Taking UVM to wider user base – The open-source way

Nagasundaram Thillaivasagam, VerifWorks

Guru Kinagi, CVC Pvt. Ltd.

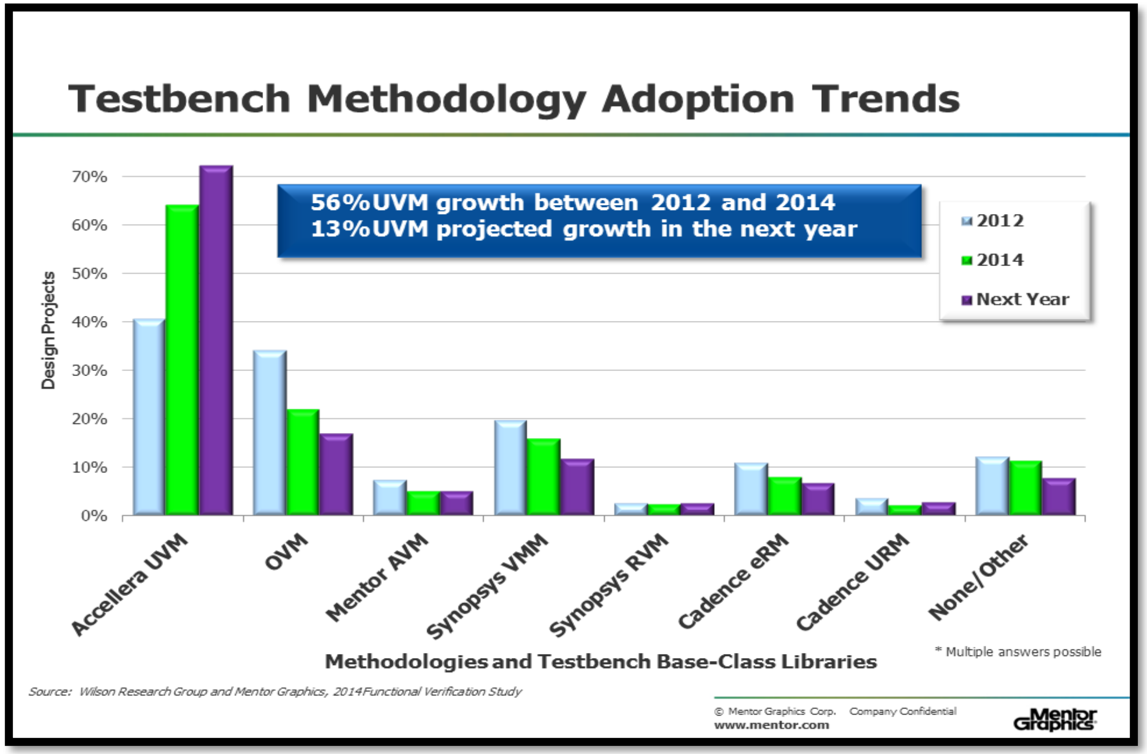
Santhosh Kumar, CVC Pvt. Ltd.

Agenda

- UVM Journey so far
- Need for Go2UVM
- Inside Go2UVM
- DVCreate Go2UVM open-source apps
- Conclusion

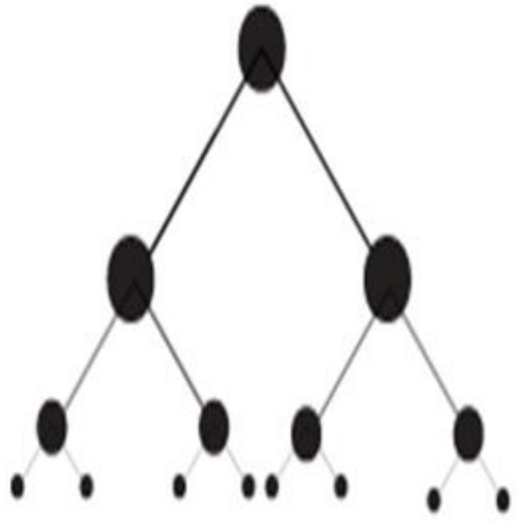
UVM – fastest growing methodology

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

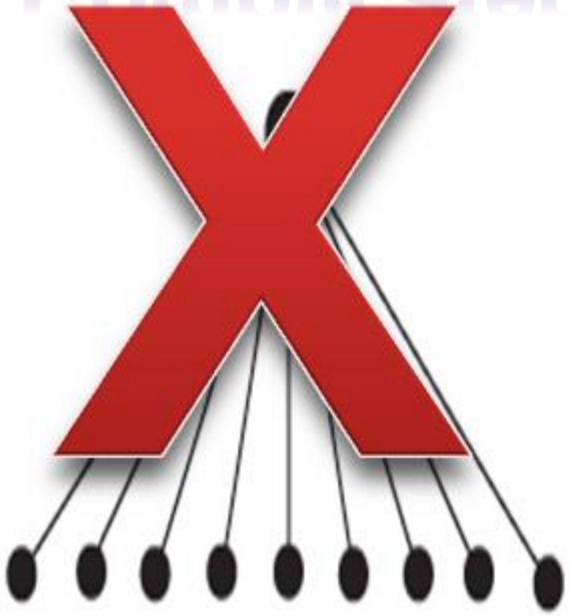


UVM is Complex, but..

