2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION CHINA

accellera SYSTEMS INITIATIVE

# DESIGN AND VERIFICATION™ CONFERENCE AND EXHIBITION

## Shanghai | September 20, 2023

KUNLUNXIN

# Use Save and Restore methodology to speed-up regression

Wenping Guo
KUNLUNXIN

# Agenda

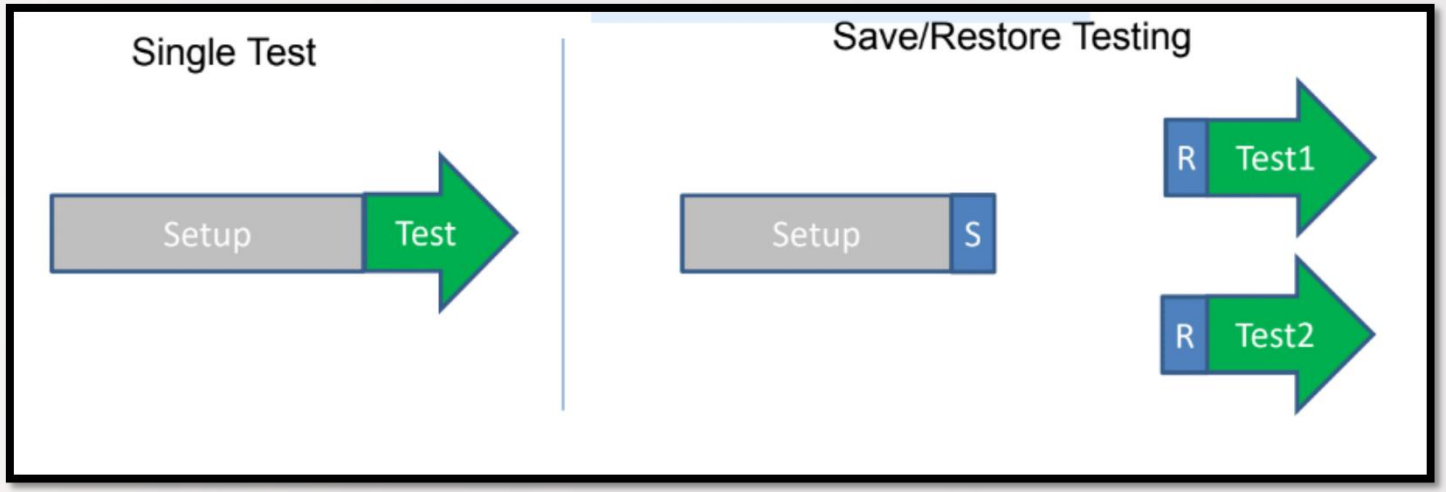
- **SnR Methodology**
  - Technology history
  - Vendor support
- **Preliminary Results**
  - Simulation result w/ SnR
  - Time consume, disk usage, restore speed-up effect summary
- **Regression&Debug speed-up**
  - Automatic save and restore regression
  - Guideline for testbench adaption
  - SnR w/ TB change
  - Project application guidance

# Agenda

- **SnR Methodology**
  - Technology history
  - Vendor support
- **Preliminary Results**
  - Simulation result w/ SnR
  - Time consume, disk usage, restore speed-up effect summary
- **Regression&Debug speed-up**
  - Automatic save and restore regression
  - Guideline for testbench adaption
  - SnR w/ TB change
  - Project application guidance

# What is SnR

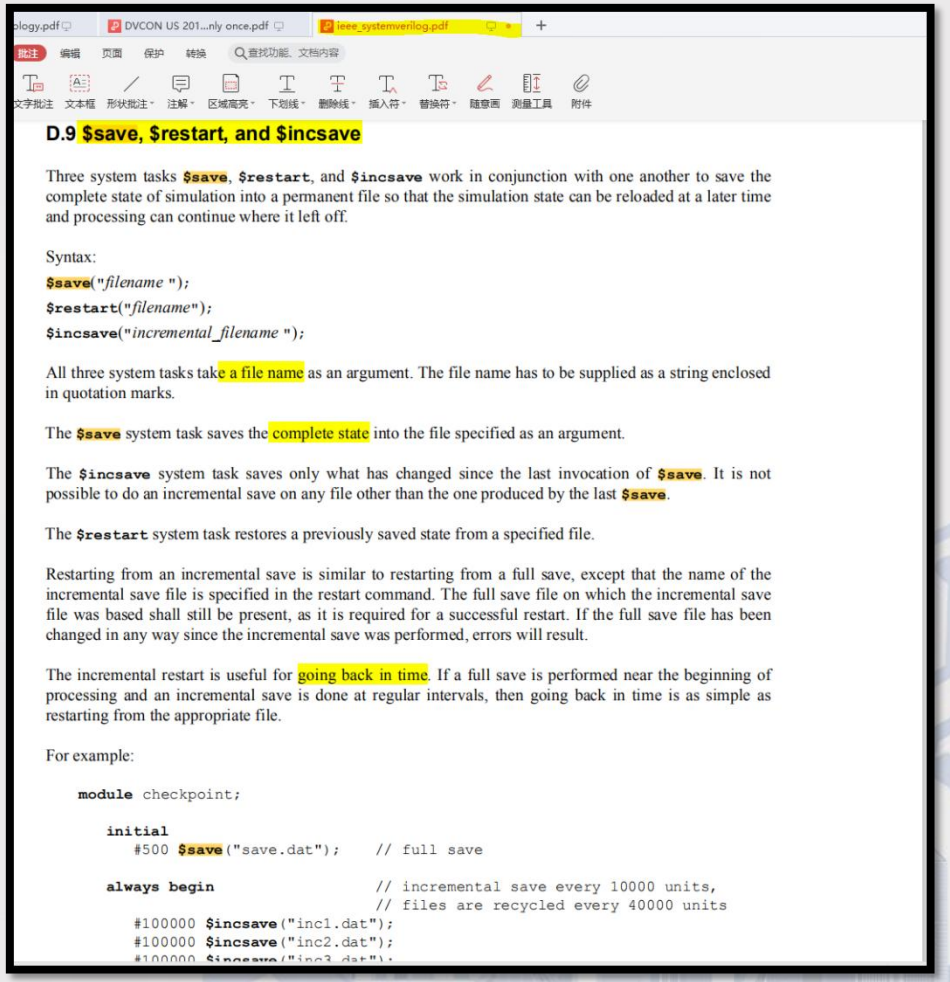
- SnR: Save and Restore




- Save the state of a simulation, then restore that state and continue
  - Save/Restore
  - Save/Restore, re-seed, and run test
  - Save/Restore, change test code, and run test

# What is SnR

- Technology History
  - IEEE 1364-1995 Supported system tasks
  - $save(“*file_name*”)
    - Save the complete state of simulation into a permanent file
  - $restart(“*file_name*”)
    - Restores a previously saved state from a specified file
    - All simulation state in IEEE std.
  - $incsave(“file_name”)
    - saves only what has changed since the last invocation of $save



D.9 $save, $restart, and $incsave

Three system tasks **$save**, **$restart**, and **$incsave** work in conjunction with one another to save the complete state of simulation into a permanent file so that the simulation state can be reloaded at a later time and processing can continue where it left off.

Syntax:

**$save**("*filename* ");
**$restart**("*filename*");
**$incsave**("*incremental_filename* ");

All three system tasks take a file name as an argument. The file name has to be supplied as a string enclosed in quotation marks.

The **$save** system task saves the complete state into the file specified as an argument.

