2017 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Debug APIs
- next wave of innovation in DV space

Srinivasan Venkataramanan (srini@cvcblr.com)

Ajeetha Kumari (akumari@cvcblr.com)

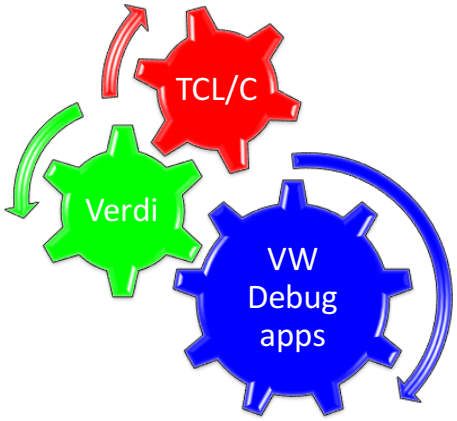
VerifWorks, http://www.verifworks.com

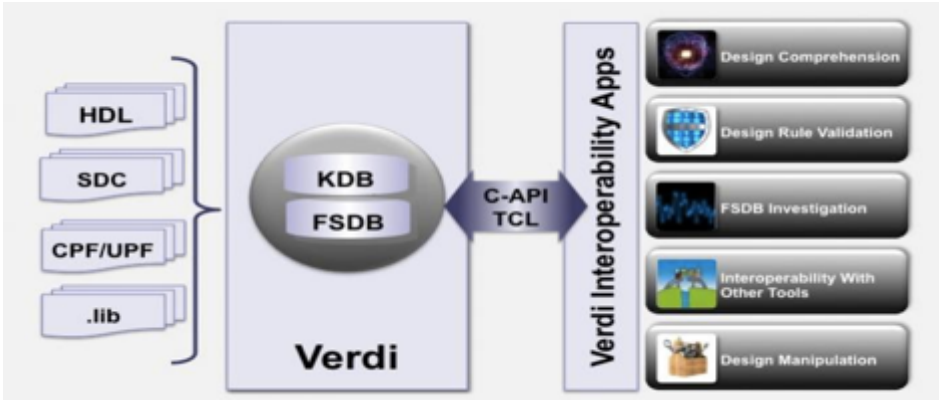
What's Debug in verification?

- Test failures:
  - Simulation
  - Formal Verification
- Debug process:
  - Traditionally more manual
  - Error prone
  - Consumes time
- Iterative
  - Run "tests" several times to corner-down the "bug"

A debug "app"

- Debug "app" – an application that aids in debug
- Apps can be of varied size, features etc.
- Cross platform "portable" apps are ideal goal