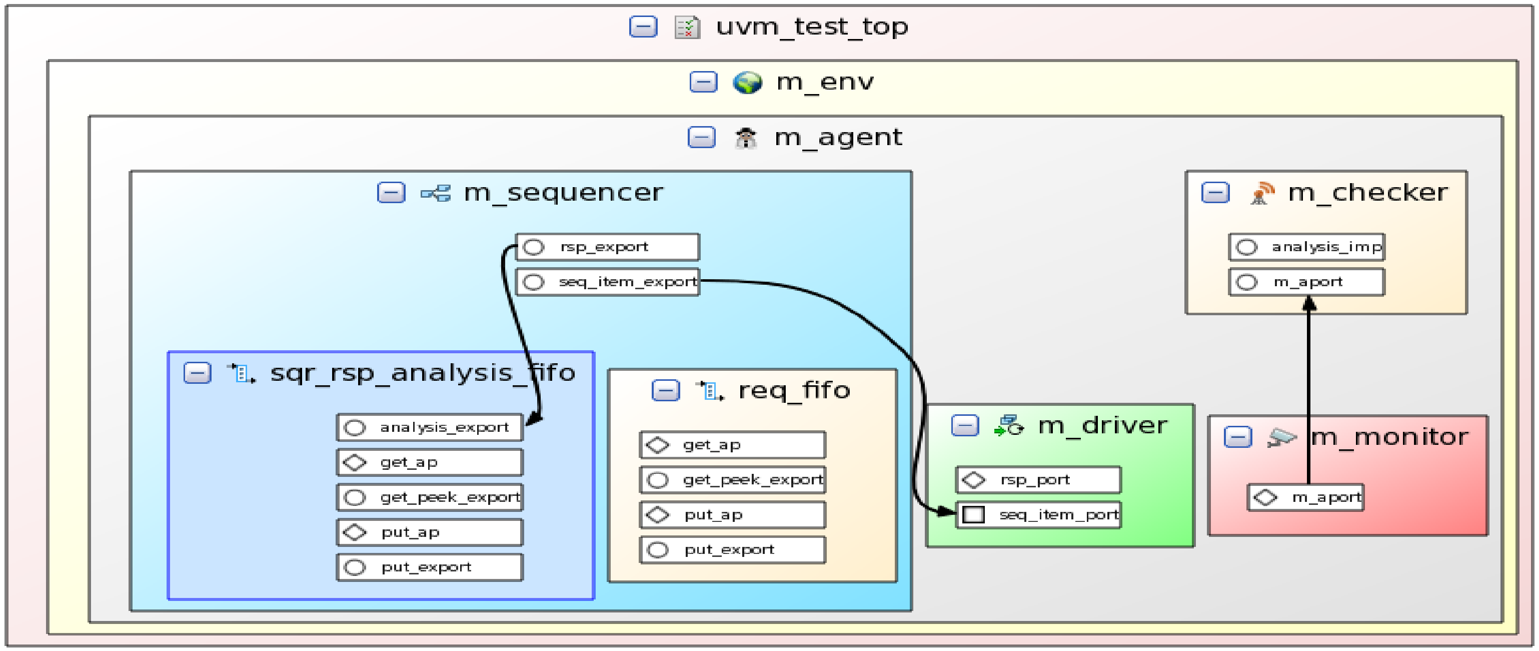
Complex

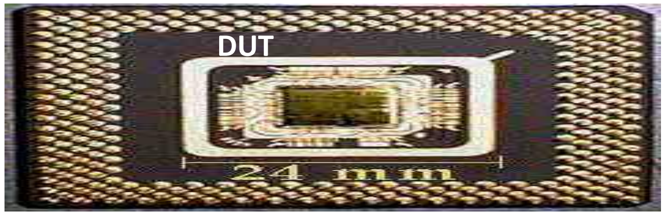


Need Not Be Complicated

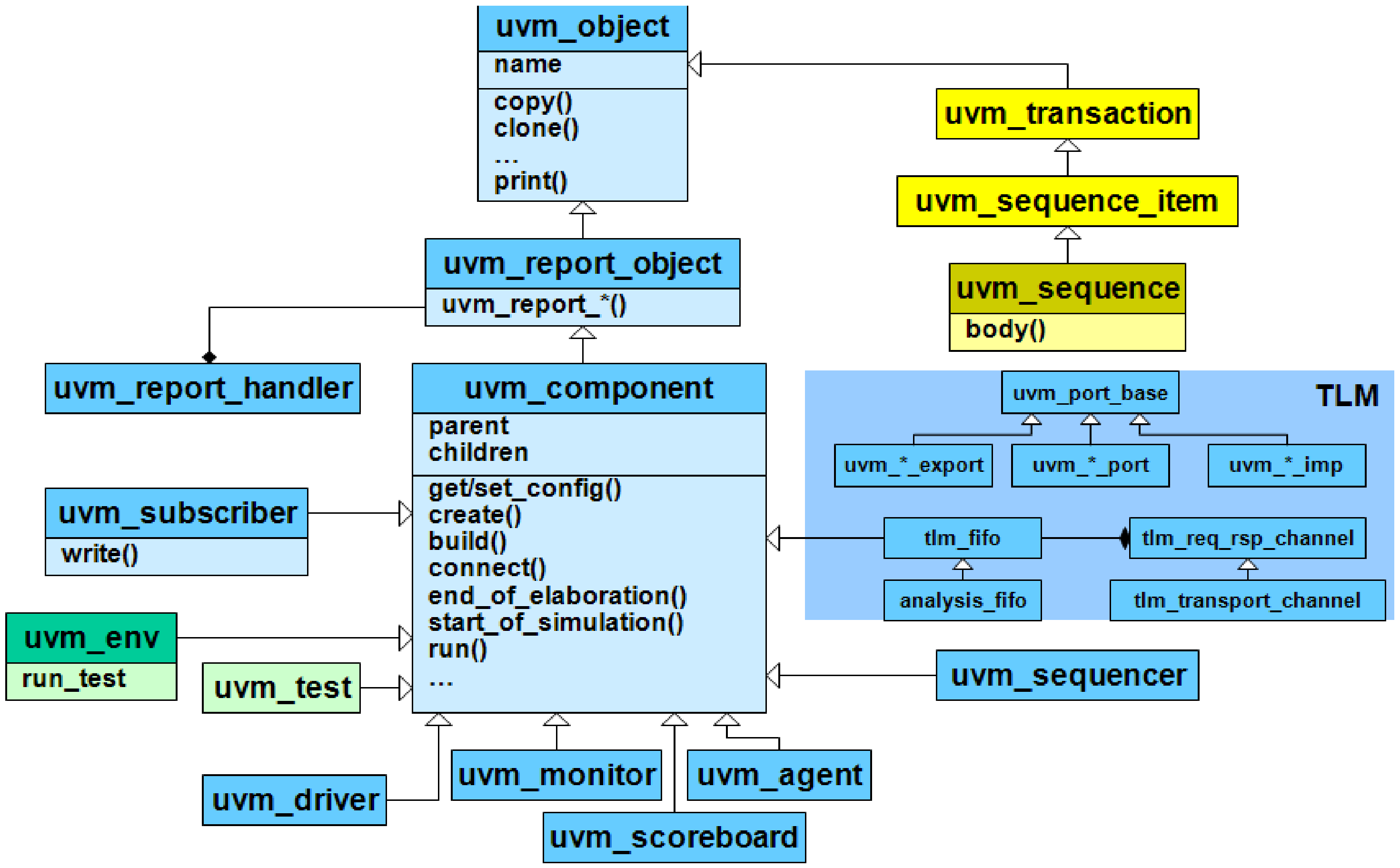


Typical UVM architecture

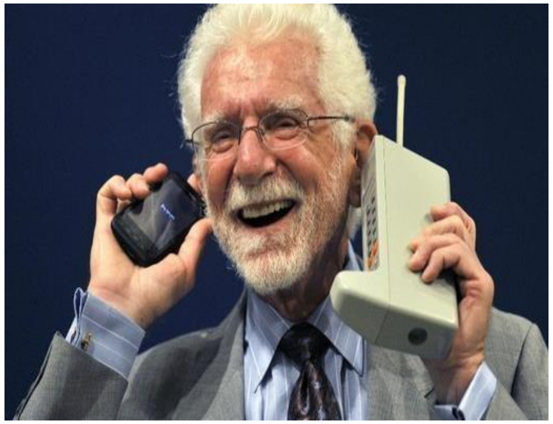




UVM evolution in UML



Motivation behind Go2UVM

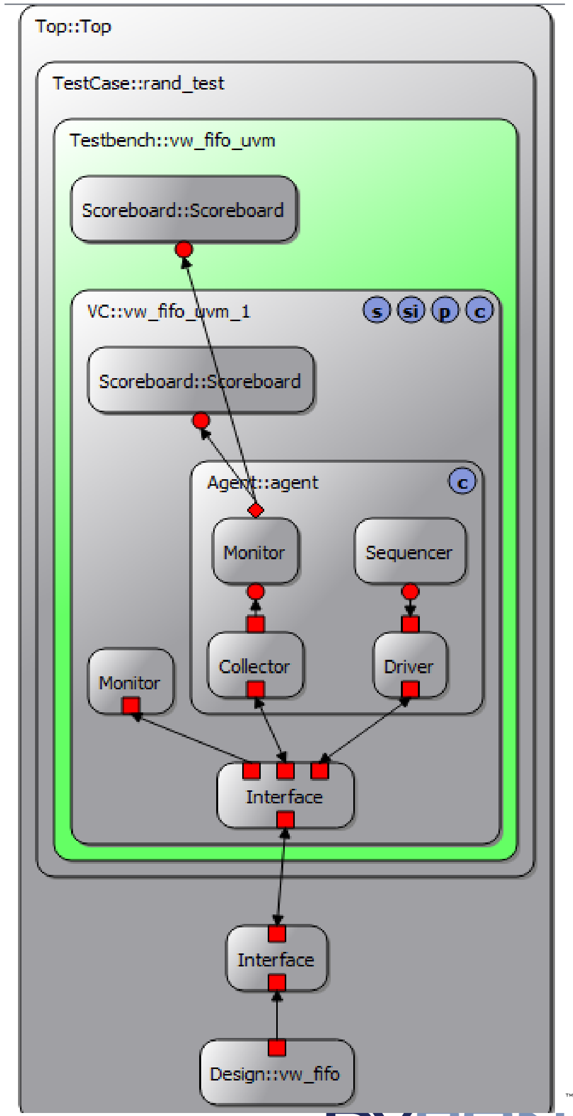


- Well, we knew that Someday everybody would have a [cell] phone
- Phones have gotten so complicated, so hard to use, that you wonder if this is designed for real people or for engineers.

UVM for verification similar?