The **$incsave** system task saves only what has changed since the last invocation of **$save**. It is not possible to do an incremental save on any file other than the one produced by the last **$save**.

The **$restart** system task restores a previously saved state from a specified file.

Restarting from an incremental save is similar to restarting from a full save, except that the name of the incremental save file is specified in the restart command. The full save file on which the incremental save file was based shall still be present, as it is required for a successful restart. If the full save file has been changed in any way since the incremental save was performed, errors will result.

The incremental restart is useful for going back in time. If a full save is performed near the beginning of processing and an incremental save is done at regular intervals, then going back in time is as simple as restarting from the appropriate file.

For example:

```
module checkpoint;

   initial
      #500 $save("save.dat");     // full save

   always begin                   // incremental save every 10000 units,
                                  // files are recycled every 40000 units
      #100000 $incsave("inc1.dat");
      #100000 $incsave("inc2.dat");
```

# Vendor support

- **Cadence**: xcelium, xrun, save and restart
  - xrun –process_save ($save)
  - xrun –r <file_name>
- **Mentor**: questa, vsim, checkpoint and restore
  - vsim –do *checkpoint.do* (checkpoint_enable = 1; ->.do)
  - vsim –restore <*file_name*>
- **Synopsys**: vcs, simv, save and restore
  - simv -ucli -ucli2Proc -do mysave.ucli
  - simv -gui , restore save_dir/save1

# Vendor support

- $save language limitation*
  - This methodology can work for designs based on various types of HDLs, i.e. Verilog, System Verilog, VHDL, or a mix of them.
  - Limited use of SystemC and C wrapper models (mostly used in an emulation environment) can be allowed with this methodology, also UVM/OVM-based designs.

- $restart status
  - All simulation status
  - Log
  - waveform

# Demo in Synopsys VCS

test.sv + (/proj/kunlun2/wenping/try/sverilog) - GVIM23

File Edit Tools Syntax Buffers Window Help

```
1  `timescale 1ps/1ps
2  module top;
3    int a;
4
5    initial begin
6       $display("a=%d", a);
7       a = 123;
8
9       #10;
10      //$system("date");
11      $display("@%0d , a=%0d", $realtime, a);
12      $save("snapshot1");
13
14      a = 456;
15      #10;
16      $display("@%0d , a=%0d", $realtime, a);
17
18      #10;
19      $value$plusargs("P=%0d", a);
20      $display("@%0d , a=%0d", $realtime, a);
21      //$system("date");
22      $finish();
23   end
24
25 endmodule
```

```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ vcs -sverilog test.sv
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ simv +P=0
Running simv on sungrid in interactive mode,please wait ...
Executing on bjkjy-ite-sillicon01.bjkjy.baidu.com
Chronologic VCS simulator copyright 1991-2018
Contains Synopsys proprietary information.
Compiler version O-2018.09-SP2-2_Full64; Runtime version O-2018.09-SP2-2_Full64;
a=          0
@10 , a=123

Warning-[STASKW_SIGTOAEF] $save will overwrite an existing file
  Warning: $save(snapshot1) is going to overwrite an existing file.


@20 , a=456
@30 , a=0
$finish called from file "test.sv", line 19.
$finish at simulation time                   30
           V C S   S i m u l a t i o n   R e p o r t
Time: 30 ps
CPU Time:      0.670 seconds;       Data structure size:   0.0Mb
Mon Feb 22 16:37:18 2021
```

KUNLUNXIN

# Demo in Synopsys VCS

test.sv + (/proj/kunlun2/wenping/try/sverilog) - GVIM23

File Edit Tools Syntax Buffers Window Help

```
1  `timescale 1ps/1ps
2  module top;
3    int a;
4
5    initial begin
6       $display("a=%d", a);
7       a = 123;
8
9       #10;
10      //$system("date");
11      $display("@%0d , a=%0d", $realtime, a);
12      $save("snapshot1");
13
14      a = 456;
15      #10;
16      $display("@%0d , a=%0d", $realtime, a);
17
18      #10;
19      $value$plusargs("P=%0d", a);
20      $display("@%0d , a=%0d", $realtime, a);
21      //$system("date");
22      $finish();
23   end
24
25 endmodule
```

```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ simv +P=0
Running simv on sungrid in interactive mode,please wait ...
Executing on bjkjy-ite-sillicon01.bjkjy.baidu.com
Chronologic VCS simulator copyright 1991-2018
Contains Synopsys proprietary information.
Compiler version O-2018.09-SP2-2_Full64; Runtime version O-2018.09-SP2-2_Full64;
a=          0
@10 , a=123

Warning-[STASKW_SIGTOAEF] $save will overwrite an existing file
  Warning: $save(snapshot1) is going to overwrite an existing file.


@20 , a=456
@30 , a=0
$finish called from file "test.sv", line 19.
$finish at simulation time                   30
          V C S   S i m u l a t i o n   R e p o r t
Time: 30 ps
CPU Time:      0.670 seconds;       Data structure size:   0.0Mb
Mon Feb 22 16:37:18 2021
```

```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ ll -sh
total 6.0M
4.0K drwxr-xr-x 3 wenping staff 4.0K Nov 30 17:55 AN.DB
8.0K drwxr-xr-x 8 wenping staff 8.0K Feb 22 17:35 csrc
788K -rwxr-xr-x 1 wenping staff 784K Feb 22 17:35 simv
4.0K drwxr-xr-x 3 wenping staff 4.0K Feb 22 17:35 simv.daidir
5.2M -rwxr-xr-x 1 wenping staff 5.2M Feb 22 17:35 snapshot1
4.0K drwxr-xr-x 2 wenping staff 4.0K Feb 22 14:40 snapshot1.FILES
4.0K drwxr-xr-x 2 wenping staff 4.0K Jan 25 17:08 stack.info.446090
4.0K -rw-r--r-- 1 wenping staff  451 Feb 22 17:34 test.sv
```

```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ ll snapshot1.FILES
total 55852
-rw-r--r-- 1 wenping staff 56962359 Feb 22 17:35 mmap.data.compressed
```

KUNLUNXIN

# Demo in Synopsys VCS

test.sv + (/proj/kunlun2/wenping/try/sverilog) - GVIM23

File Edit Tools Syntax Buffers Window Help

```
1  `timescale 1ps/1ps
2  module top;
3    int a;
4
5    initial begin
6       $display("a=%d", a);
7       a = 123;
8
9       #10;
10      //$system("date");
11      $display("@%0d , a=%0d", $realtime, a);
12      $save("snapshot1");
13
14      a = 456;
15      #10;
16      $display("@%0d , a=%0d", $realtime, a);
17
18      #10;
19      $value$plusargs("P=%0d", a);
20      $display("@%0d , a=%0d", $realtime, a);
21      //$system("date");
22      $finish();
23   end
24
25 endmodule
```

```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ simv +P=0
Running simv on sungrid in interactive mode,please wait ...
Executing on bjkjy-ite-sillicon01.bjkjy.baidu.com
Chronologic VCS simulator copyright 1991-2018
Contains Synopsys proprietary information.
Compiler version O-2018.09-SP2-2_Full64; Runtime version O-2018.09-SP2-2_Full64;
a=          0
@10 , a=123

Warning-[STASKW_SIGTOAEF] $save will overwrite an existing file
  Warning: $save(snapshot1) is going to overwrite an existing file.


@20 , a=456
@30 , a=0
$finish called from file "test.sv", line 19.
$finish at simulation time                   30
          V C S   S i m u l a t i o n   R e p o r t
Time: 30 ps
CPU Time:      0.670 seconds;       Data structure size:   0.0Mb
Mon Feb 22 16:37:18 2021
```