First gen debug apps

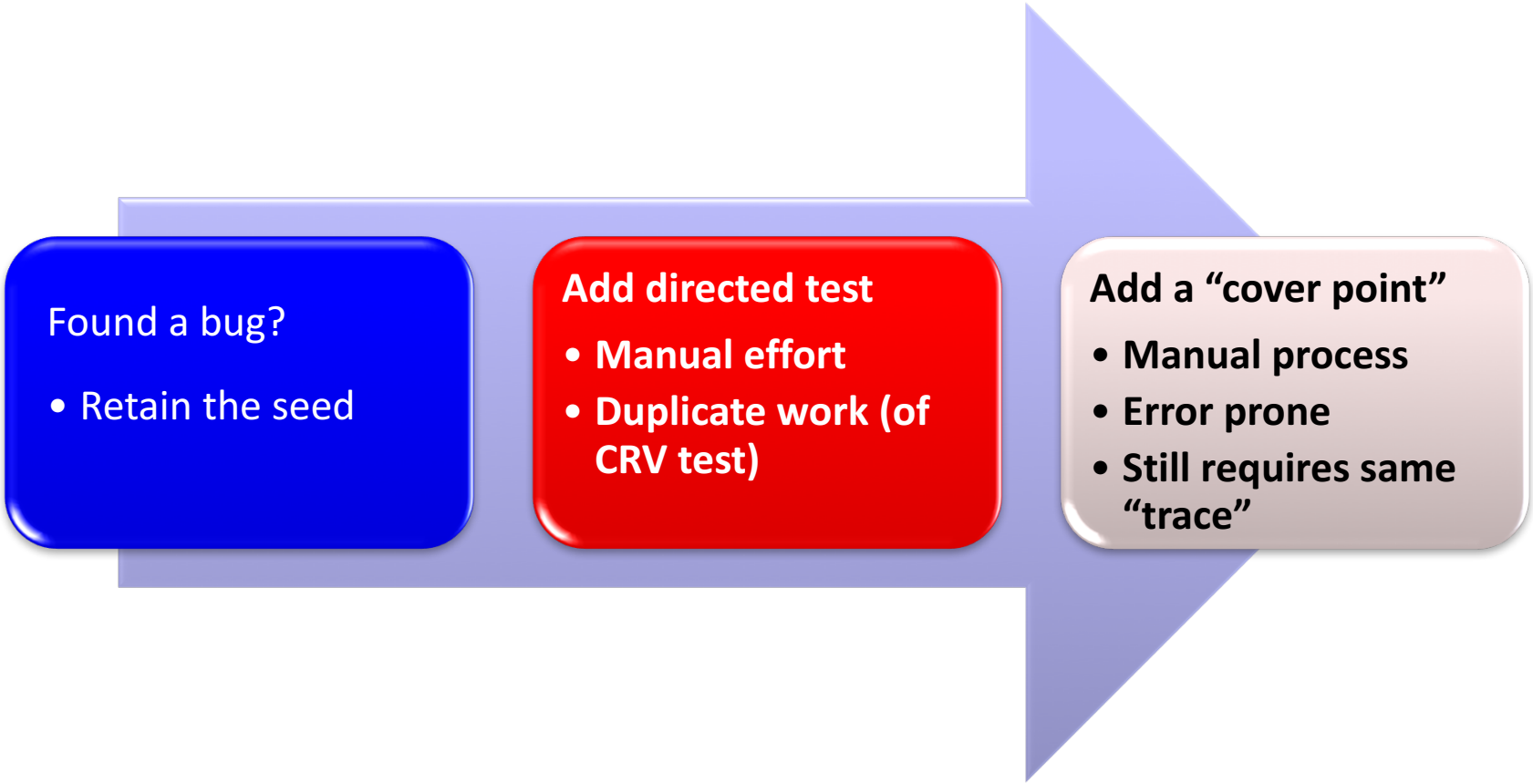
- Huge dump file size?
  - Split into multiple physical files
  - Circumvent 32-bit OS limitation
  - Smaller files are faster to analyze
- Selected dumping
  - Saves memory
  - Faster simulation
  - Selected parts of design
  - Selected time period(s)
- RTL-2-GL mapping apps
- X-tracing apps

```
assertion action -cond pass
  top.dump_start_seq  -exec {
  dump -fsdb smart_dump.fsdb}
```

RTGen – a novel debug app

- **R**egression **T**est **Gen**erator
- Extracts a testcase from waveform
- Helps in capturing quality traces
  - Typical error scenarios
  - Past bugs
  - Specification driven
- Formal Verification (FV) to simulation
  - Extract a test from a FV failure trace
- Executable specification
  - Timing Diagrams to testcase

