2017 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

UVM Interactive Debug Library: Shortening the Debug Turnaround Time

Horace Chan, Microsemi Corporation

Why Interactive Debug?

- SystemVerilog is **SLOW** to debug
  - Primitive peek/poke/force only
  - No user inputs
  - Recompile, recompile, recompile

- Specman/Cocotb(Python) is **FAST** to debug
  - Call any testbench function
  - Read or write registers
  - Start or kill sequences
  - No recompile

Setup the uvm_debug library

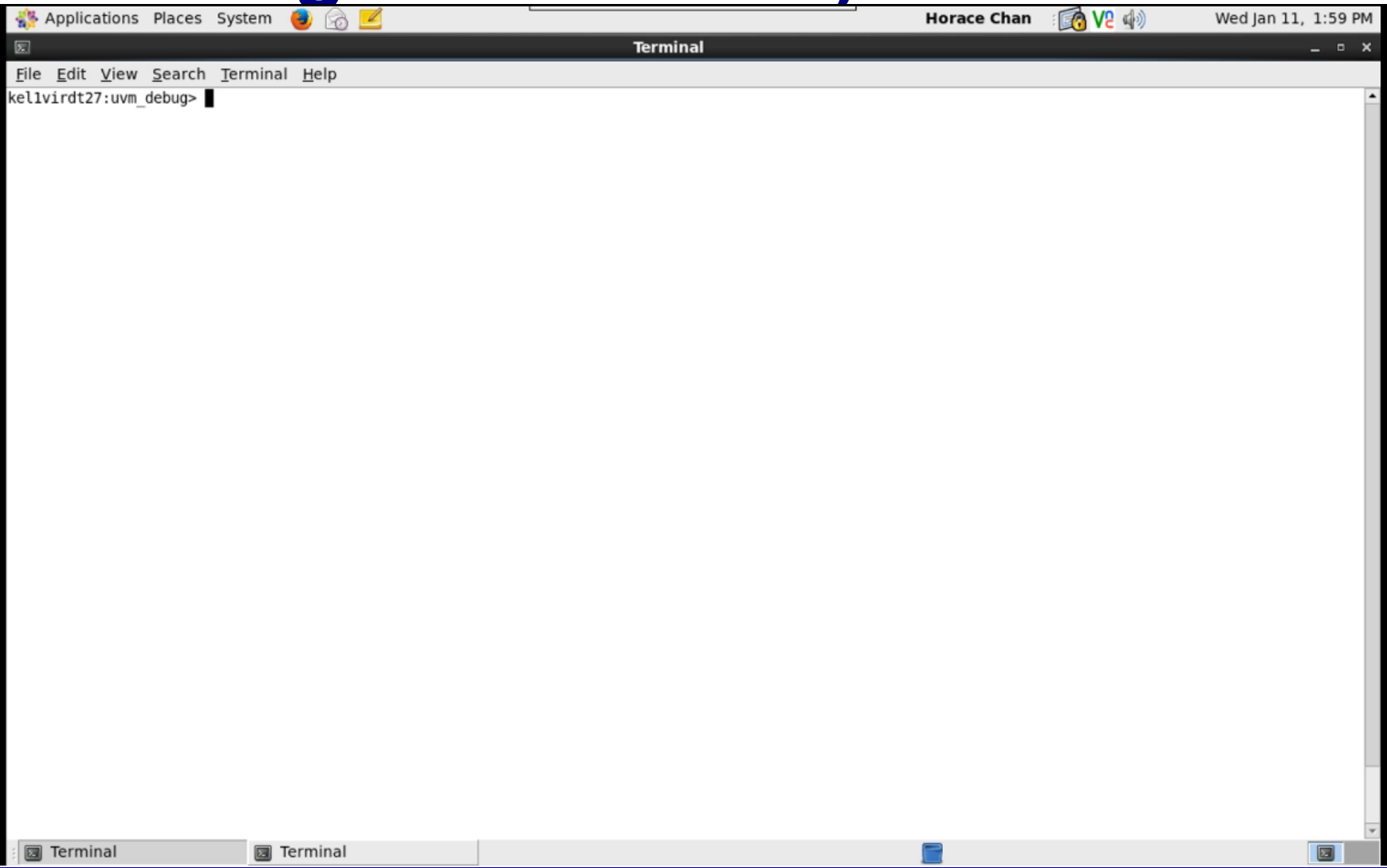
- Easy to setup, non-intrusive
  - In SV, add this to your testbench

```
import uvm_debug_pkg::*;
...
uvm_debug_util uvm_debug = uvm_debug_util::get();
uvm_debug.reg_util.set_top(my_top_reg_block);
uvm_debug.prompt(1);
```

