DFT Concept & TetraMAX

Advisor: Prof. YU-CHENG FAN
Presenter: JIAN-PING, WANG

2023.03.20

Outline

- DFT Concept
- Tetramax
- Practice

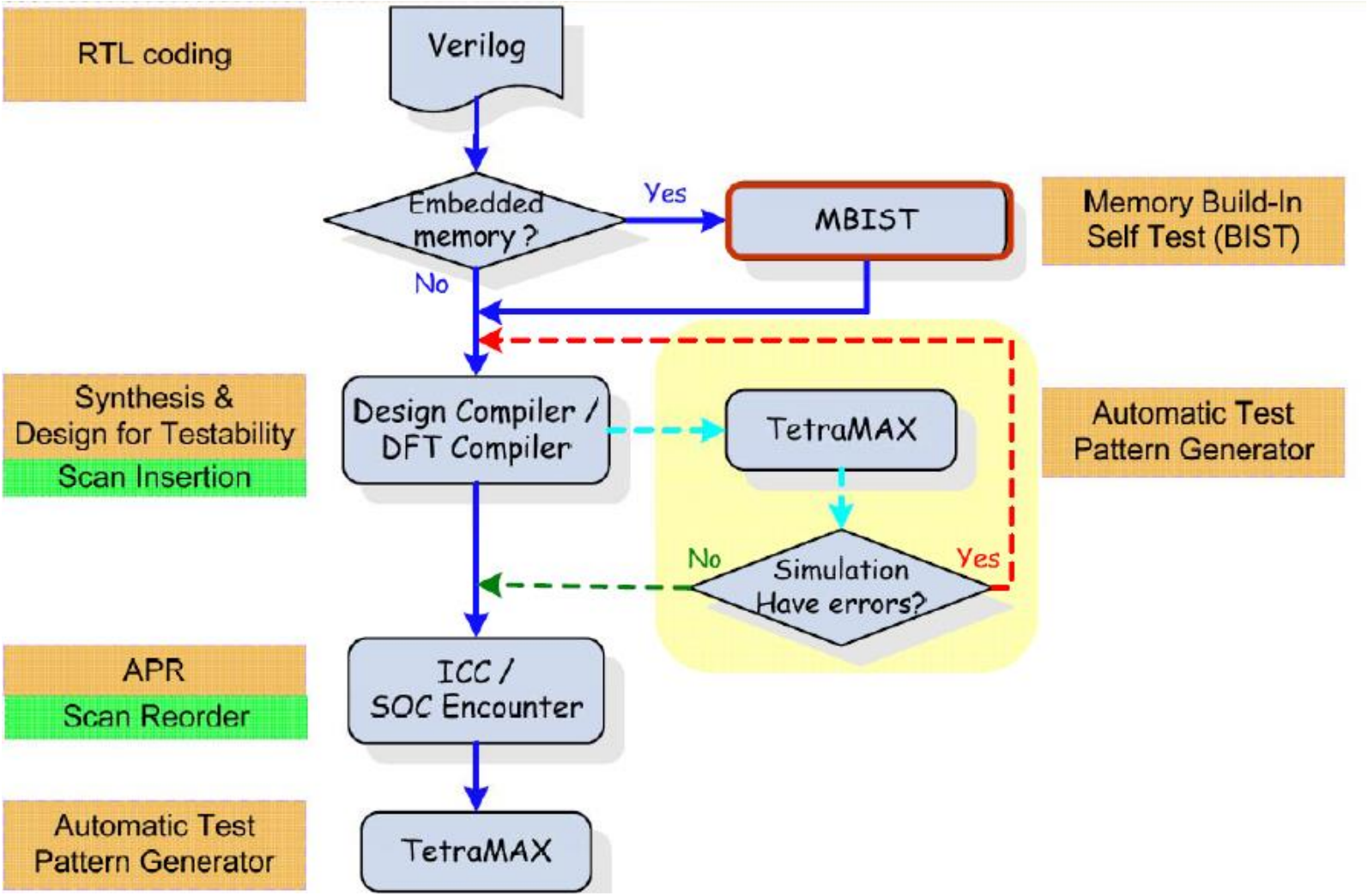
DFT Concept

- Design For Testability

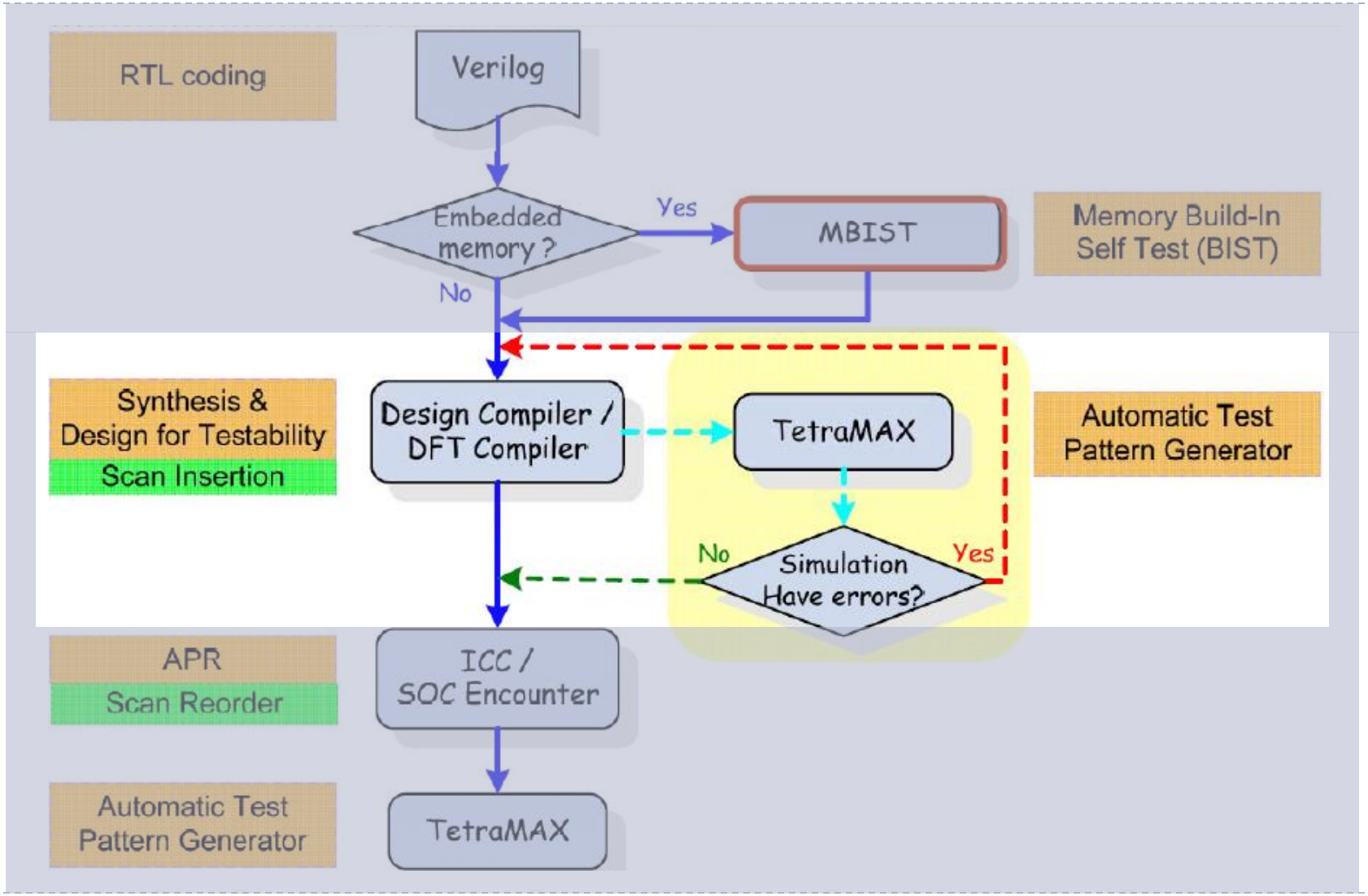
- What is testing?
- Testing is a process of determining whether a device is good or not.
- Testing includes test pattern generation, application and output evaluation.

- Why testing?
- In order to guarantee the product quality, reliability, performances, etc.
- Cost is the most important.