Life without RTGen in UVM flow



RTGen + Verdi flow

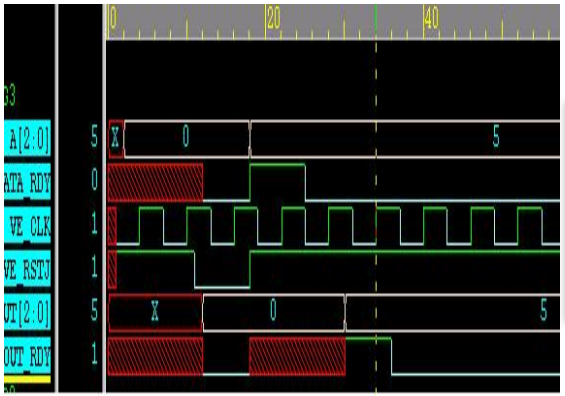
RTGen

FSDB

VCD

VPD

EVCD



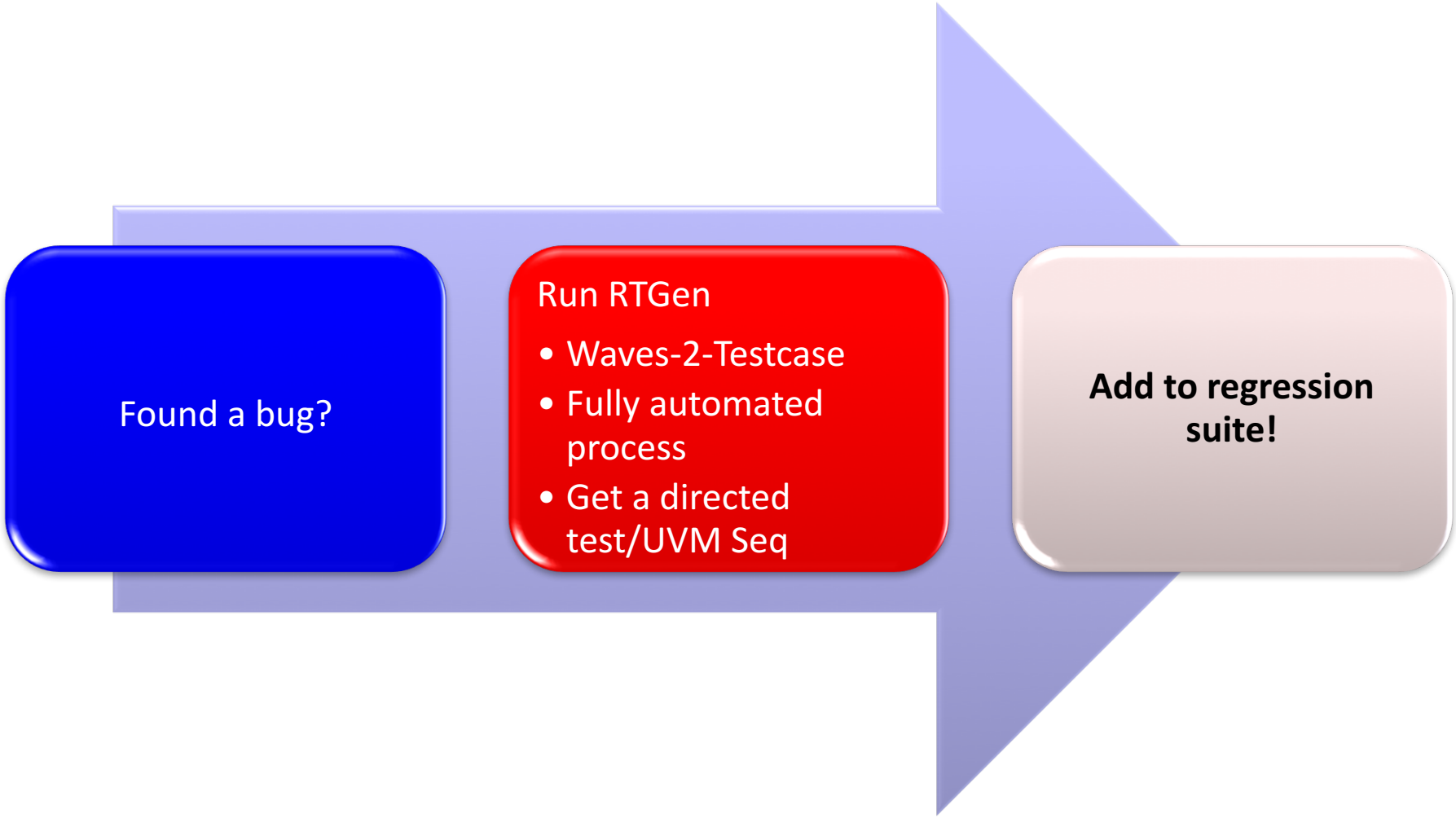
Verilog

UVM

SV

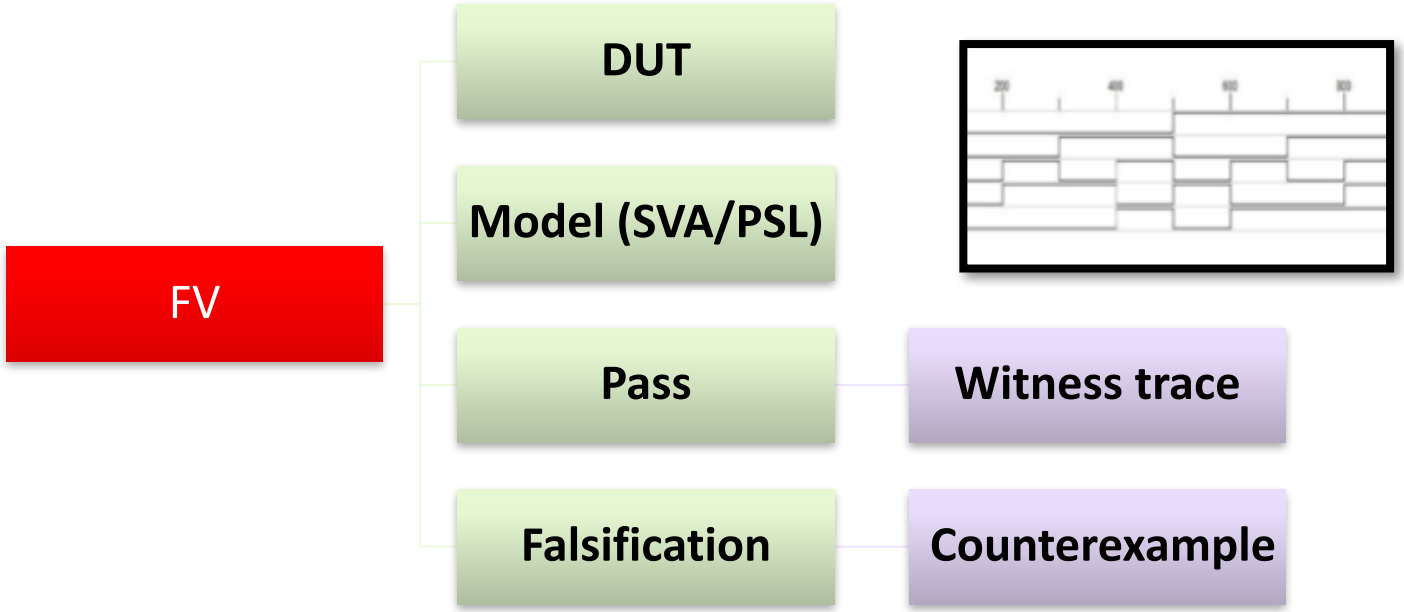
VHDL

RTGen in UVM flow

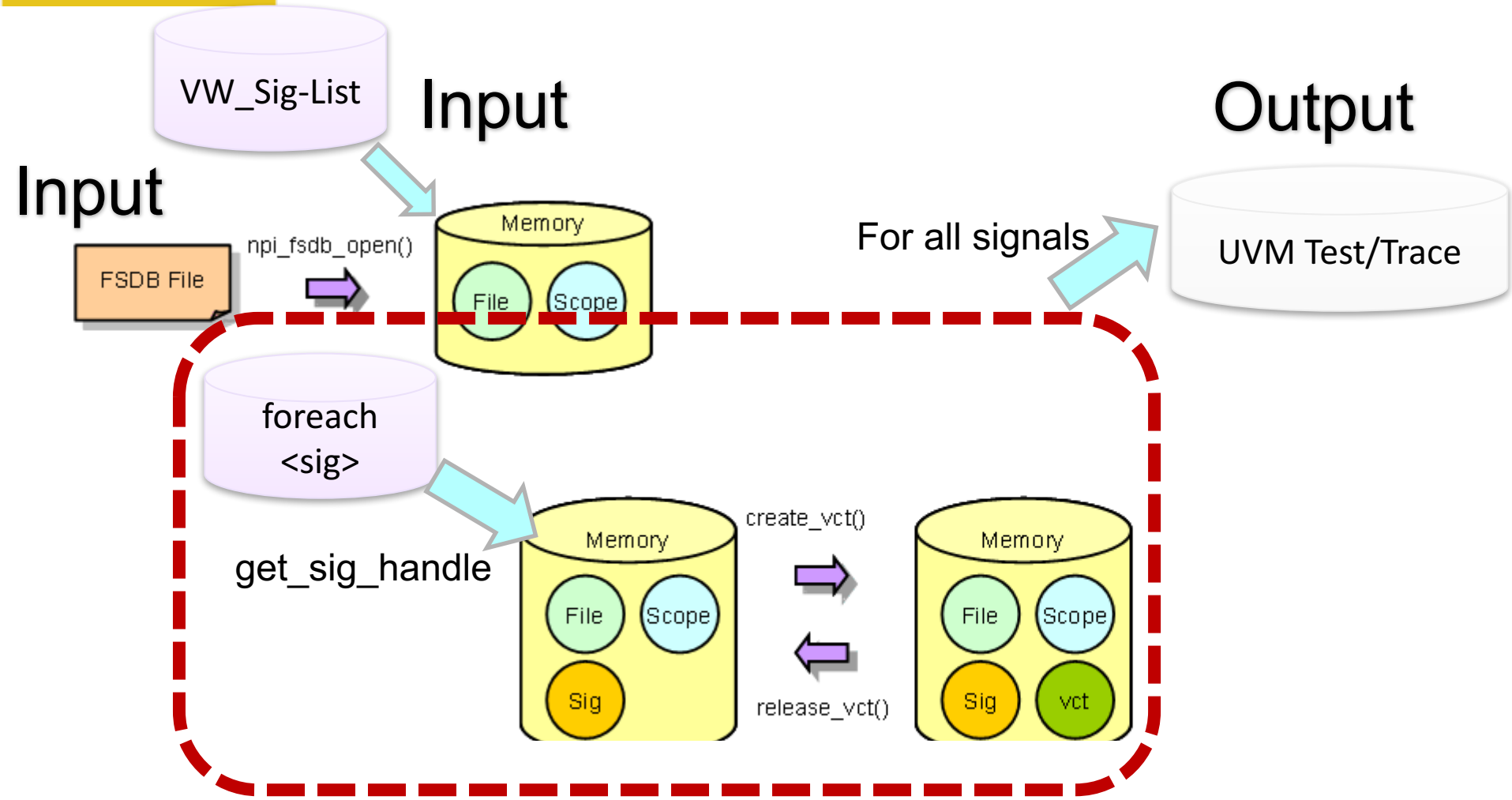


RTGen in Formal flow

- RTGen is a proven technology in FV centric flow



RTGen Architecture

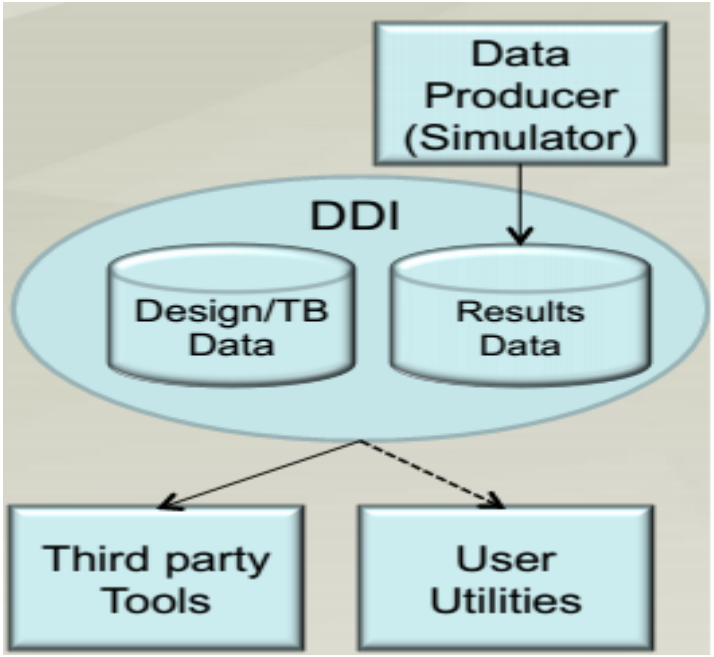


RTGen extracted test


```
task main(); begin
  fork
    drive_memread();
    drive_reset();
    drive_adr();
    drive_memwrite();
    drive_memdata();
    drive_writedata();
    drive_clk();
  join
end
```

Debug Data API (DD-API)

DDI flow



Sample DDI code

```
p_vpi_time  T1 ;
p_vpi_time  T2 ;
char*  prevValStr ; // e.g *,'b0, 'h*ffe*, 'h000
char*  currValStr ; // e.g *,'b1, 'h*ffe*, 'h001
T1->type = vpiMinTime ;
vpi_get_time (waveFileHandle,T1);
T2->type = vpiMaxTime ;
vpi_get_time (waveFileHandle,T2);

vpiHandle vctHandle = vpi_handle_vct  (var1Handle,waveFileHandle);
vpiHandle vctIter =
      vpi_iterate_vc (vctHandle,T1,T2,prevValStr,currValStr);
vpiHandle vcHandle ;
while (vcHandle = vpi_scan(vctIter))
{
   time_p->type = vpiSimTime;
   vpi_get_time  (vcHandle,time_p);
   vpi_get_value (vcHandle,value_p);
   edgeType = vpi_get (vpiEdge, vcMemberHandle);
}
```