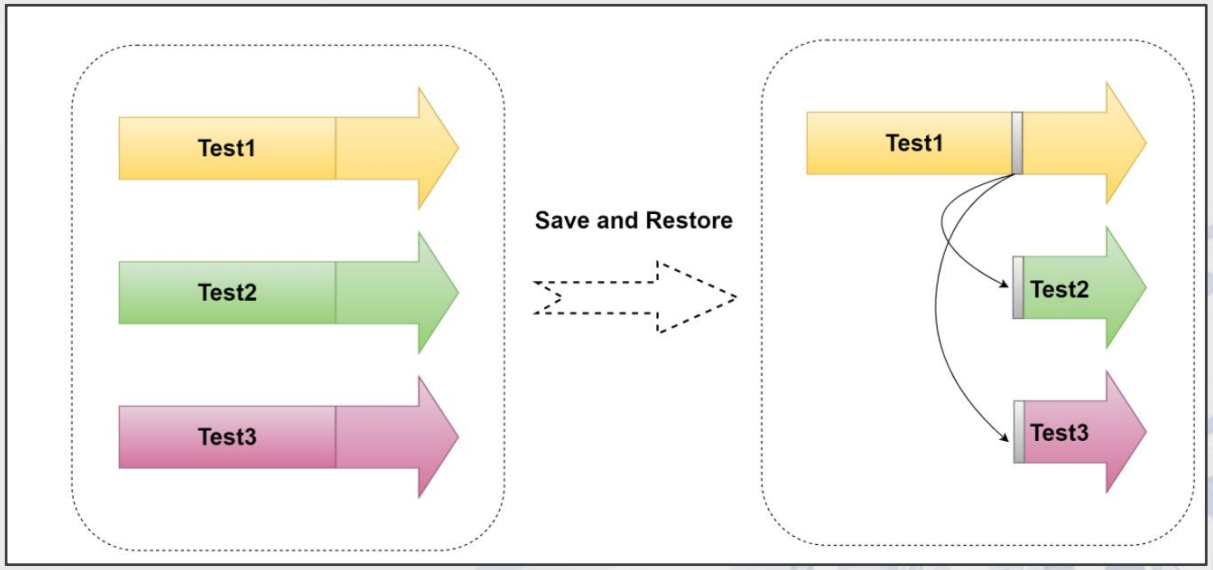
```
[wenping@ /proj/kunlun2/wenping/try/sverilog]$ simv +P=1 -r snapshot1
Running simv on sungrid in interactive mode,please wait ...
Executing on bjkjy-ite-sillicon01.bjkjy.baidu.com
@20 , a=456
@30 , a=1
$finish called from file "test.sv", line 19.
$finish at simulation time                   30
          V C S   S i m u l a t i o n   R e p o r t
Time: 30 ps
CPU Time:      0.440 seconds;       Data structure size:   0.0Mb
Mon Feb 22 16:37:28 2021
[wenping@ /proj/kunlun2/wenping/try/sverilog]$
```

# Agenda

- **SnR Methodology**
  - Technology history
  - Vendor support
- **Preliminary Results**
  - Simulation result w/ SnR
  - Time consume, disk usage, restore speed-up effect summary
- **Regression&Debug speed-up**
  - Automatic save and restore regression
  - Guideline for testbench adaption
  - SnR w/ TB change
  - Project application guidance

# Simulation result w/ SnR

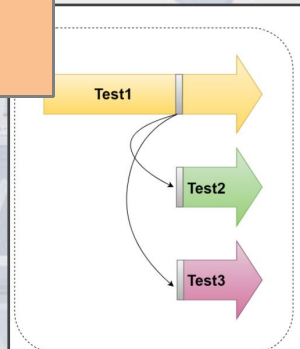
- Average speed-up effect?
- Time consume for save stage?
- Disk usage for save stage?
- Log for restore case?
- FSDB for restore case?



# Simulation result w/ SnR

| Test case (GDDR6 IP verification) | Save and Restore | CPU time(seconds) | Disk Usage for Save |
|---|---|---|---|
| the Save case:only 1 burst write and read | w/o save | 40.300 | |
| | w/ save | 40.630<br>**+8.1%** | 1.5M /simv<br>624M /compile<br>**+1.3M /snapshot: fsdb*, log, mmap.data.compressed** |
| the Restore case: multiple bursts write and read | w/o restore | 48 | |
| | w/ restore | 10.160<br>**4.72x** | |

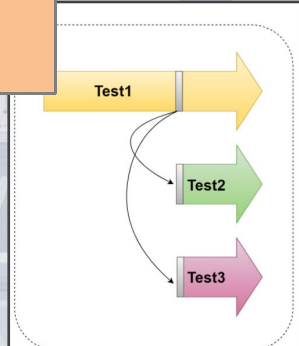
RTL Simulation result on time consume & disk usage



# Simulation result w/ SnR

| **Test case (GDDR6 netlist verification)** | **Save and Restore** | **CPU time(seconds)** | **Disk Usage for Save** |
|---|---|---|---|
| the Save case:only 1 burst write and read | w/o save | 538 | |
| | w/ save | 596<br>**+10.78%** | 3.2G /simv<br>4G /compile<br>**+900M /snapshot: fsdb*, log, mmap.data.compressed** |
| the Restore case:14bursts writeand read | w/o restore | 706 | |
| | w/ restore | 158<br>**4.46x** | |

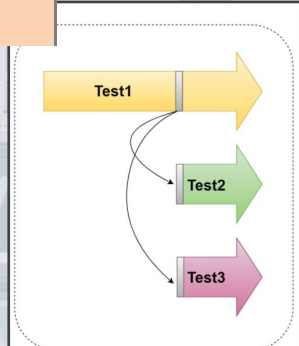
Netlist Simulation result on time consume & disk usage



# Simulation result w/ SnR

| Test case | Save and Restore | CPU time(seconds) | Speed-up |
|---|---|---|---|
| PCIe IP-level RTL simulation | w/o restore | 1142 | **29.2x** |
| | w/ restore | 39 | |
| CCIX SOC-level RTL simulation | w/o restore | 48064 | **4.36x** |
| | w/ restore | 11013 | |
| Security SOC-level RTL simulation | w/o restore | 19263 | **2.25x** |
| | w/ restore | 8556 | |

Overall speed-up result



# Simulation result w/ SnR

- Log for restore case?
  - if save case name equals restore case name, then only 1 log present, which shows save and restore message;
  - otherwise, 2 logs will present, 1 refers to the saving one, and the other refers to the restoring one, e.g.