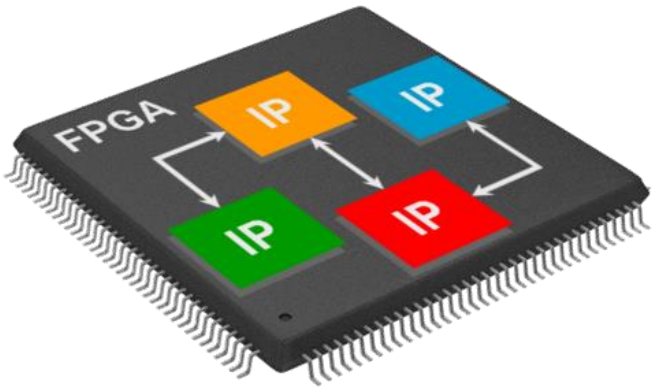
UVM mechanics

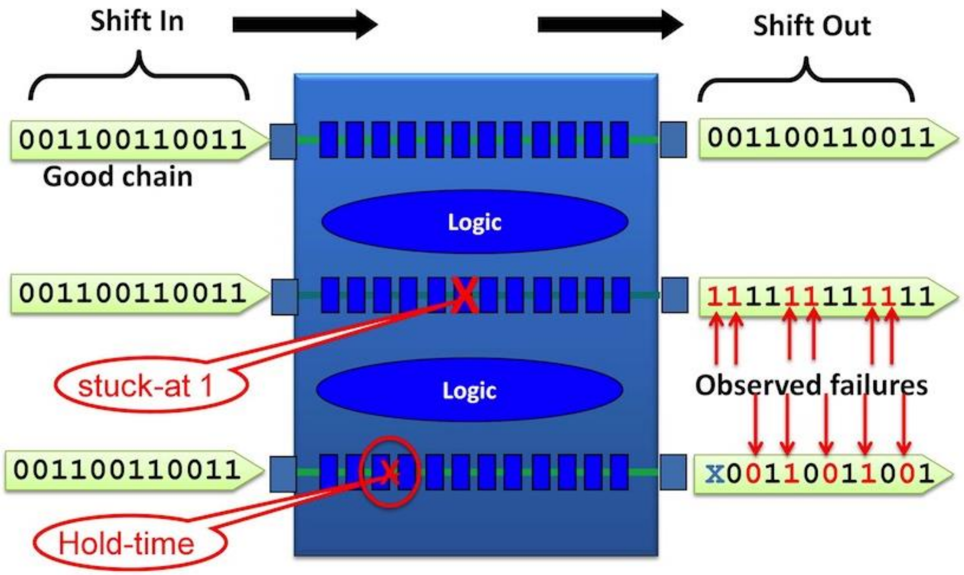
- Multiple layers of components
- Hierarchical component hook-ups
  - uvm_component::new (string name, uvm_component parent)
- UVM macros to automate lot of features
- Phasing (reset_phase, main_phase, run_phase etc.)
- Objection mechanism (A must in UVM to get even a simple stimulus through to the DUT)



Beyond ASIC verification

- UVM is widely used in ASIC DV projects
- Other DV domains:
  - FPGA
  - DFT
  - Unit Tests
  - Generated tests





Taking UVM beyond ASIC DV

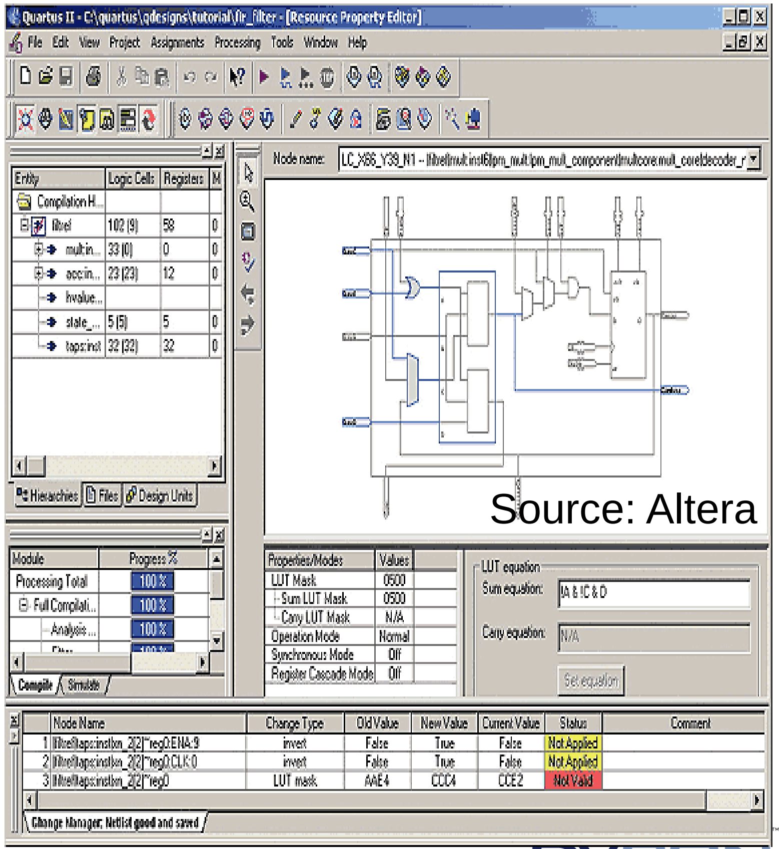
- Phones have gotten so complicated, so hard to use, that you wonder if this is designed for real people or for engineers.

- "***U***" in *UVM* reads "Universal"
- Wonder if it is only for software savvy, OOP fanatic engineers?
- Can it be used by many others who are more of hardware design engineers trying to get simple verification done.

Simplifying UVM adoption

- First time, Verilog test writers need simple test based stimuli
- Students/budding engineers need quick ramp to UVM

- IoT, FPGA design teams
  - No formal SV/UVM training
  - Smaller teams, often designers do Verification as well
  - More GUI based test/design authoring
- Auto-generated tests/traces
  - FSDB → Verilog Test/UVM Seq
- DFT patterns
  - No randomization, direct wire level stimulus



Entity	Logic Cells	Registers	M
Compilation H...			
filter	102 (9)	58	0
mult:in...	33 (0)	0	0
acc:in...	23 (23)	12	0
hvalue...			
state_...	5 (5)	5	0
taps:inst	32 (32)	32	0

Module	Progress %
Processing Total	100 %
Full Compilati...	100 %
Analysis ...	100 %

Properties/Modes	Values
LUT Mask	0500
Sum LUT Mask	0500
Carry LUT Mask	N/A
Operation Mode	Normal
Synchronous Mode	Off
Register Cascade Mode	Off

Sum equation: !A & !C & D

Carry equation: N/A

	Node Name	Change Type	Old Value	New Value	Current Value	Status	Comment
1	|filter|taps:inst|xn_2[2]~reg0:ENA:9	invert	False	True	False	Not Applied	
2	|filter|taps:inst|xn_2[2]~reg0:CLK:0	invert	False	True	False	Not Applied	
3	|filter|taps:inst|xn_2[2]~reg0	LUT mask	AAE4	CCC4	CCE2	Not Valid	

Change Manager: Netlist good and saved

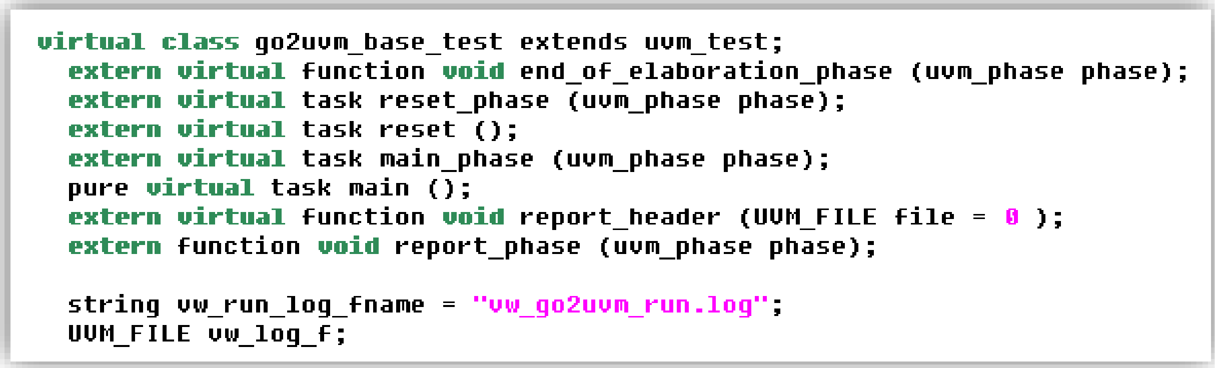
What is *Go2UVM*?

- SystemVerilog package
- TCL “apps” to auto-create Go2UVM files
- Package on top of Standard UVM framework
  - A minimal sub-set
- Messaging same as `uvm_info
  - Simplified versions available as well
- 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?