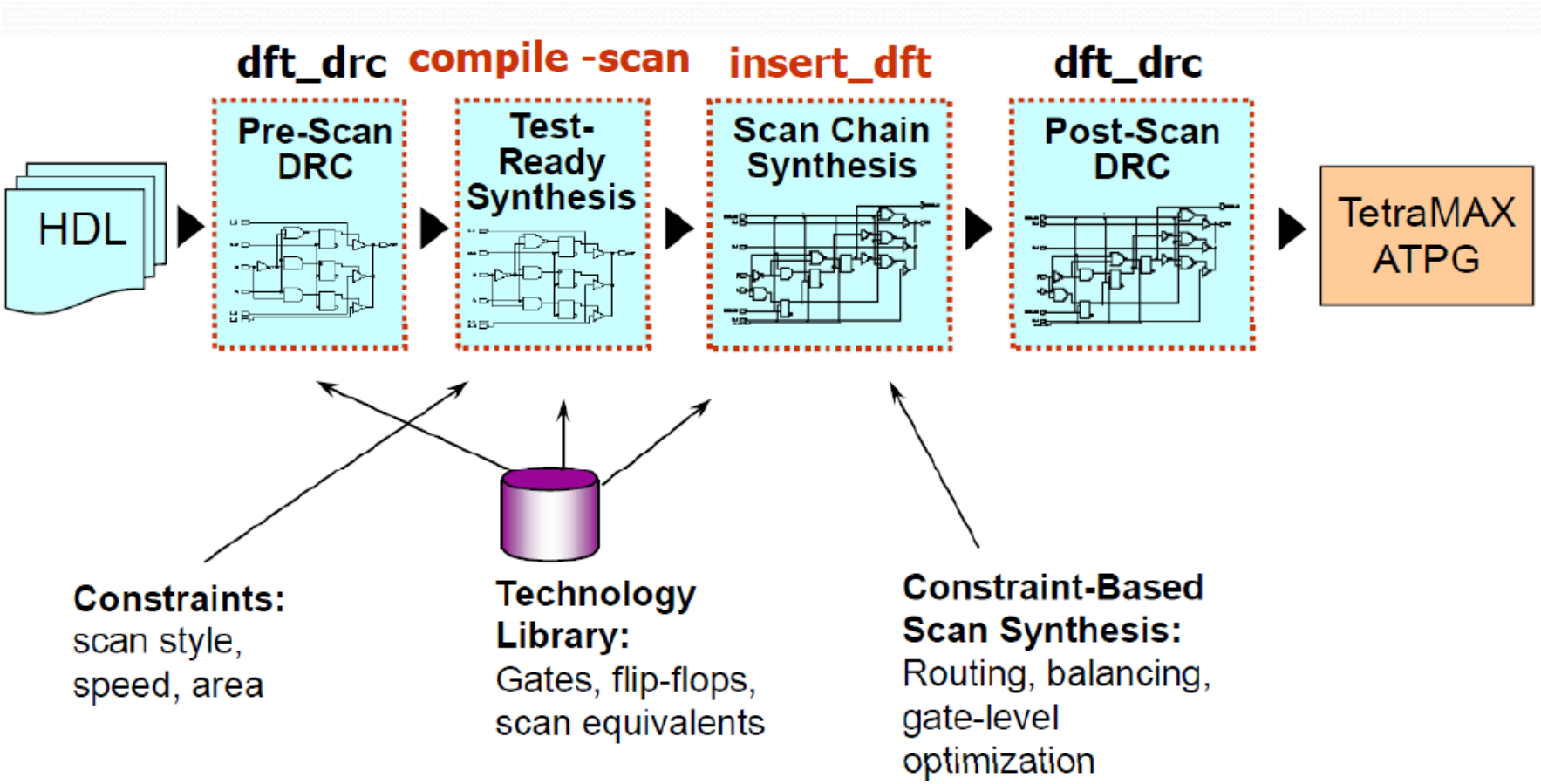
Test Circuit Design Flow



Test Circuit Design Flow



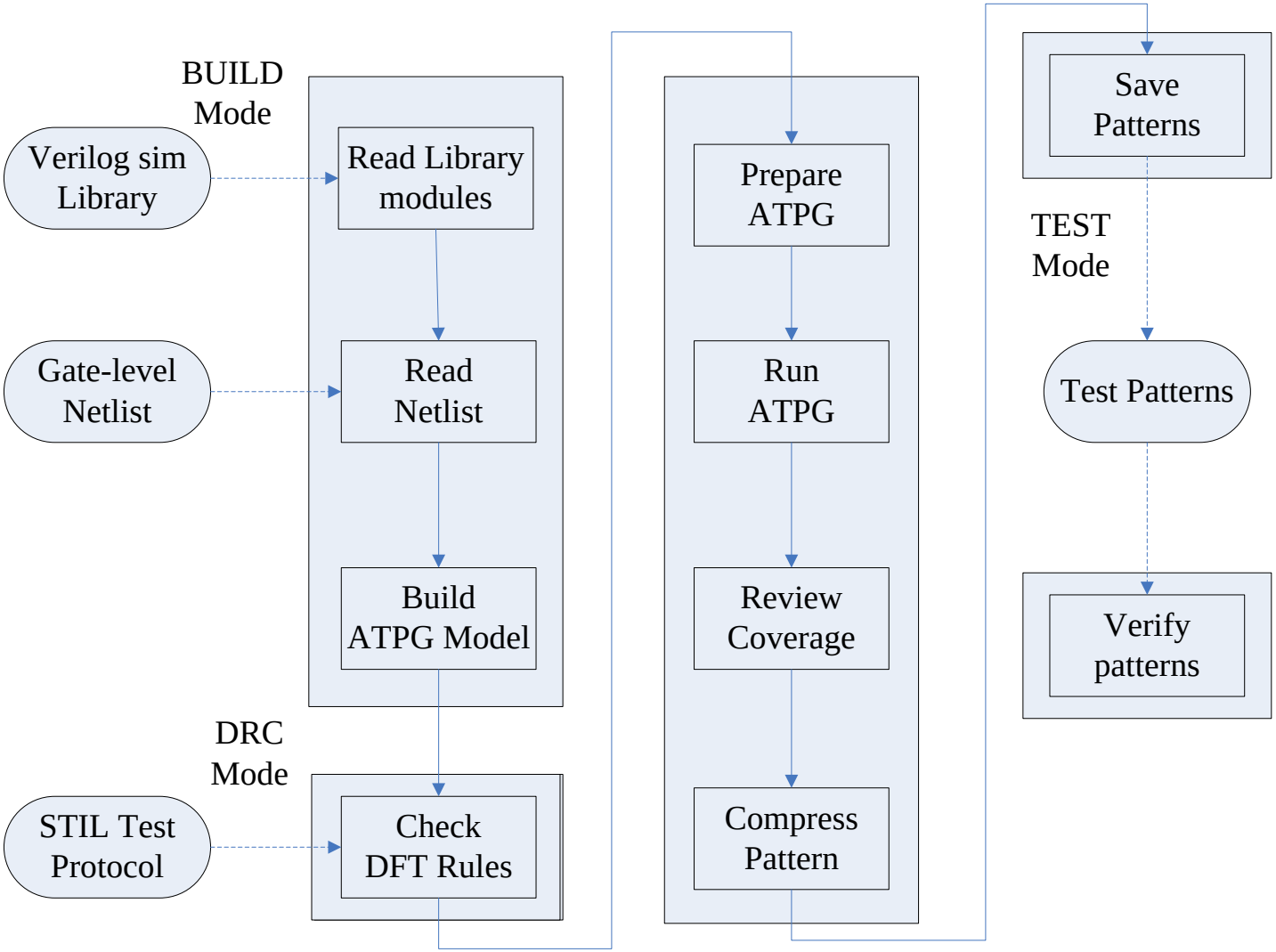
Overview of DFT Compiler Flow



Automatic Test Pattern Generator(ATPG)

- Generate the test patterns for target fault model and keep the number of test pattern as small as possible
- We will use Synopsys **Tetramax** EDA tool to automatically generate test pattern
- Besides, “fault coverage” of the testing circuits can also be estimated

ATPG Flow in TetraMAX



TetraMAX ATPG Command Modes

- BUILD mode:
  - Initial mode
  - Read in design, libraries, models
  - Construct ATPG simulation model in preparation for DRCs
- DRC mode:
  - Testability Design Rule Checks (DRCs) are performed
  - Successful conclusion of DRCs sets mode to “TEST”
- TEST mode:
  - ATPG, fault Simulation, Fault Diagnosis are performed
  - Test program files, simulation testbenches, etc. written out