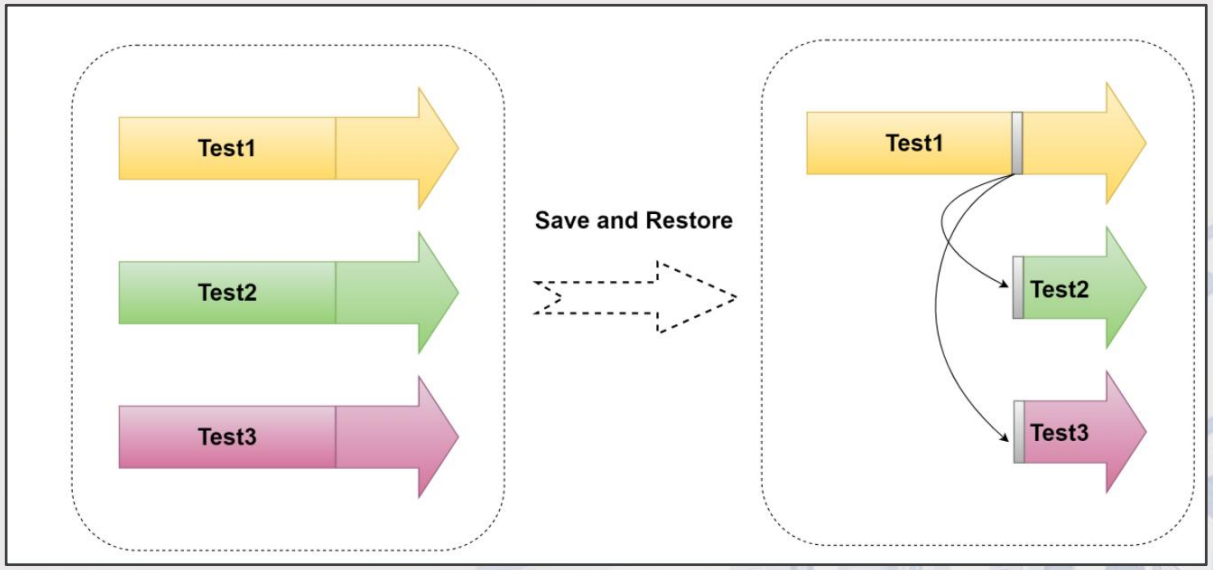
```
simv@
test_data_size_len_run.log
test_freq16g_force_run.log
```

# Simulation result w/ SnR

- FSDB for restore case?
  - if save case name equals restore case name, then the saving fsdb is a complete one;
  - otherwise, the saving fsdb contains the restoring part only

# Time consume, disk usage, restore speed-up effect summary

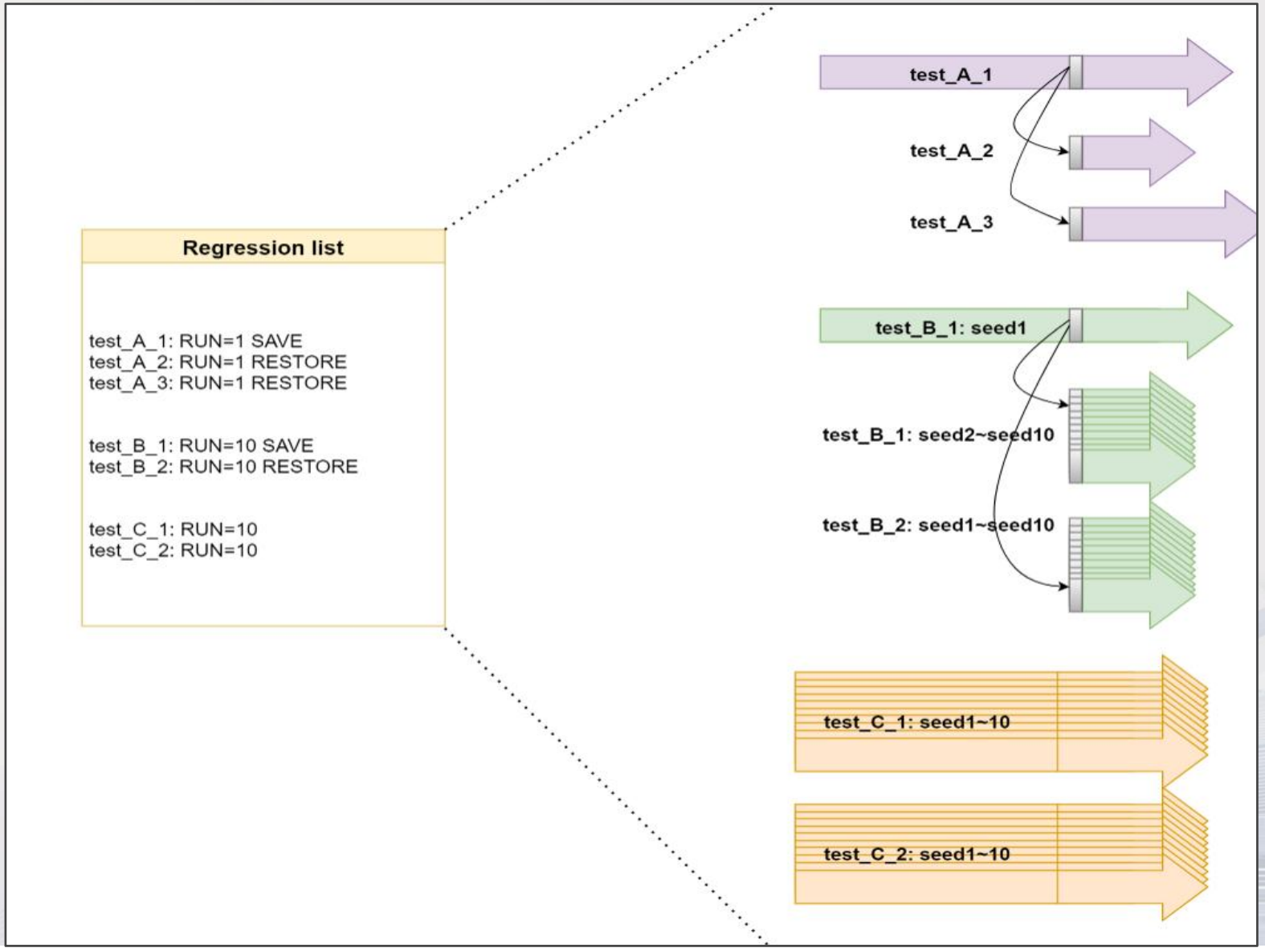
- Average speed-up effect!
- Time consume for save stage!
- Disk usage for save stage!
- Log for restore case!
- FSDB for restore case!