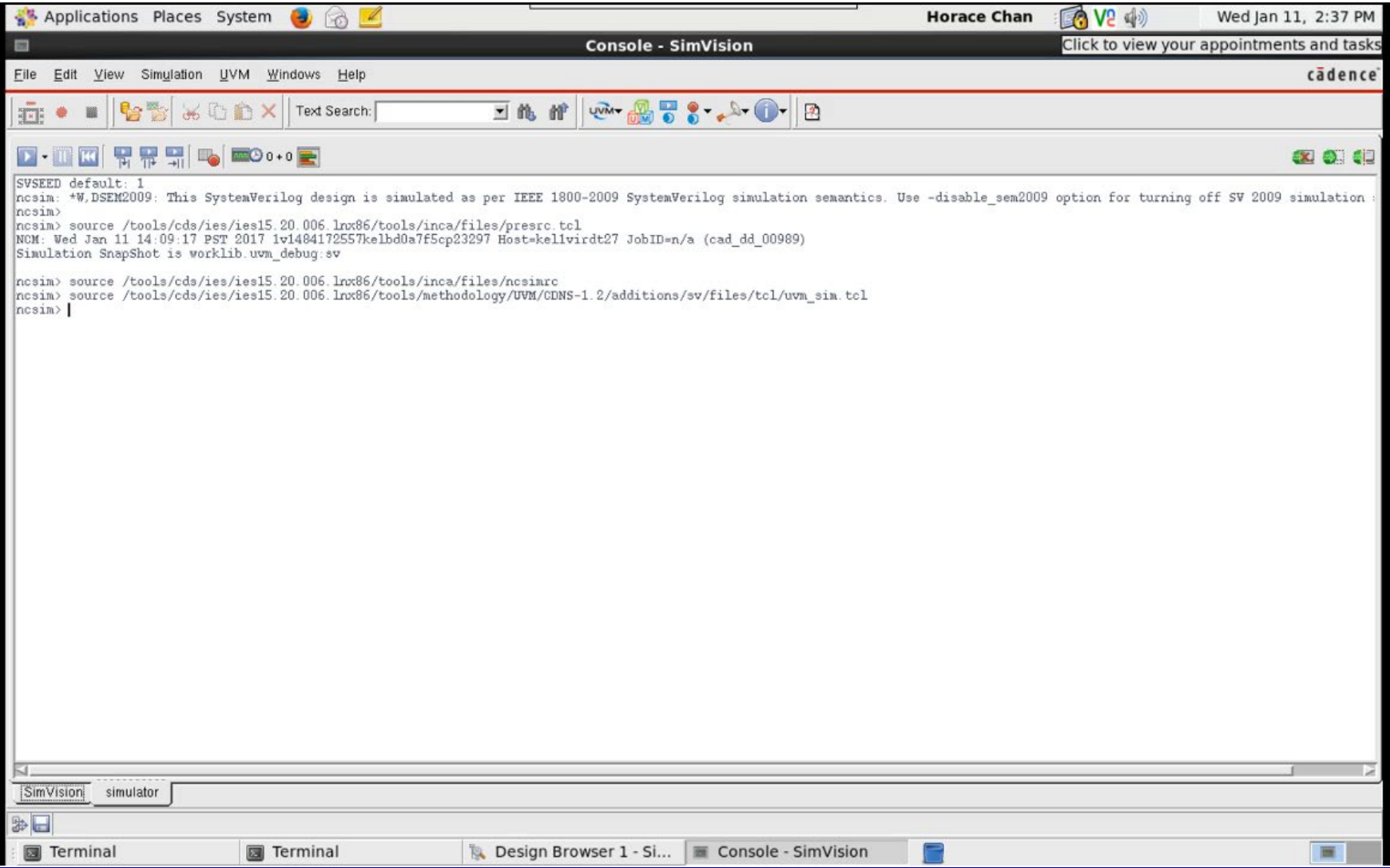
  - In the Tcl prompt, type this

```
ncsim> call debug_prompt
debug prompt (help for all commands)
1000ns: debug >
```

Demo (debug prompt and register access)



Demo (sequence control)



Built-in Debug Commands

- Housekeeping Commands:

```
help, continue, pause, run, history, repeat,
read_file, save_checkpoint
```

- Register Commands

```
wr_addr, rd_addr, wr_reg, rd_reg, wr_regfld, rd_regfld
```

- Sequence Commands

```
seq_list, seq_create, seq_rand
seq_set_fields, seq_start, seq_kill
seq_item_list, seq_item_create, seq_item_rand
seq_item_set_fields
seqr_stop_sequences, seqr_execute_item
```

User define debug command

```
class custom_debug_command extends uvm_debug_command_cb;
  function new(string name = "custom_debug_command");
    super.new(name);
    command =        "cmd_name";
    usage =          "<arg1> <arg2>";
    description =    "description of the command showed in help”;
  endfunction
  task parse_args(string args[$]);
    // parse the arguments and call the testbench function/task
        ...
    // set the return value (string)
    uvm_debug_util.rv = ...
    endtask
endclass: custom_debug_command
```

Parser Helper Functions

- **`extract_options`** support two option formats:

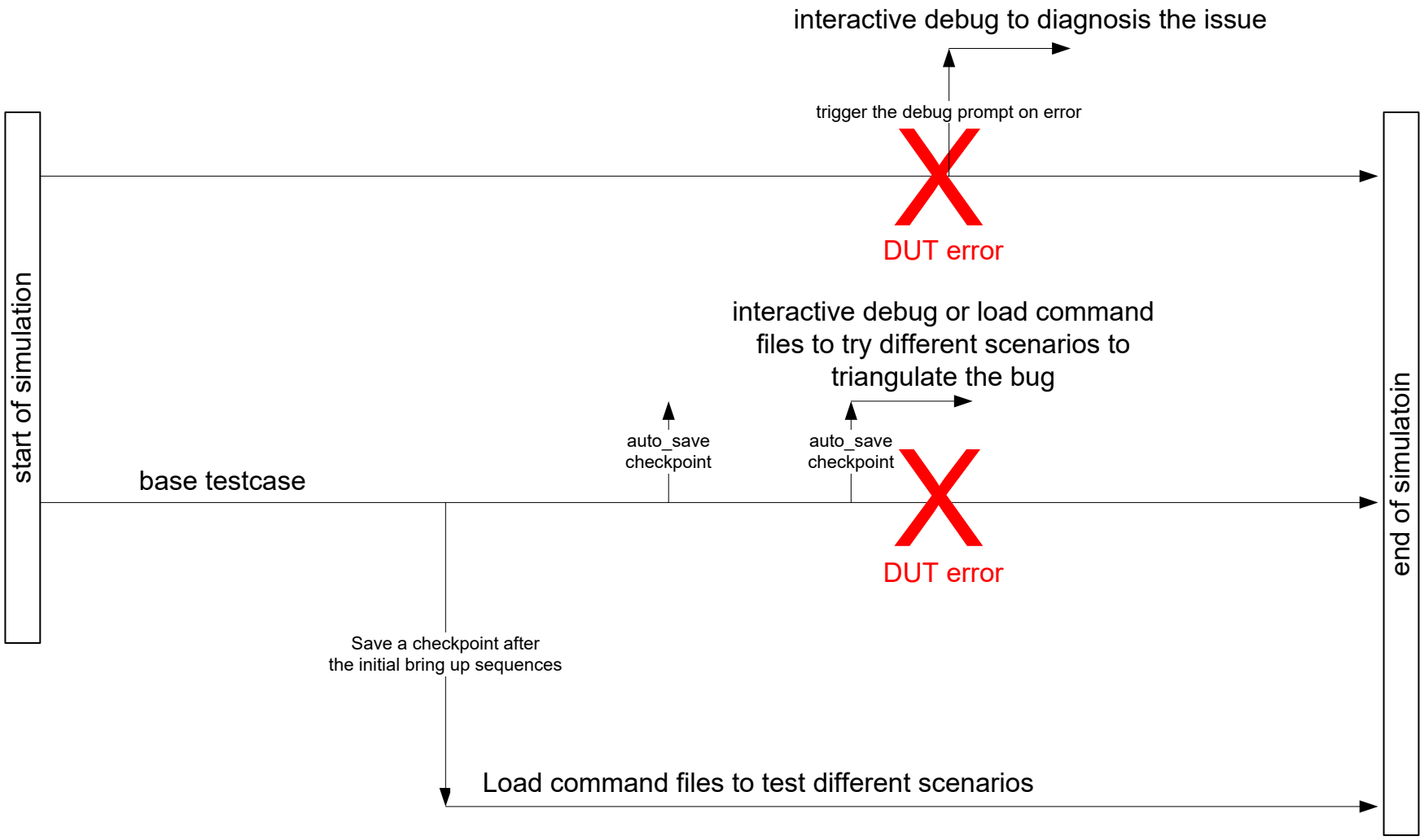
```
Tcl: -option value
SV argument: +option=value
```

- **`has_option_flag`** check does option flag exist
- String conversion functions:

```
str_to_int – convert bin/oct/dec/hex string to integer
str_to_qint – convert string to list of integer
              supports integer range <min>..<max>

get_option_string – support default value
get_option_int
get_option_int_list
```

Use Cases


interactive debug to diagnosis the issue
trigger the debug prompt on error
DUT error
interactive debug or load command files to try different scenarios to triangulate the bug
auto_save checkpoint
auto_save checkpoint
DUT error
start of simulation
base testcase
end of simulatoin
Save a checkpoint after the initial bring up sequences
Load command files to test different scenarios

2017 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Simulators Support

Features	Cadence	Mentor	Syonpsys	other
Basic features (pure SV and SV-DPI C code)	YES	YES	YES	YES
Tcl (simulator GUI) integration	YES	YES	NOT YET	NO
Simulator Tcl commands integration	YES	NOT YET	NO	NO
Demo run script	YES	NO	NO	NO

2017 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Conclusion

- Debug turnaround time reduced by **90%**
- Regression time reduced by **25%**

- Less simulator licenses
- Shorten the project schedule

- Open Source Library

download @ https://github.com/uvmdebug

Q & A