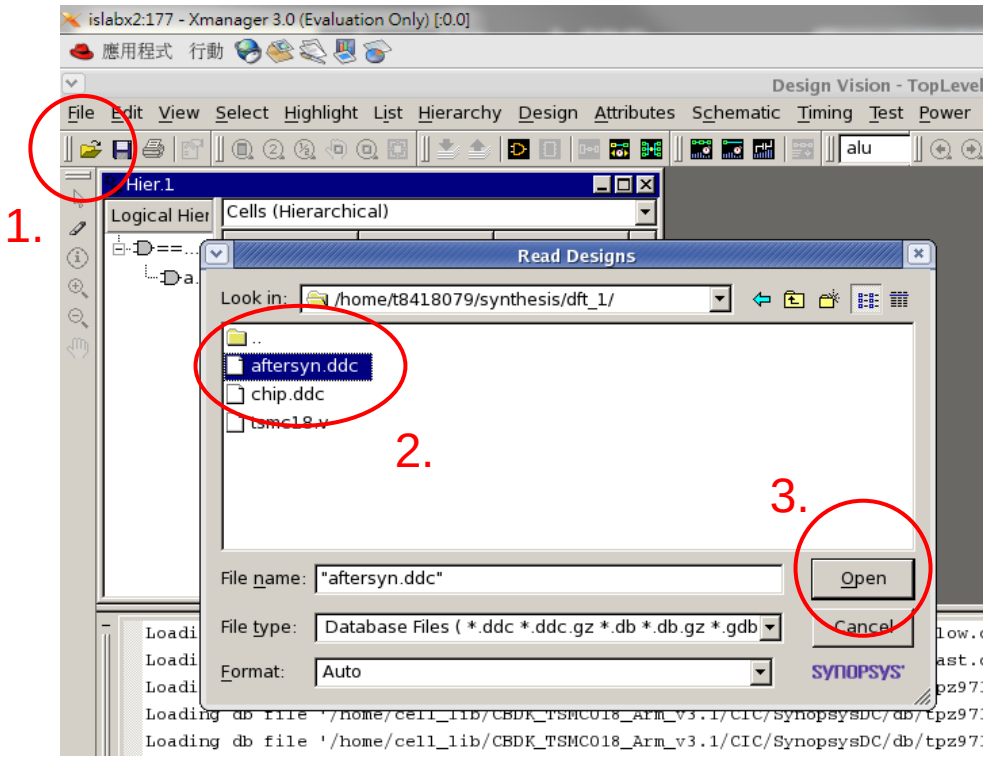
Lab practice

- Objective
  - Understand the baseline DFT Compiler flow.
  - Learn how to use TetraMAX after we generated the STIL procedure file and synthesized netlist from DFT Compiler.

Practice

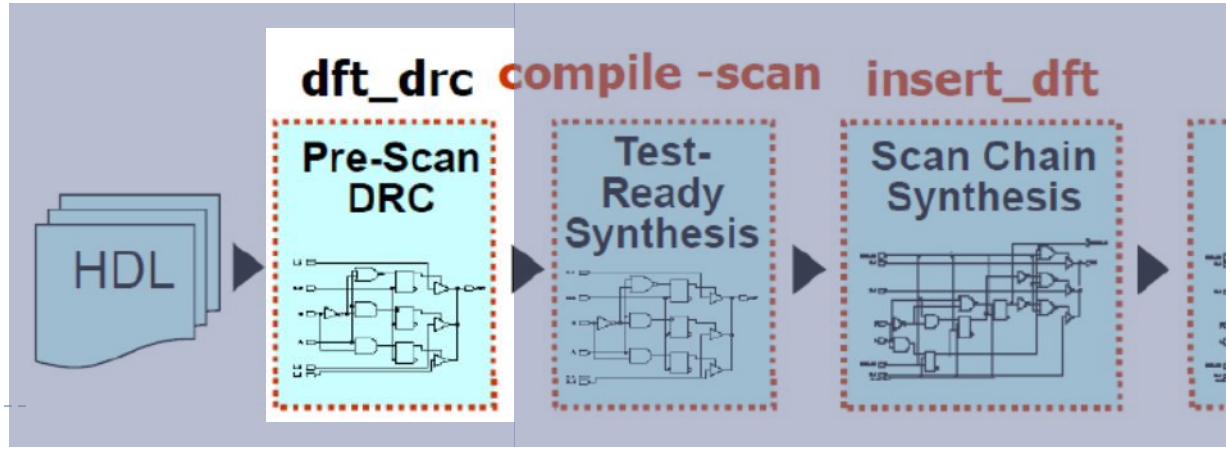
- copy dft folder from /home/standard/multimedia2023/dft .
- Paste to your folder
- Open Design vision