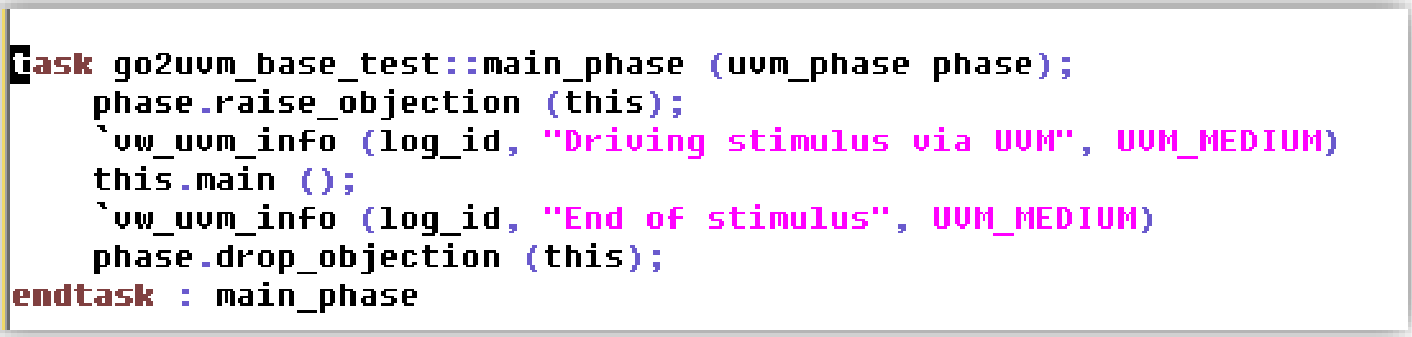
- Wrapper around UVM Test layer
- Hides phasing completely from users
  - Uses ***main_phase*** – mandatory (User fills ***main()***)
  - ***reset_phase*** optional (User fills ***reset()***)
- Leverages on "pure virtual" to guard against misuse
- Internally handles objections
  - A must in UVM
  - Big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

Inside Go2UVM



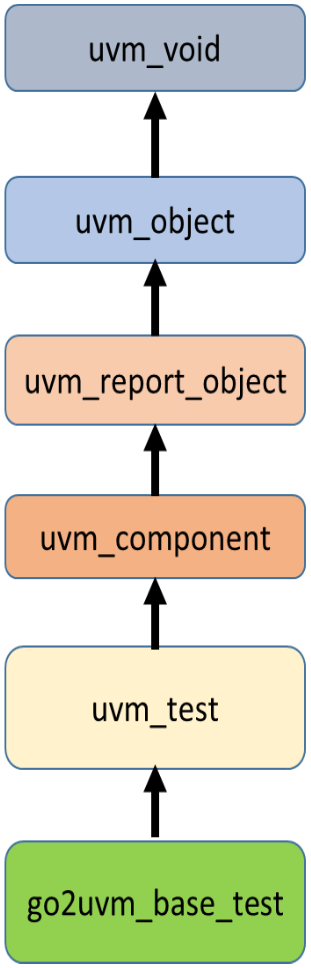
```
virtual class go2uvm_base_test extends uvm_test;
  extern virtual function void end_of_elaboration_phase (uvm_phase phase);
  extern virtual task reset_phase (uvm_phase phase);
  extern virtual task reset ();
  extern virtual task main_phase (uvm_phase phase);
  pure virtual task main ();
  extern virtual function void report_header (UVM_FILE file = 0 );
  extern function void report_phase (uvm_phase phase);

  string vw_run_log_fname = "vw_go2uvm_run.log";
  UVM_FILE vw_log_f;
```

Go2UVM base test



```
task go2uvm_base_test::main_phase (uvm_phase phase);
    phase.raise_objection (this);
    `vw_uvm_info (log_id, "Driving stimulus via UVM", UVM_MEDIUM)
    this.main ();
    `vw_uvm_info (log_id, "End of stimulus", UVM_MEDIUM)
    phase.drop_objection (this);
endtask : main_phase
```

Go2UVM main phase



VW’s Go2UVM Package

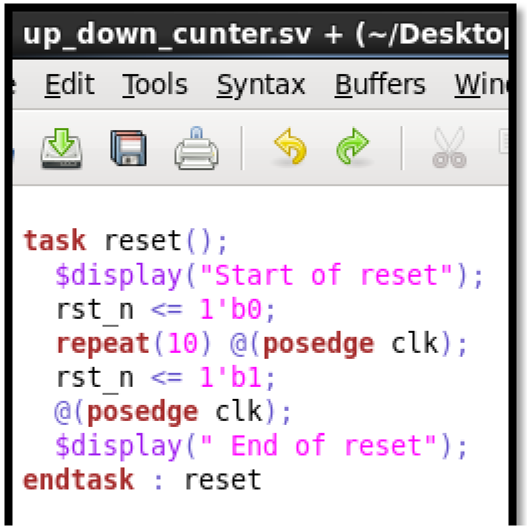
- Our goals:
  - Provide simple framework for writing high quality simulation traces
  - Not to deviate from UVM approach
  - Introduce UVM to Verilog users
  - Make them future ready for more powerful UVM
- Simple package on top of standard UVM, released in open-source via http://www.go2uvm.org

Hookup – DUT, Interface & UVM

- Very similar to Verilog flow
  - Create module tb_top
  - Instantiate DUT
  - Instantiate interface
  - Generate clock
- Go2UVM
  - Instantiate *UVM test*
  - Call run_test ()
- All of the above is automated via TCL "apps"

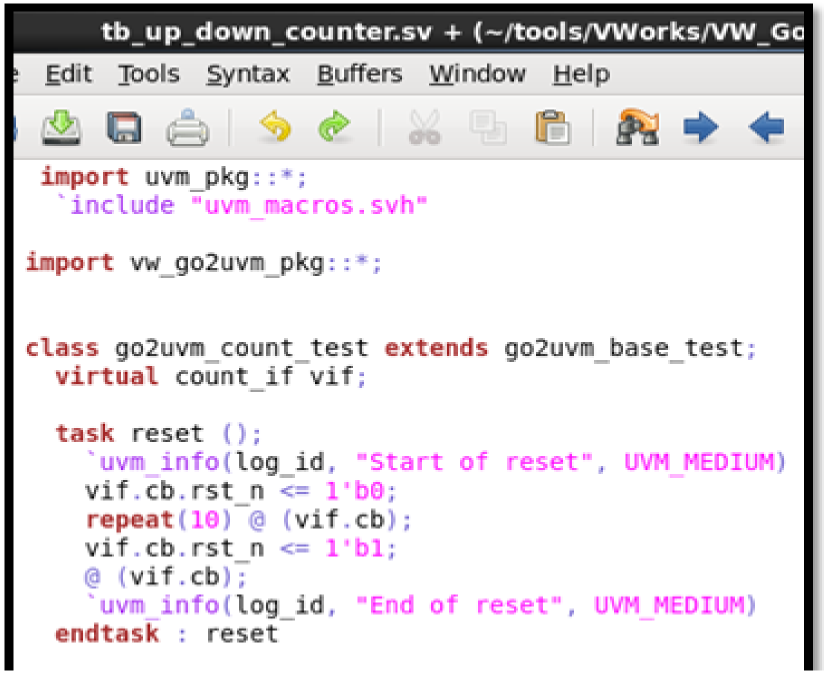
VW_Go2UVM_pkg - Verilog vs. UVM stimulus

Verilog



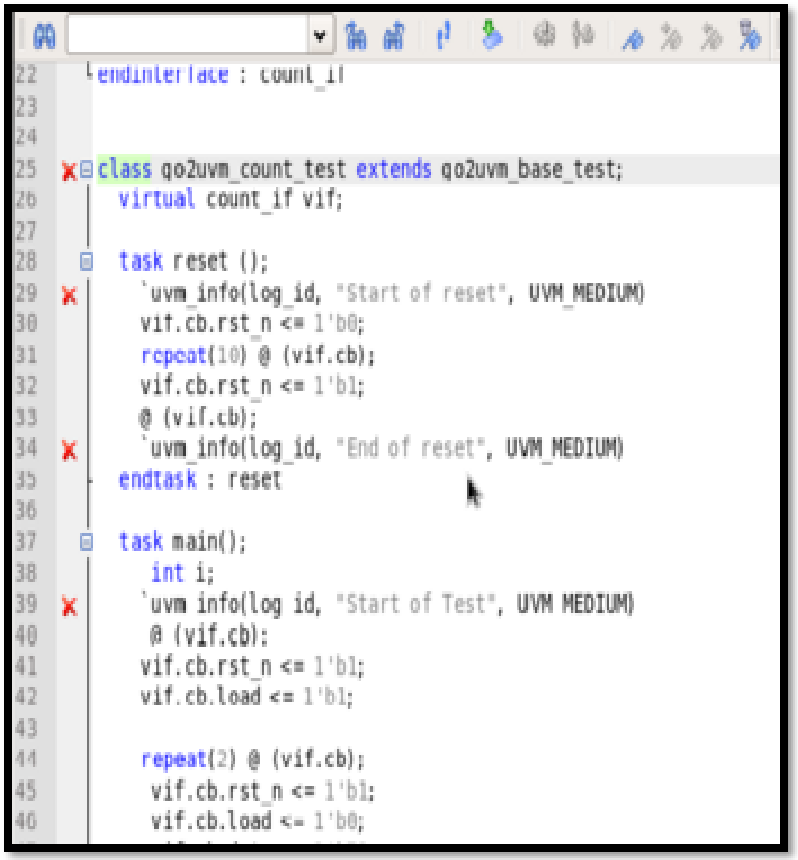
```
task reset();
  $display("Start of reset");
  rst_n <= 1'b0;
  repeat(10) @(posedge clk);
  rst_n <= 1'b1;
  @(posedge clk);
  $display(" End of reset");
endtask : reset
```

