standard UVM framework
  - A minimal sub-set
- Messaging same as `uvm_info
- Test from ***uvm_test*** base class
- Phasing (Active)
  - Main Phase – mandatory
  - Reset Phase – optional, recommended
- Objections mechanism
  - Automated, users don't have to bother
- Test PASS/FAIL declaration
  - Automated

What's inside VW_Go2UVM Package?

- Wrapper around UVM Test layer
- Hides phasing completely from users
  - Uses ***main_phase*** – mandatory (User fills ***main()***)
  - ***reset_phase*** optional (User fills ***reset()***)
- Leverages on "pure virtual" to guard against misuse
- Internally handles objections
  - A must in UVM
  - Big time saver for first time UVM users

VW/CVC's role in the Verification industry

- Drove DVCon India 2014 from the front
- World-wide recognized Verification experts
  - Co-authors of 4 assertion books (Ajeetha & Srini)
- Coded VMM base classes in Synopsys
- UVMWorld.org contributions to UVM-Ref-Flow
- Delivered several training on UVM, OVM, VMM, SV, SVA, PSL
- Worked in evangelizing ABV, VMM, UVM across the world through several EDA companies
- Presented at various forums:
  - DVCon, SNUG-India, Taiwan, China, DVM (India) etc.
- Assisted standards development:
  - PSL, OVL, SVA, SV, UVM, eWG, VHDL, SV-AMS

How to access Go2UVM Package?

- Contact us via www.go2uvm.org
- Limited customer availability