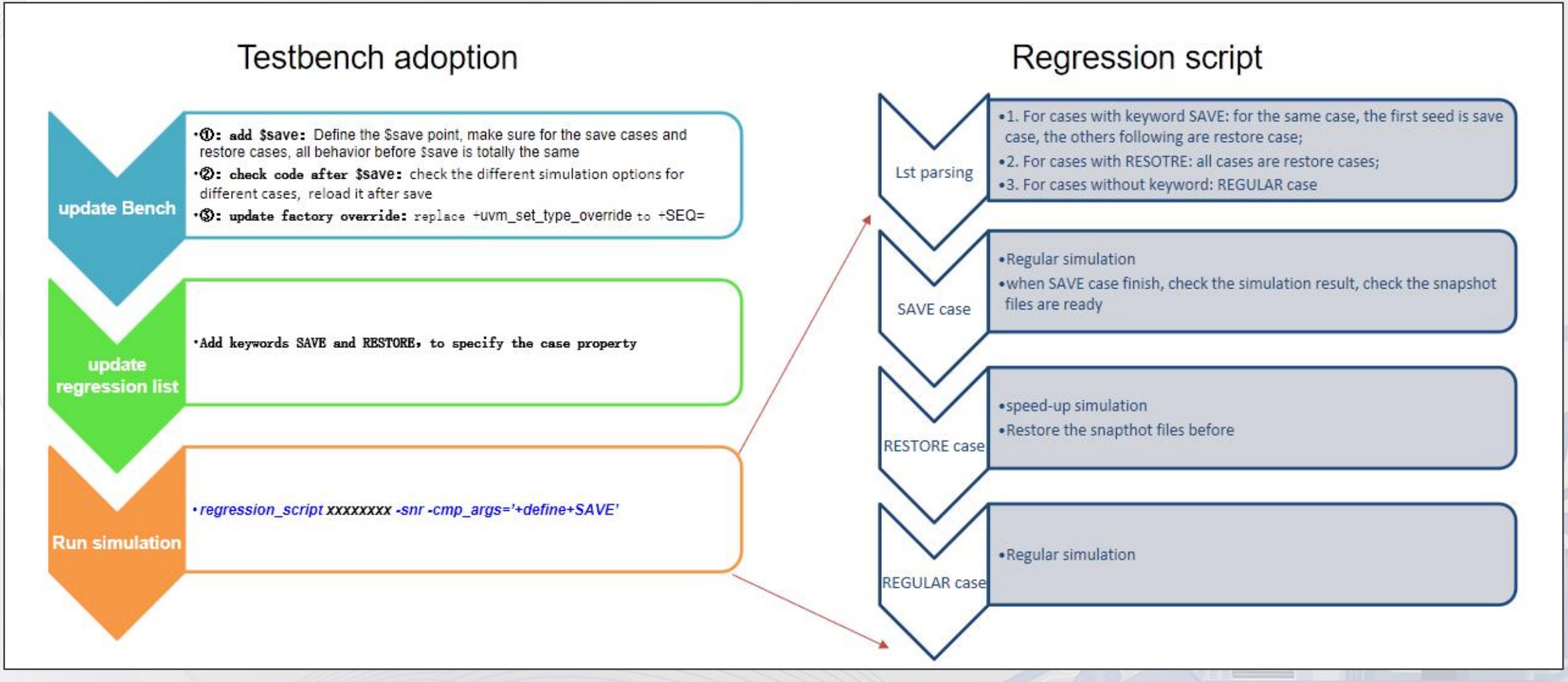
# Agenda

- **SnR Methodology**
  - Technology history
  - Vendor support
- **Preliminary Results**
  - Simulation result w/ SnR
  - Time consume, disk usage, restore speed-up effect summary
- **Regression&Debug speed-up**
  - Automatic save and restore regression
  - Guideline for testbench adaption
  - SnR w/ TB change
  - Project application guidance