UVM

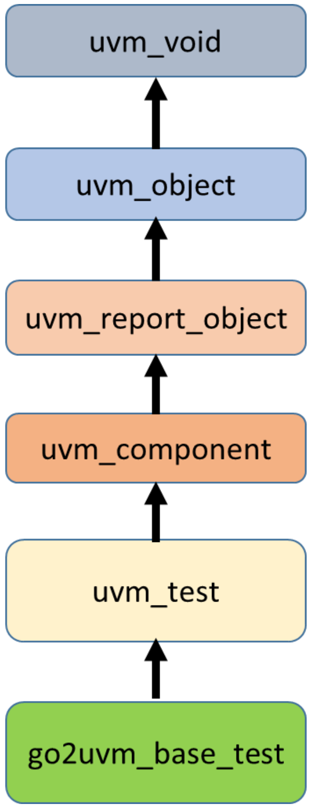

tb_up_down_counter.sv + (~/tools/VWorks/VW_Go
Edit Tools Syntax Buffers Window Help

```
 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
```

First look at Go2UVM test

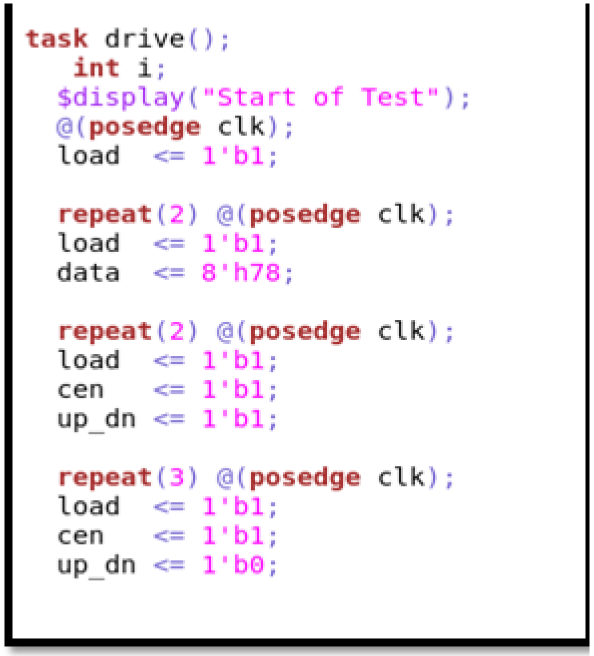


```
endinterface : count_if


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

  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;
```



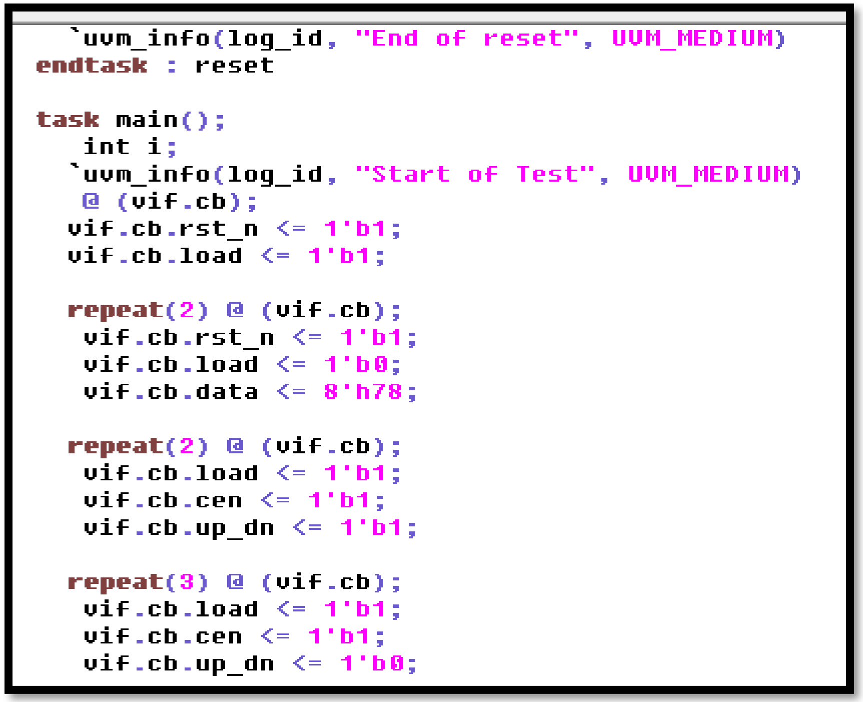
VW_Go2UVM_pkg - Verilog vs. UVM stimulus