  Command: dv
- Read aftersyn.ddc file



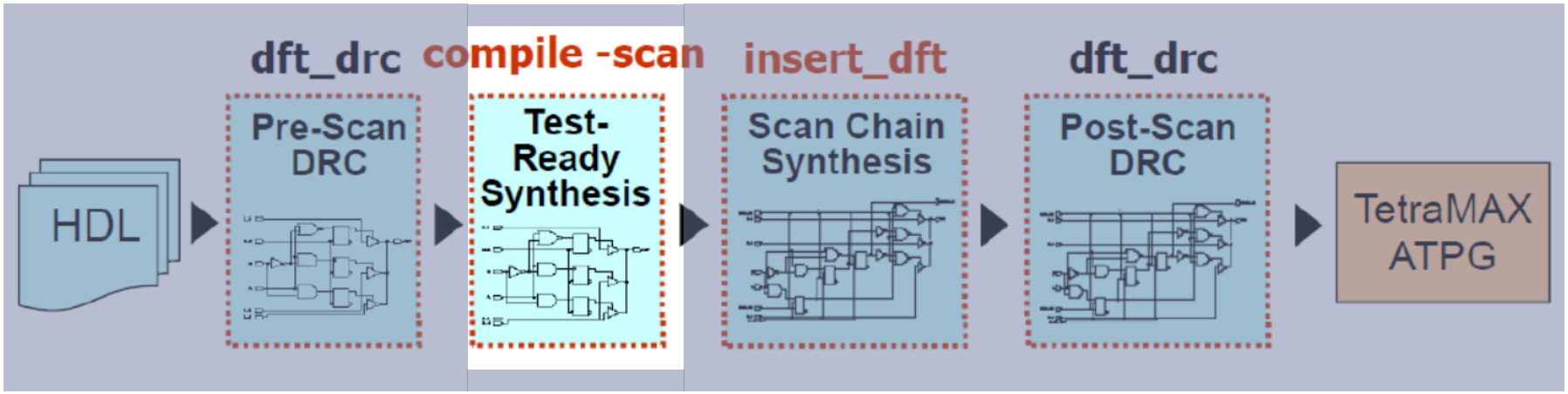
Scripts for DFT

- Create test protocol and Perform pre-DFT DRC
  - create_port -dir in SCAN_IN
  - create_port -dir out SCAN_OUT
  - create_port -dir in SCAN_EN
  - set_dft_signal -view exist -type ScanClock -timing {45 55} -port clk
  - set_dft_signal -view exist -type Reset -active_state 1 -port reset
  - create_test_protocol
  - dft_drc



Scripts for DFT(cont.)

- Perform Test-Ready Compile
  - compile -scan -map_effort high -area_effort high -boundary_optimization

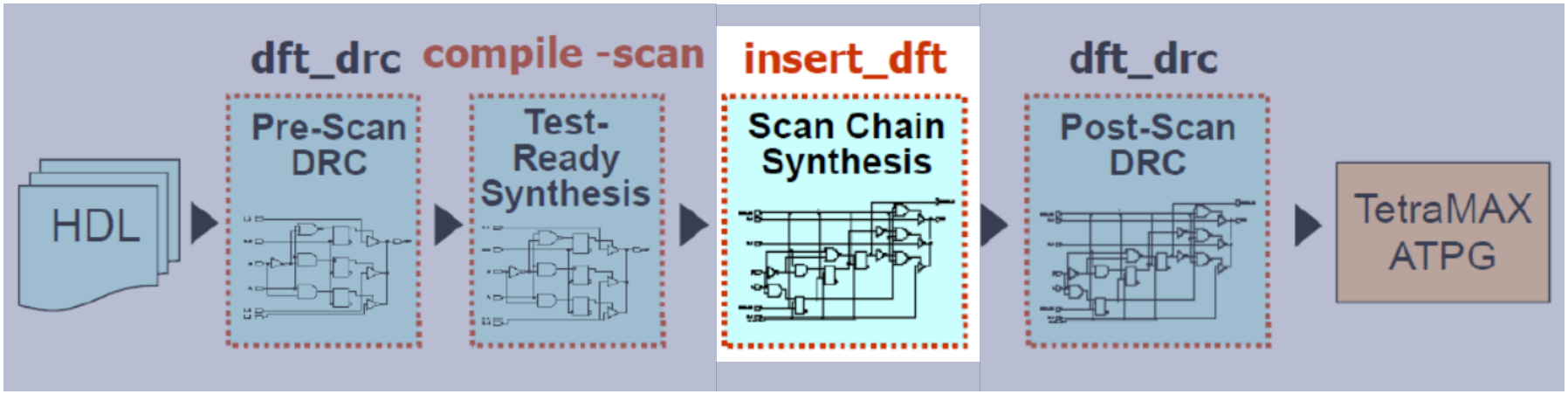


Scripts for DFT(cont.)