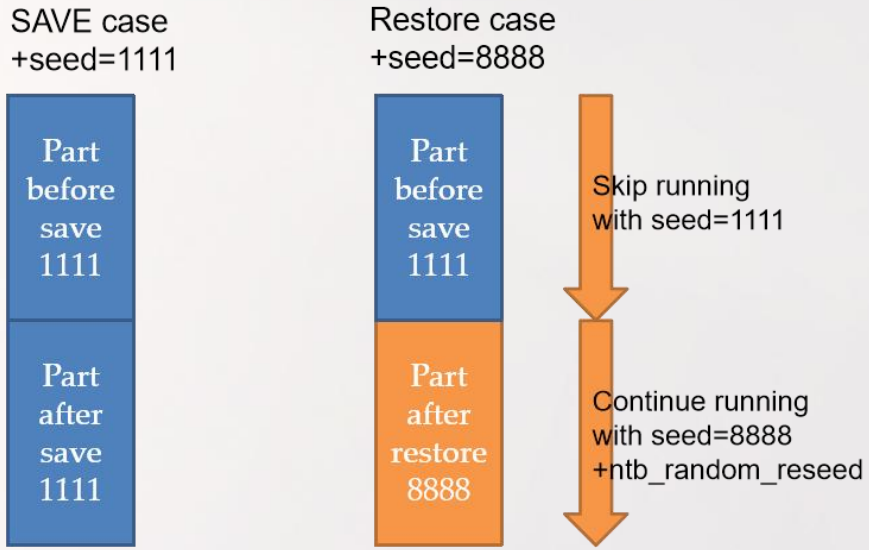
# Automatic save and restore regression



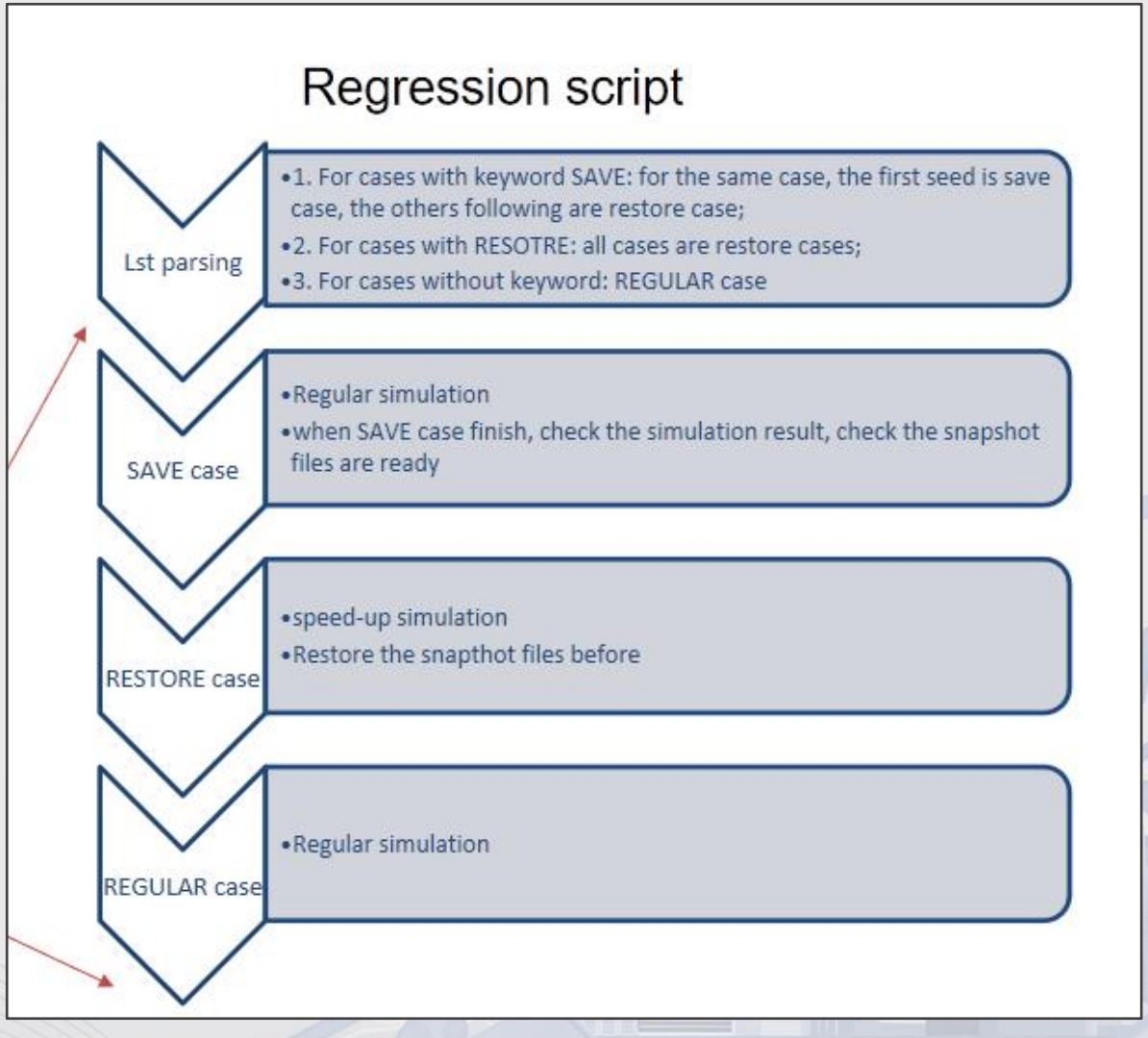
# Guideline for testbench adaption



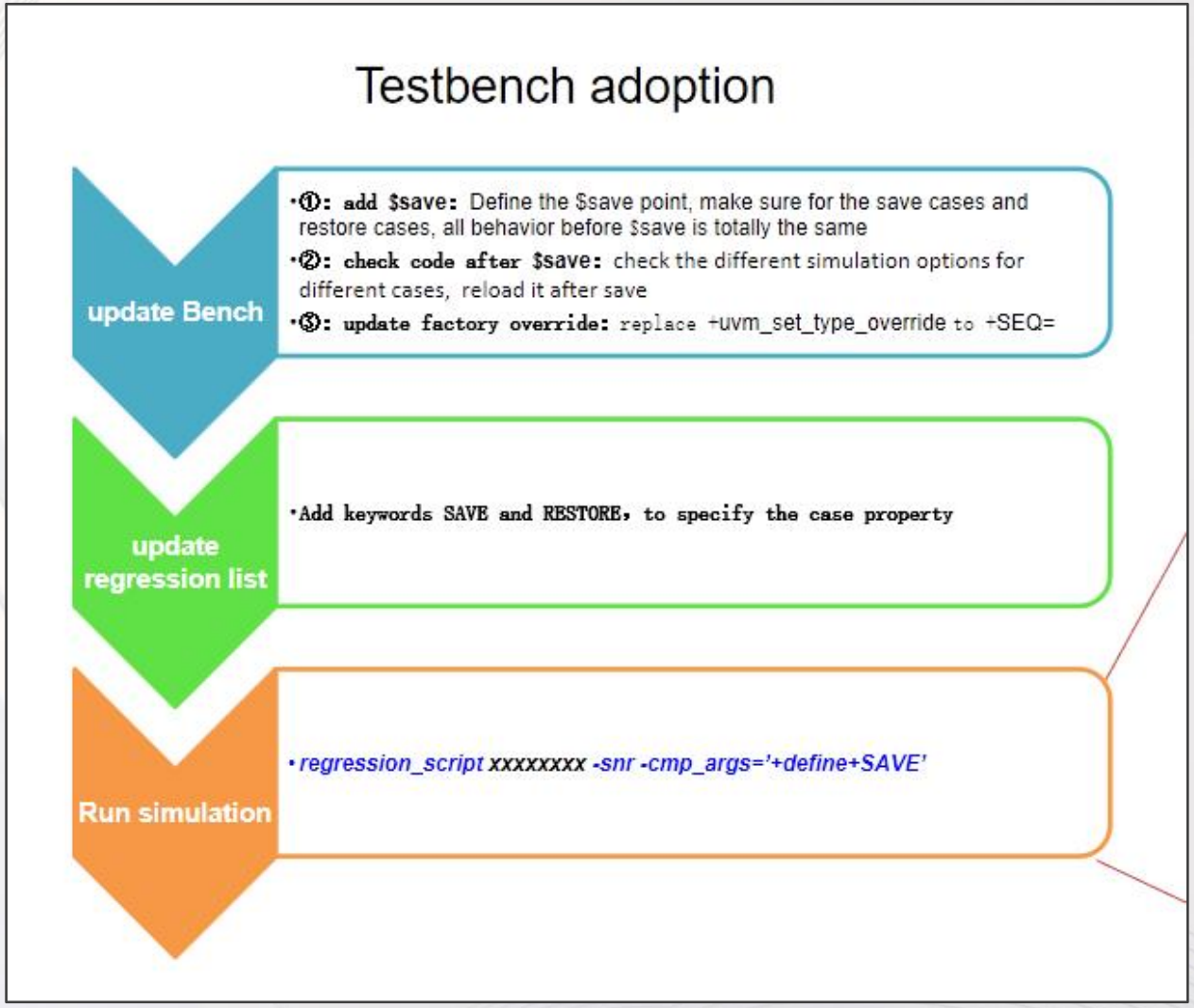
# Guideline for testbench adaption




```
Note-[RNG-RESEED] +ntb_random_reseed specified
  +ntb_random_reseed used in the command line. New seed 417405 will be used.
  New seed is based on +ntb_random_seed= or +ntb_random_seed_automatic.
```



# Guideline for testbench adaption



# Guideline for testbench adaption

verif/gddr6_wrapper/tb/seq/senario_template_vseq.sv

```
1
2  `ifndef _SENARIO_TEMPLATE_VSEQ_SV_
3  `define _SENARIO_TEMPLATE_VSEQ_SV_
4
5  class senario_template_vseq extends gddr6_wrapper
6
7    `uvm_object_utils_begin(senario_template_vseq)
8    `uvm_object_utils_end
9
10   `uvm_declare_p_sequencer(gddr6_wrapper_virtual_
11
12   function new (string name = "senario_template_v
13     super.new(name);
14   endfunction : new
15
16
17   virtual task body();
18     axi_cfg_base_vseq     seq_cfg_base;
19     axi_data_base_vseq    seq_data_base;
20
21     `uvm_info(get_type_name(), "entering body..."
22
23     `uvm_create(seq_cfg_base)
24     `uvm_create(seq_data_base)
25
26     seq_cfg_base.start(p_sequencer);
27     seq_data_base.start(p_sequencer);
28
29     `uvm_info(get_type_name(), "senario seq endin
30   endtask:body
31
32 endclass: senario_template_vseq
33
34
35 `endif //_
```

```
17   virtual task body();
18     axi_cfg_base_vseq     seq_cfg_base;
19     axi_data_base_vseq    seq_data_base;
20
21     `uvm_info(get_type_name(), "entering body...", UVM_LOW)
22
23     `uvm_create(seq_cfg_base)
24     seq_cfg_base.start(p_sequencer);
25
2      //save snapshot here
2  `ifdef SAVE
2      `uvm_info(get_type_name(), "Phy initial finished, now saving snapshot", UVM_LOW)
2      $save("snapshot1");
3      #1;
3  `endif
3
3      //go-on data write and read
3      begin
3        string selected_seq = "axi_data_base_vseq";
3        uvm_object_wrapper wrapper;
3        $value$plusargs("SEQ=%0s", selected_seq);
3        wrapper = factory.find_by_name(selected_seq);
3        `uvm_info(get_type_name(), "create seq_data_base", UVM_LOW)
4        $cast(seq_data_base, create_item(wrapper, m_sequencer, "seq_data_base"));
4        seq_data_base.start(p_sequencer);
4      end
4
44     `uvm_info(get_type_name()
45   endtask:body
46
47 endclass: senario_template_vs
```

```
25
26 test_data_size_len : gddr6_wrapper_base_test
27     +uvm_set_type_override=gddr6_wrapper_base_vseq,senario_template_vseq
28     +uvm_set_type_override=axi_cfg_base_vseq,cfg_init_vseq
29     +uvm_set_type_override=axi_data_base_vseq,data_traverse_vseq
30     +SEQ=data_traverse_vseq
31     +ddr_init_by_force
32 endargs
33
```