Verilog



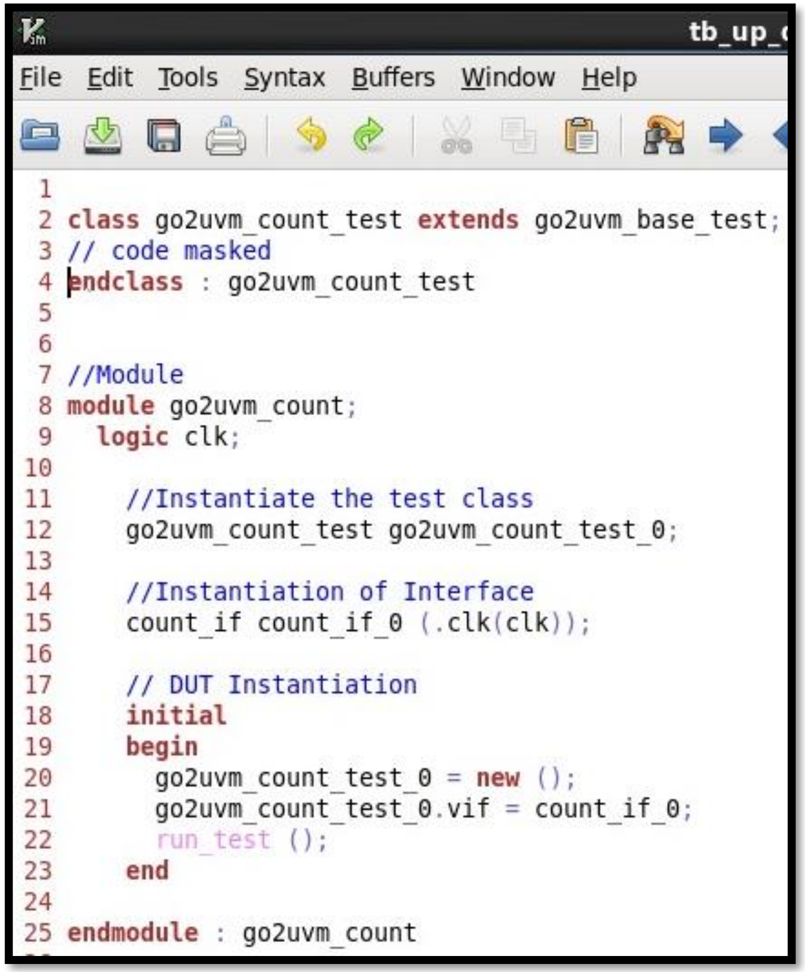
```
task drive();
   int i;
  $display("Start of Test");
  @(posedge clk);
  load   <= 1'b1;

  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..

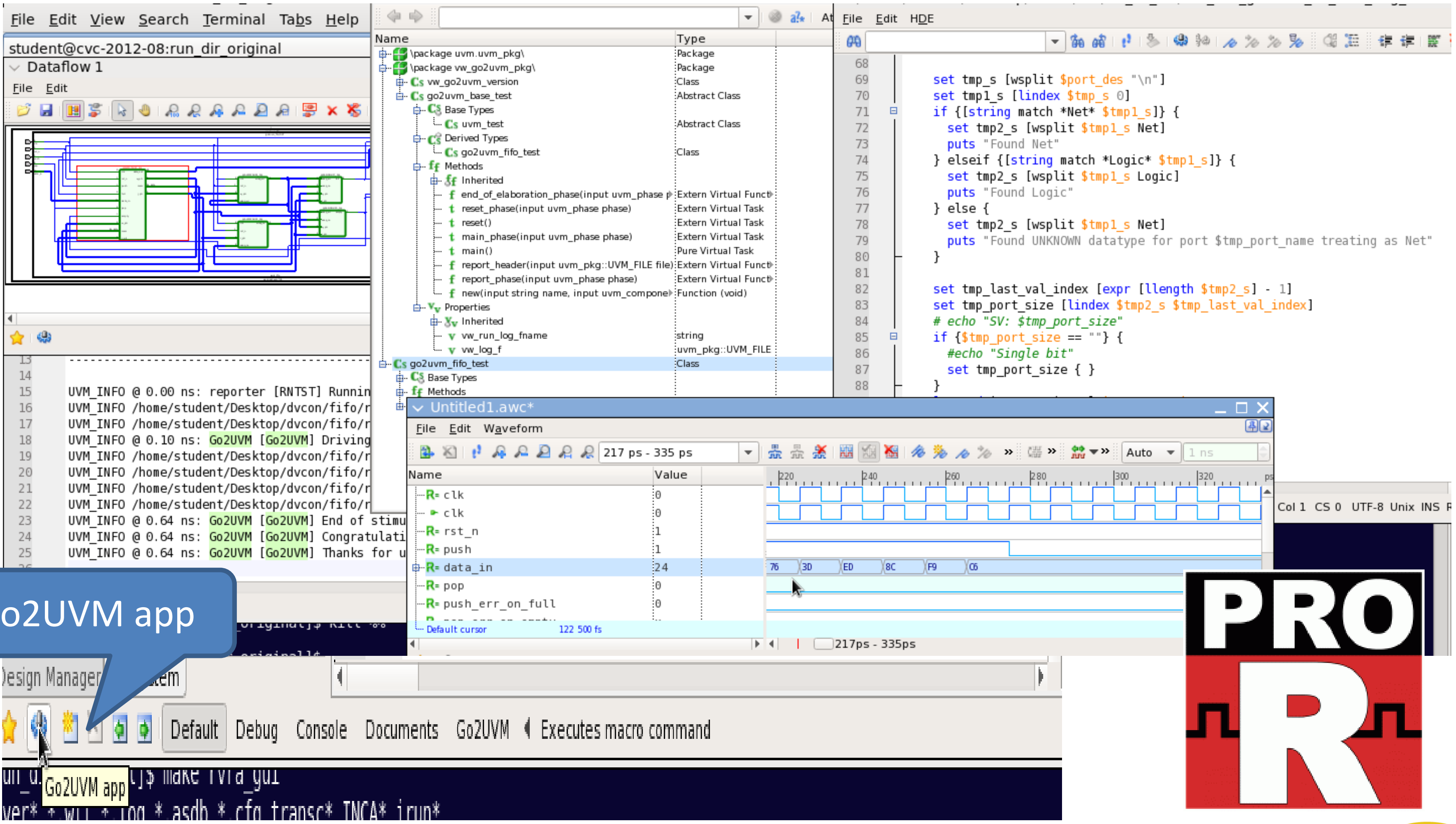


```

class go2uvm_count_test extends go2uvm_base_test;
// code masked
endclass : go2uvm_count_test


//Module
module go2uvm_count;
  logic clk;

    //Instantiate the test class
    go2uvm_count_test go2uvm_count_test_0;

    //Instantiation of Interface
    count_if count_if_0 (.clk(clk));

    // DUT Instantiation
    initial
    begin
      go2uvm_count_test_0 = new ();
      go2uvm_count_test_0.vif = count_if_0;
      run_test ();
    end

endmodule : go2uvm_count
```

TCL Apps – DVC_Go2UVM

- Open-source "apps" to generate Go2UVM files for a given RTL
- Works with all major EDA tools:
  - Aldec, Riviera-PRO
  - Cadence, IUS
  - Mentor Graphics, Questa
  - Synopsys, Verdi
- Given RTL top, the app generates:
  - SystemVerilog interface
  - Full go2uvm package (source code)
  - A template test that extends go2uvm_base_test
  - Scripts to compile and run on ALL major EDA tools

Aldec Riviera-PRO + Go2UVM app


File Edit View Search Terminal Tabs Help
student@cvc-2012-08:run_dir_original
Dataflow 1
File Edit
File Edit HDE
Untitled1.awc*
File Edit Waveform
Go2UVM app
Design Manager
Default Debug Console Documents Go2UVM Executes macro command
Go2UVM app
Col 1 CS 0 UTF-8 Unix INS

```
--------------------------------------------

UVM_INFO @ 0.00 ns: reporter [RNTST] Runnin
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO @ 0.10 ns: Go2UVM [Go2UVM] Driving
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO /home/student/Desktop/dvcon/fifo/r
UVM_INFO @ 0.64 ns: Go2UVM [Go2UVM] End of stimu
UVM_INFO @ 0.64 ns: Go2UVM [Go2UVM] Congratulati
UVM_INFO @ 0.64 ns: Go2UVM [Go2UVM] Thanks for u
```

Name	Type
\package uvm.uvm_pkg\	Package
\package vw_go2uvm_pkg\	Package
vw_go2uvm_version	Class
go2uvm_base_test	Abstract Class
Base Types	
uvm_test	Abstract Class
Derived Types	
go2uvm_fifo_test	Class
Methods	
Inherited	
end_of_elaboration_phase(input uvm_phase p	Extern Virtual Funct
reset_phase(input uvm_phase phase)	Extern Virtual Task
reset()	Extern Virtual Task
main_phase(input uvm_phase phase)	Extern Virtual Task
main()	Pure Virtual Task
report_header(input uvm_pkg::UVM_FILE file)	Extern Virtual Funct
report_phase(input uvm_phase phase)	Extern Virtual Funct
new(input string name, input uvm_compone	Function (void)
Properties	
Inherited	
vw_run_log_fname	string
vw_log_f	uvm_pkg::UVM_FILE
go2uvm_fifo_test	Class
Base Types	
Methods	

```
    set tmp_s [wsplit $port_des "\n"]
    set tmp1_s [lindex $tmp_s 0]
    if {[string match *Net* $tmp1_s]} {
      set tmp2_s [wsplit $tmp1_s Net]
      puts "Found Net"
    } elseif {[string match *Logic* $tmp1_s]} {
      set tmp2_s [wsplit $tmp1_s Logic]
      puts "Found Logic"
    } else {
      set tmp2_s [wsplit $tmp1_s Net]
      puts "Found UNKNOWN datatype for port $tmp_port_name treating as Net"
    }