- Specify test components
  - set_scan_configuration -chain_count 1 -clock_mixing mix_clocks_not_edges -internal_clocks single -add_lockup false
  - set_dft_signal -view spec -port SCAN_IN -type ScanDataIn
  - set_dft_signal -view spec -port SCAN_OUT -type ScanDataOut
  - set_dft_signal -view spec -port SCAN_EN -type ScanEnable -active_state 1
  - set_scan_path chain1 -scan_data_in SCAN_IN -scan_data_out SCAN_OUT

Scripts for DFT(cont.)

- Preview the scan synthesis, if it is ok, then insert scan.
  - preview_dft -show all
  - insert_dft

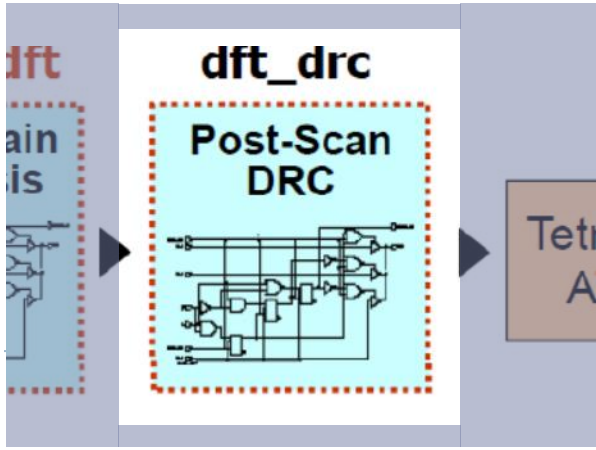


Scripts for DFT(cont.)

- Check scan rules after scan inserting and report the result.
  - dft_drc -coverage_estimate

  What was the test coverage reported? __98__%

- How many scan chains dis you get? ___1_
  - report_scan_path -view existing_dft -chain all
- How many flip-flops are in each chain? __75___
  - report_scan_path -view existing_dft -cell all

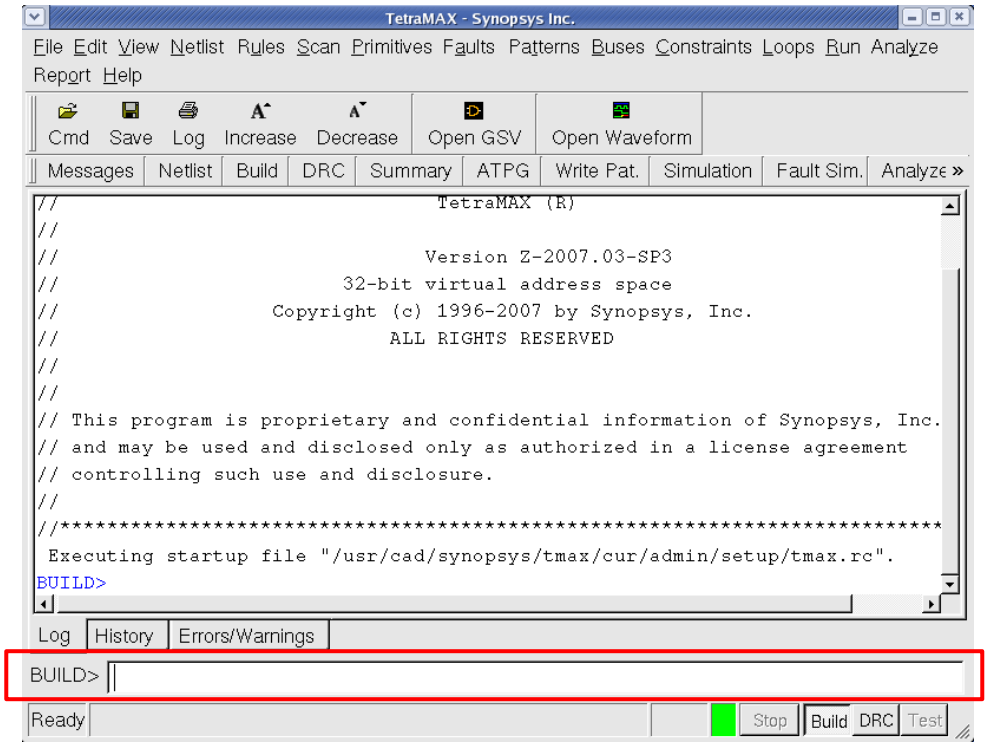


Scripts for DFT(cont.)