# Guideline for testbench adaption

```
559
560
561 // build_phase
562 // -----
563
564 function void uvm_root::build_phase(uvm_phase phase);
565
566   super.build_phase(phase);
567
568   m_set_cl_msg_args();
569
570   m_do_verbosity_settings();
571
572   m_do_factory_settings();
573
574   m_do_max_quit_settings();
575   m_do_dump_args();
576
577 endfunction
578
579
```

在uvm_root的build_phase被调用

所以如果我们使用restore的方式跳过了bench的build_phase，这段代码就无法被执行，会漏掉所有新增的、新改变的 override

```
35
36  virtual task body();
37    bit status;
38    // get iconfig
39    status = uvm_config_db#(iconfig)::get(null, get_full_name
40    if(! status) `uvm_error(get_type_name(),"get iconfig faile
41    `uvm_info(get_type_name(),"iconfig.get_args...",UVM_MEDIUM
42    icfg.get_args();
43    icfg.print();
44
45    // timeout count
46    fork
47        forever begin
48            #1us;
49            icfg.pkt_timeout++;
```

```
646
647 // m_do_factory_settings
648 // ---------------------
649
650 function void uvm_root::m_do_factory_settings();
651   string args[$];
652
653   void'(clp.get_arg_matches("/^\\+(UVM_SET_INST_OVERRIDE|uvm_set_inst_override)=/",args));
654   foreach(args[i]) begin
655     m_process_inst_override(args[i].substr(23, args[i].len()-1));
656   end
657   void'(clp.get_arg_matches("/^\\+(UVM_SET_TYPE_OVERRIDE|uvm_set_type_override)=/",args));
658   foreach(args[i]) begin
659     m_process_type_override(args[i].substr(23, args[i].len()-1));
660   end
661 endfunction
662
663
```

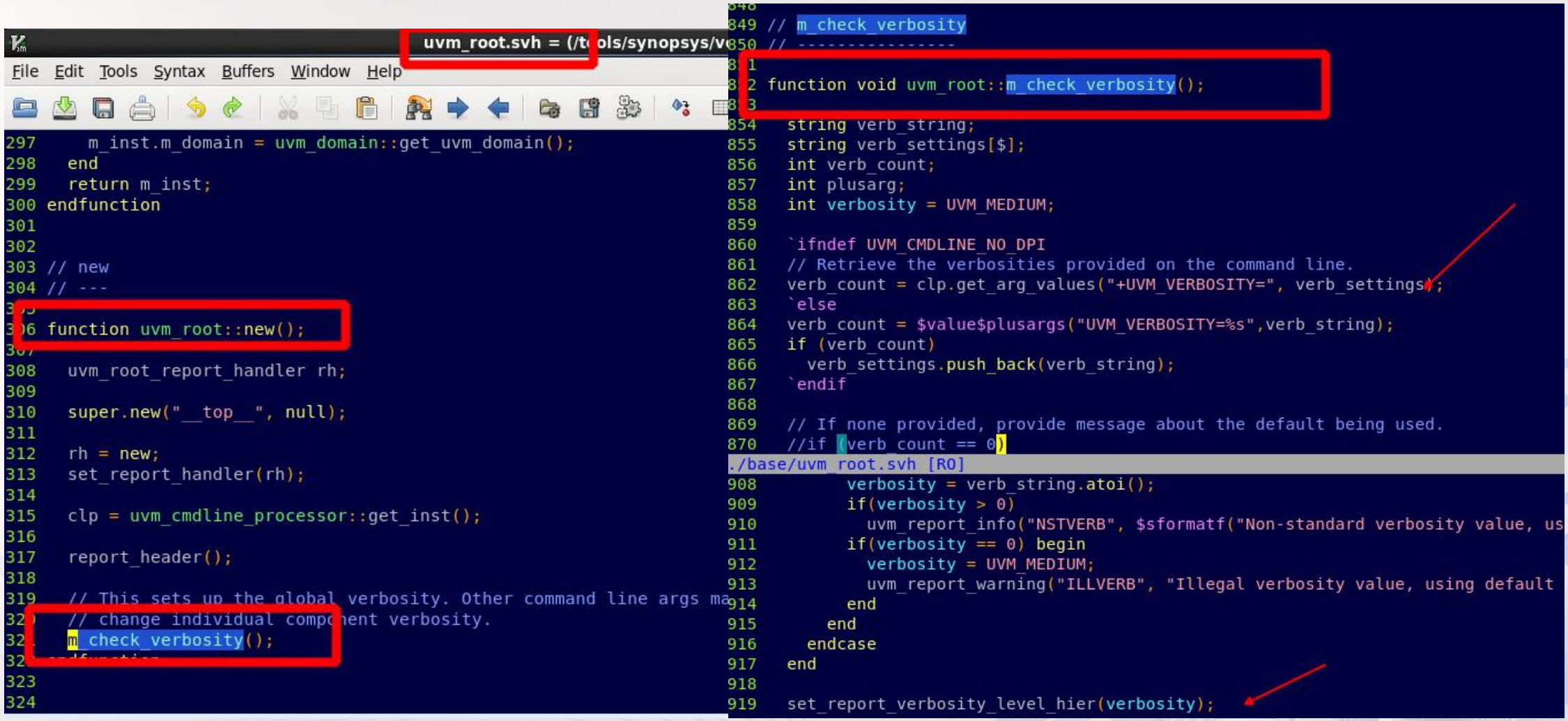
解析plusargs

灌入factory

# Guideline for testbench adaption

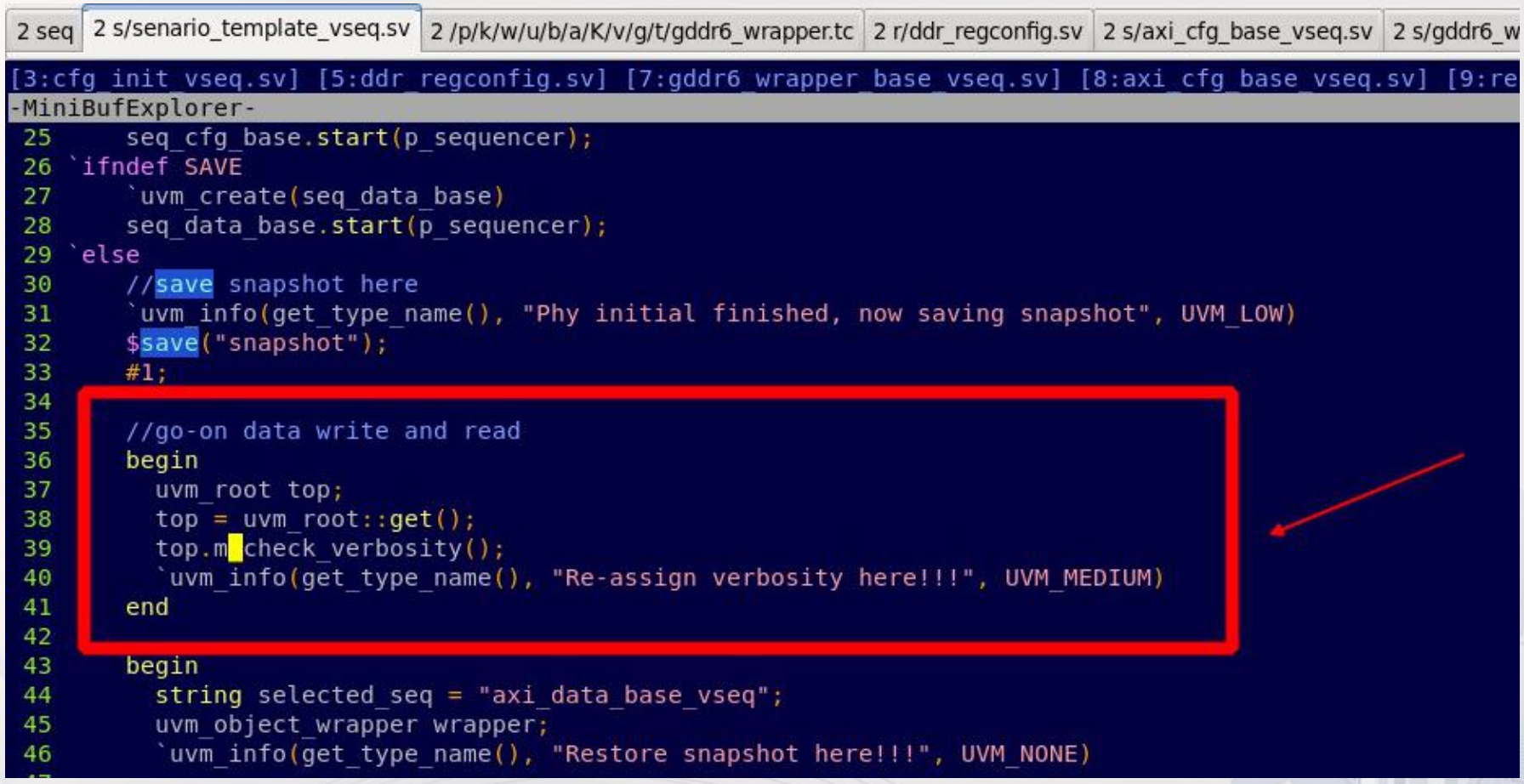
- Verbosity Override requirement
  - Since UVM_VERBOSITY plusargs is handled in new function of uvm_root, if restore jumps new function, then this option will not work any more in restore cases.
  - In this case, user need to apply the verbosity once again after save point, and apply method set_report_verbosity_level_hier to top.