    set tmp_last_val_index [expr [llength $tmp2_s] - 1]
    set tmp_port_size [lindex $tmp2_s $tmp_last_val_index]
    # echo "SV: $tmp_port_size"
    if {$tmp_port_size == ""} {
      #echo "Single bit"
      set tmp_port_size { }
    }
```

217 ps - 335 ps

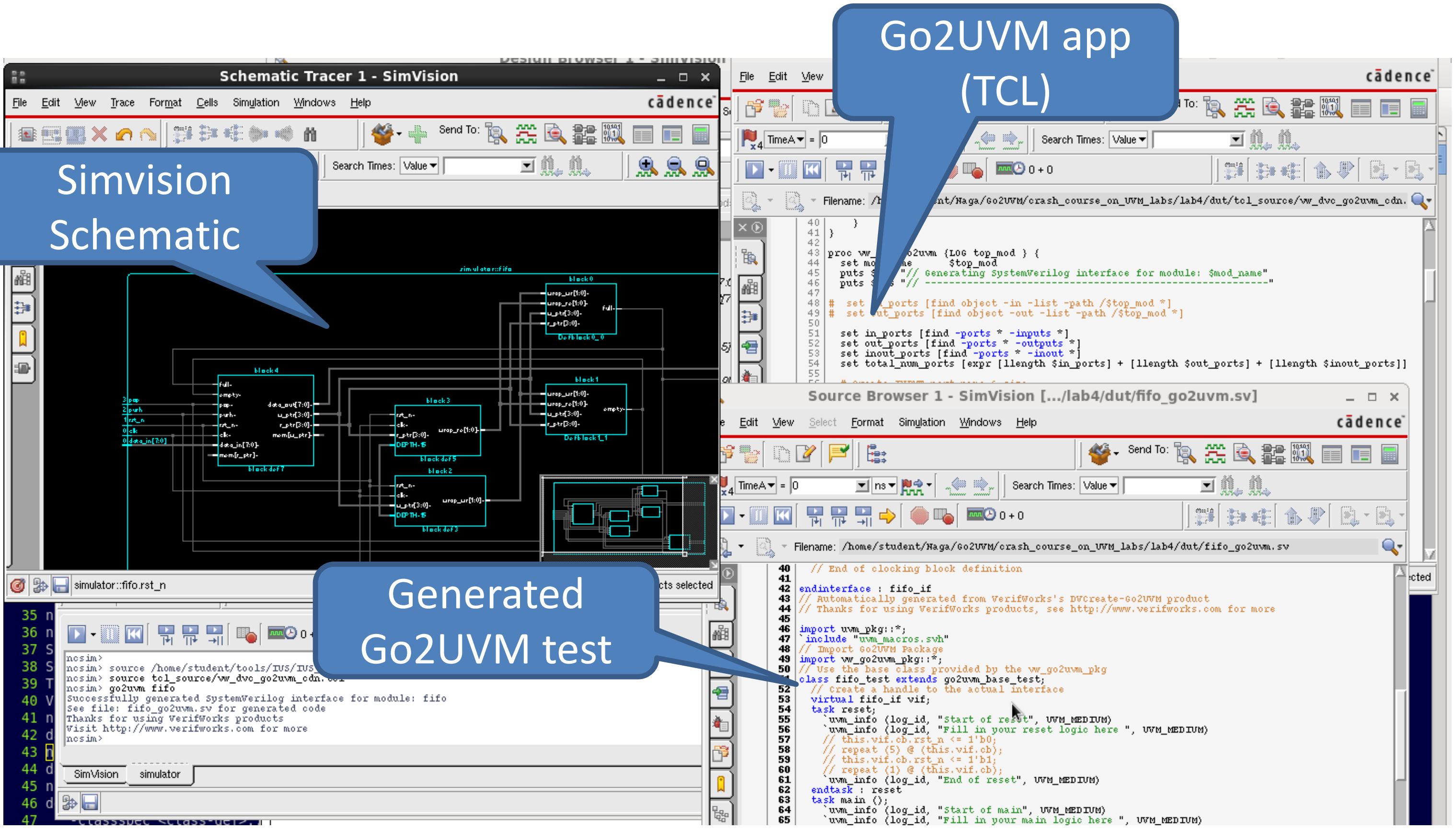
Name	Value
clk	0
clk	0
rst_n	1
push	1
data_in	24
pop	0
push_err_on_full	0
Default cursor	122 500 fs

76 3D ED 8C F9 C6

217ps - 335ps

```
make rvra_gui
ver* * wlf * log * asdb * cfg transc* INCA* irun*
```

Cadence IUS + Go2UVM app



Filename: /h…nt/Naga/Go2UVM/crash_course_on_UVM_labs/lab4/dut/tcl_source/vw_dvc_go2uvm_cdn.

```
    }
}

proc vw_      o2uvm {LOG top_mod } {
  set mo      me        $top_mod
  puts $      "// Generating SystemVerilog interface for module: $mod_name"
  puts $    s "// -----------------------------------------------------------"

#  set     _ports [find object -in -list -path /$top_mod *]
#  set    ut_ports [find object -out -list -path /$top_mod *]

  set in_ports [find -ports * -inputs *]
  set out_ports [find -ports * -outputs *]
  set inout_ports [find -ports * -inout *]
  set total_num_ports [expr [llength $in_ports] + [llength $out_ports] + [llength $inout_ports]]
```

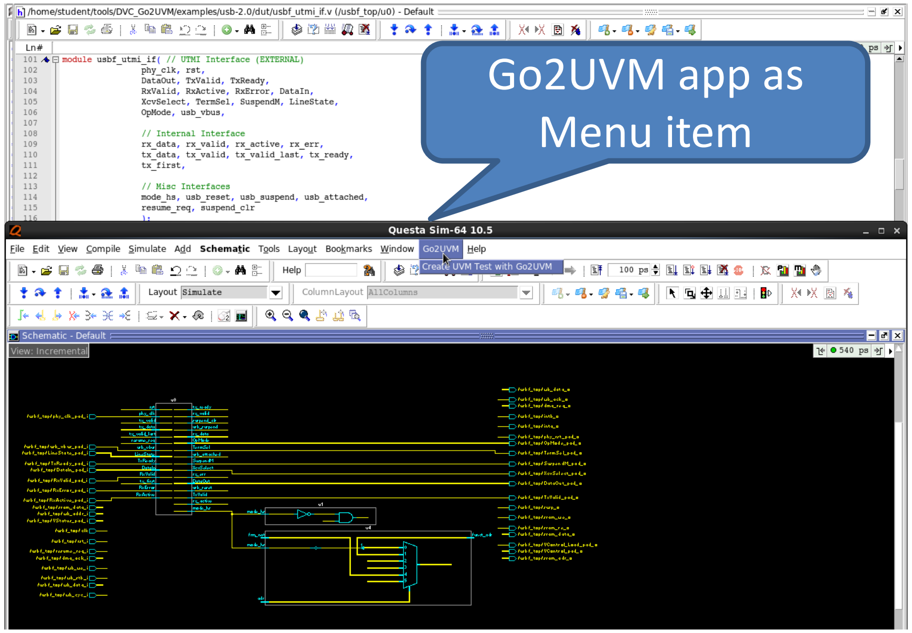
Filename: /home/student/Naga/Go2UVM/crash_course_on_UVM_labs/lab4/dut/fifo_go2uvm.sv

```
  // End of clocking block definition

endinterface : fifo_if
// Automatically generated from VerifWorks's DVCreate-Go2UVM product
// Thanks for using VerifWorks products, see http://www.verifworks.com for more

import uvm_pkg::*;
`include "uvm_macros.svh"
// Import Go2UVM Package
import vw_go2uvm_pkg::*;
// Use the base class provided by the vw_go2uvm_pkg
class fifo_test extends go2uvm_base_test;
  // Create a handle to the actual interface
  virtual fifo_if vif;
  task reset;
    `uvm_info (log_id, "Start of reset", UVM_MEDIUM)
    `uvm_info (log_id, "Fill in your reset logic here ", UVM_MEDIUM)
    // this.vif.cb.rst_n <= 1'b0;
    // repeat (5) @ (this.vif.cb);
    // this.vif.cb.rst_n <= 1'b1;
    // repeat (1) @ (this.vif.cb);
    `uvm_info (log_id, "End of reset", UVM_MEDIUM)
  endtask : reset
  task main ();
    `uvm_info (log_id, "Start of main", UVM_MEDIUM)
    `uvm_info (log_id, "Fill in your main logic here ", UVM_MEDIUM)
```

```
ncsim>
ncsim> source /home/student/tools/IUS/IUS
ncsim> source tcl_source/vw_dvc_go2uvm_cdn.
ncsim> go2uvm fifo
Successfully generated SystemVerilog interface for module: fifo
See file: fifo_go2uvm.sv for generated code
Thanks for using VerifWorks products
Visit http://www.verifworks.com for more
ncsim>
```

Questa + Go2UVM app

- Mentor Graphics's Questa has several interfaces
  - VPI
  - TCL
- Go2UVM app is integrated with Questa GUI


/home/student/tools/DVC_Go2UVM/examples/usb-2.0/dut/usbf_utmi_if.v (/usbf_top/u0) - Default
Go2UVM app as Menu item
Questa Sim-64 10.5
File Edit View Compile Simulate Add Schematic Tools Layout Bookmarks Window Go2UVM Help
Create UVM Test with Go2UVM
Layout Simulate
ColumnLayout AllColumns
Schematic - Default
View: Incremental

```
module usbf_utmi_if( // UTMI Interface (EXTERNAL)
                phy_clk, rst,
                DataOut, TxValid, TxReady,
                RxValid, RxActive, RxError, DataIn,
                XcvSelect, TermSel, SuspendM, LineState,
                OpMode, usb_vbus,