- Save synthesized netlist & STIL procedure file
  - write -format verilog -hier -out chip_scan.vg
  - write_test_protocol -out chip_scan.spf
  - Now you can close Design Vision

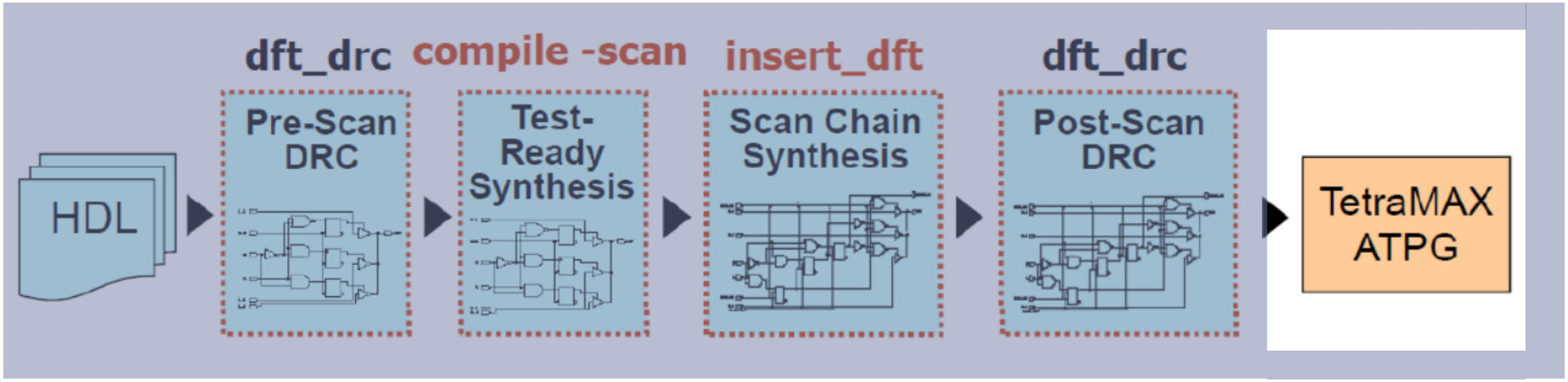
How to open Tetramax

- In terminal, please key in “tmax &”, and then Tetramax GUI will display.
- Fill the scripts in “BUILD>” column



Scripts for Tetramax

- Set messages
  - set_messages -log chip.log -replace
- Read in library and netlist file
  - read_netlist tsmc18.v
  - read_netlist chip_scan.vg



Scripts for Tetramax(cont.)

- Build the fault simulation model
  - run_build_model STI_DAC
- Run design rule checking using STIL procedure file
  - set_rules C4 ignore
  - run_drc chip_scan.spf
- Add fault list
  - add_faults -all

Scripts for Tetramax(cont.)

- Run ATPG
  - set_pat -internal
  - run_atpg -auto
  - set_faults -fault_coverage
  - set_faults -summary verbose
  - report_summaries
- How many patterns needed? __125___
- What is the total fault count? _3658____
- What is the test coverage? ___98.06__%
- What is the fault coverage? __95.22___%

```
    Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT       1130
  detected_by_simulation         DS      (1060)
  detected_by_implication        DI        (70)
Possibly detected                PT          0
Undetectable                     UD          7
  undetectable-redundant         UR        (7)
ATPG untestable                  AU          5
  atpg_untestable-not_detected   AN        (5)
Not detected                     ND          0
-----------------------------------------------
total faults
test coverage
fault coverage
-----------------------------------------------
          Pattern Summary Report
-----------------------------------------------
#internal patterns
    #basic_scan patterns
-----------------------------------------------
```

$$\text{Test Coverage} = \frac{DT + PT}{\text{total fault} - UD}$$

$$\text{Fault Coverage} = \frac{DT + PT}{\text{total fault}}$$

Thanks for your attention !