# Guideline for testbench adaption



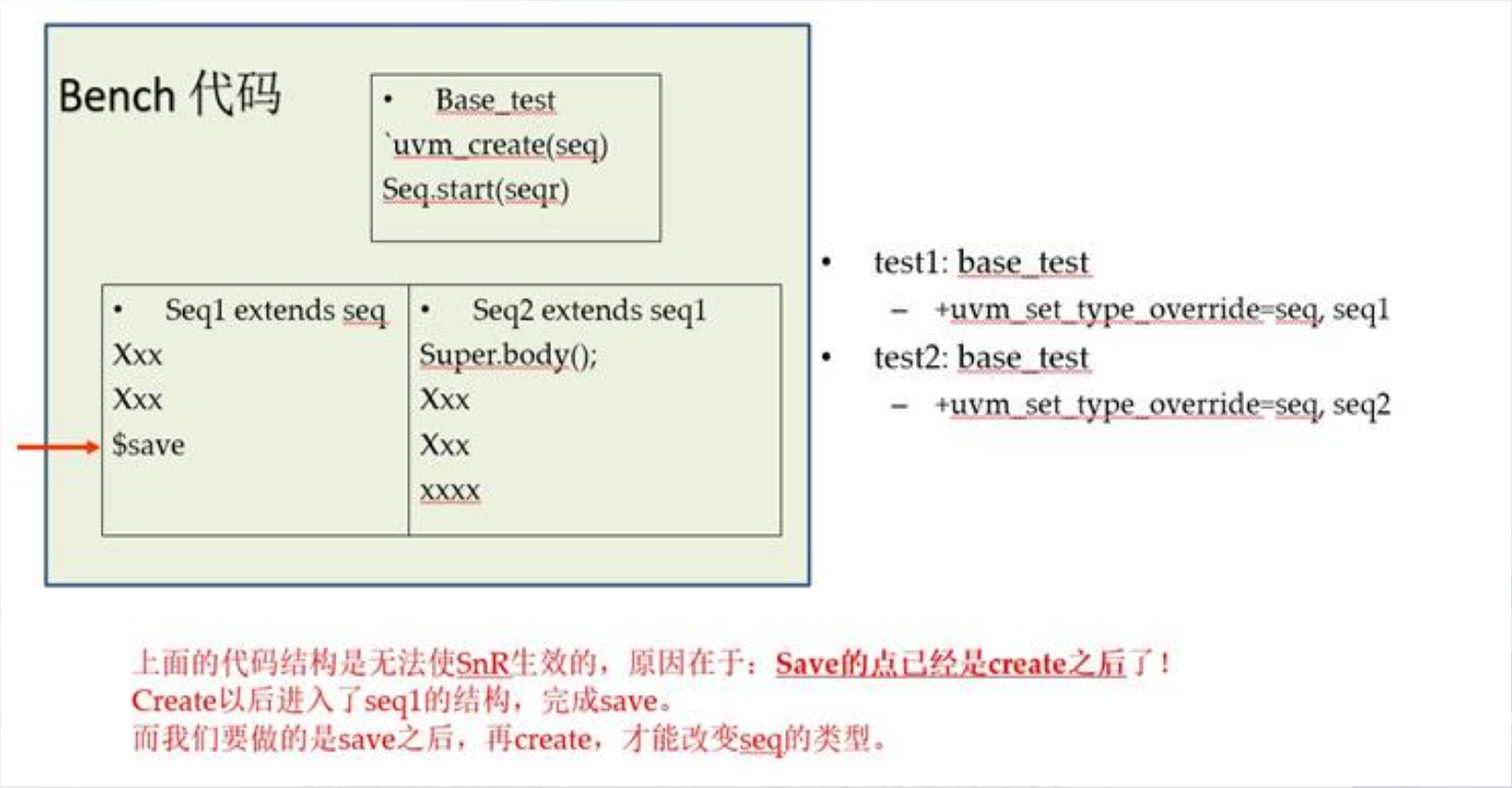
KUNLUNXIN

# Guideline for testbench adaption



# Guideline for testbench adaption

- Bench reorganization



# SnR w/ TB change

- SnR + DTL
  - Use SnR together with DTL, so that when TB changes, we can still use restore.
    - For Debug:
      - Address the amount of time to reproduce failures, save image of simulation around the failure point, and debug from that point.
      - Phased saving, address the latest snapshop before failure.
    - For Test develpment:
      - Avoid running setup and initialization when developing test sequences changes.

# Project application guidance

- When
  - Can use only if a common initial setup phase for all tests
    1. UVM build phase, configuration phase
    2. Reset state: CMU, PMU
    3. Design initialization: DDR PHY initialization, training, calibration; SDRAM loading initial memory
    4. Sequence initialization: SW driver, video/graphics controller setup after reset
    5. ……
  - In addition, this setup phase itself takes a substantially large amount of cycles with respect to overall length of the simulation for a test, and these cycles must be repeated over and over again for all subsequent tests.

# Project application guidance

- Gain Throughput on Simulation
  - Regression/Debug/Development
    - Reduce time spent simulating training links and initializing design
    - Reduce rerunning of same initialization sequence
    - Focus on testing on “interesting” part of the simulation

# Agenda

- **SnR Methodology**
  - Technology history
  - Vendor support
- **Preliminary Results**
  - Simulation result w/ SnR
  - Time consume, disk usage, restore speed-up effect summary
- **Regression&Debug speed-up**
  - Automatic save and restore regression
  - Guideline for testbench adaption
  - SnR w/ TB change
  - Project application guidance

KUNLUNXIN

# Thanks