                // Internal Interface
                rx_data, rx_valid, rx_active, rx_err,
                tx_data, tx_valid, tx_valid_last, tx_ready,
                tx_first,

                // Misc Interfaces
                mode_hs, usb_reset, usb_suspend, usb_attached,
                resume_req, suspend_clr
                );
```

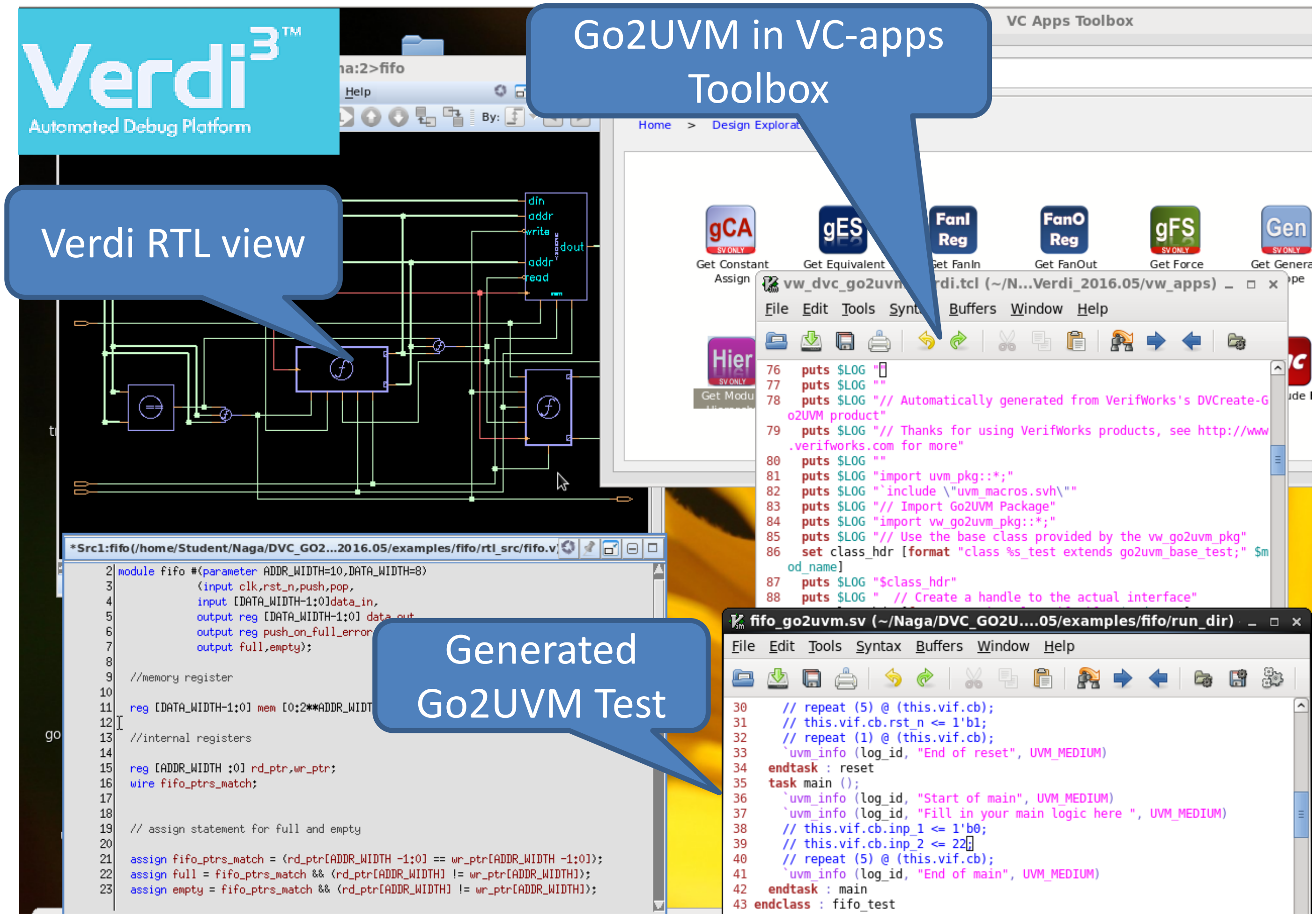
SV Interface

Go2UVM pkg

Go2UVM test template

Makefile

Verdi + Go2UVM app (VC Apps)


Go2UVM in VC-apps Toolbox
Verdi RTL view
Generated Go2UVM Test
Verdi3™ Automated Debug Platform
VC Apps Toolbox
Home > Design Explorat
Get Constant Assign
Get Equivalent
Get FanIn
Get FanOut
Get Force
Get Genera
*Src1:fifo(/home/Student/Naga/DVC_GO2...2016.05/examples/fifo/rtl_src/fifo.v)
vw_dvc_go2uvm...rdi.tcl (~/N...Verdi_2016.05/vw_apps)
fifo_go2uvm.sv (~/Naga/DVC_GO2U....05/examples/fifo/run_dir)
File Edit Tools Syntax Buffers Window Help

```
module fifo #(parameter ADDR_WIDTH=10,DATA_WIDTH=8)
            (input clk,rst_n,push,pop,
            input [DATA_WIDTH-1:0]data_in,
            output reg [DATA_WIDTH-1:0] data_out
            output reg push_on_full_error
            output full,empty);

 //memory register

 reg [DATA_WIDTH-1:0] mem [0:2**ADDR_WIDT

 //internal registers

 reg [ADDR_WIDTH :0] rd_ptr,wr_ptr;
 wire fifo_ptrs_match;


 // assign statement for full and empty

 assign fifo_ptrs_match = (rd_ptr[ADDR_WIDTH -1:0] == wr_ptr[ADDR_WIDTH -1:0]);
 assign full = fifo_ptrs_match && (rd_ptr[ADDR_WIDTH] != wr_ptr[ADDR_WIDTH]);
 assign empty = fifo_ptrs_match && (rd_ptr[ADDR_WIDTH] != wr_ptr[ADDR_WIDTH]);
```

```
  puts $LOG ""
  puts $LOG ""
  puts $LOG "// Automatically generated from VerifWorks's DVCreate-Go2UVM product"
  puts $LOG "// Thanks for using VerifWorks products, see http://www.verifworks.com for more"
  puts $LOG ""
  puts $LOG "import uvm_pkg::*;"
  puts $LOG "`include \"uvm_macros.svh\""
  puts $LOG "// Import Go2UVM Package"
  puts $LOG "import vw_go2uvm_pkg::*;"
  puts $LOG "// Use the base class provided by the vw_go2uvm_pkg"
  set class_hdr [format "class %s_test extends go2uvm_base_test;" $mod_name]
  puts $LOG "$class_hdr"
  puts $LOG "  // Create a handle to the actual interface"
```

```
    // repeat (5) @ (this.vif.cb);
    // this.vif.cb.rst_n <= 1'b1;
    // repeat (1) @ (this.vif.cb);
    `uvm_info (log_id, "End of reset", UVM_MEDIUM)
  endtask : reset
  task main ();
    `uvm_info (log_id, "Start of main", UVM_MEDIUM)
    `uvm_info (log_id, "Fill in your main logic here ", UVM_MEDIUM)
    // this.vif.cb.inp_1 <= 1'b0;
    // this.vif.cb.inp_2 <= 22;
    // repeat (5) @ (this.vif.cb);
    `uvm_info (log_id, "End of main", UVM_MEDIUM)
  endtask : main
endclass : fifo_test
```

More artillery to GO-2-UVM

- Latest Go2UVM package
- TCL apps to generate Go2UVM for given RTL
- DVC_UVM app to generate full UVM bench for an IP
- Tons of examples
- Free training for Academia across India
  - http://www.go2uvm.org/resources/free-go2uvm-training/
- SVA Book examples with Go2UVM traces
  - Assertions are great candidates for Unit testing
  - Refer to our DVCon India paper on: “Verify thy Verifier”

Conclusion and looking forward

- VW's Go2UVM Package is really simple to use.
- Enables quick start to UVM for Verilog users
- *Go2UVM* is 100% IEEE 1800 and IEEE P1800.2 compatible, users can easily start with *Go2UVM* and move to a full-fledged UVM environment
- *Go2UVM* associated "apps" it is set to reach those users hitherto untouched owing to the complexity of UVM for first-time users.

Questions…?