Lab2
Scan Chain Insertion and ATPG Using DFTADVISOR and FASTSCAN

Prof: Chia-Tso Chao
TA: Yu-Teng Nien
2018-05-28

Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

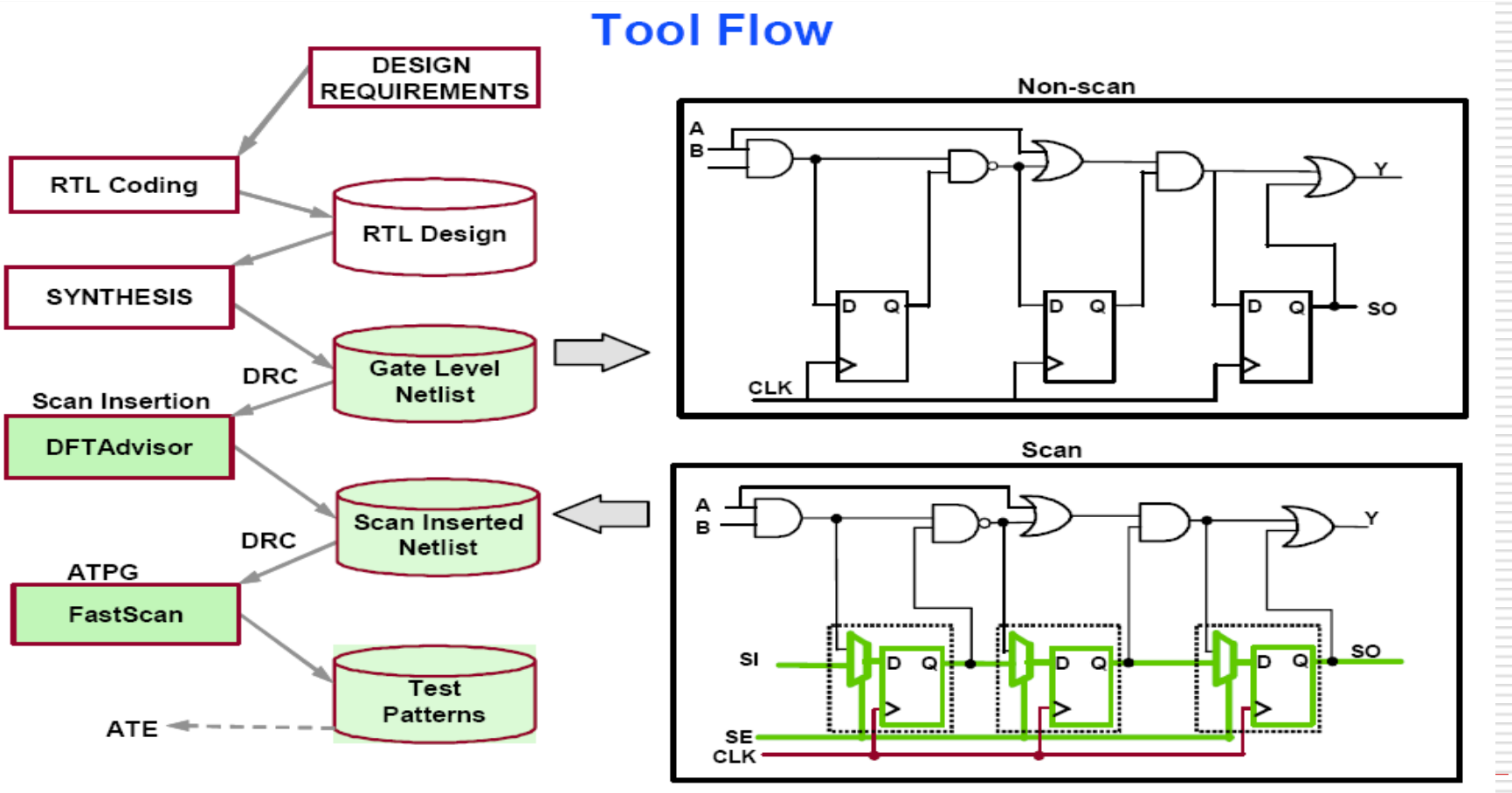
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

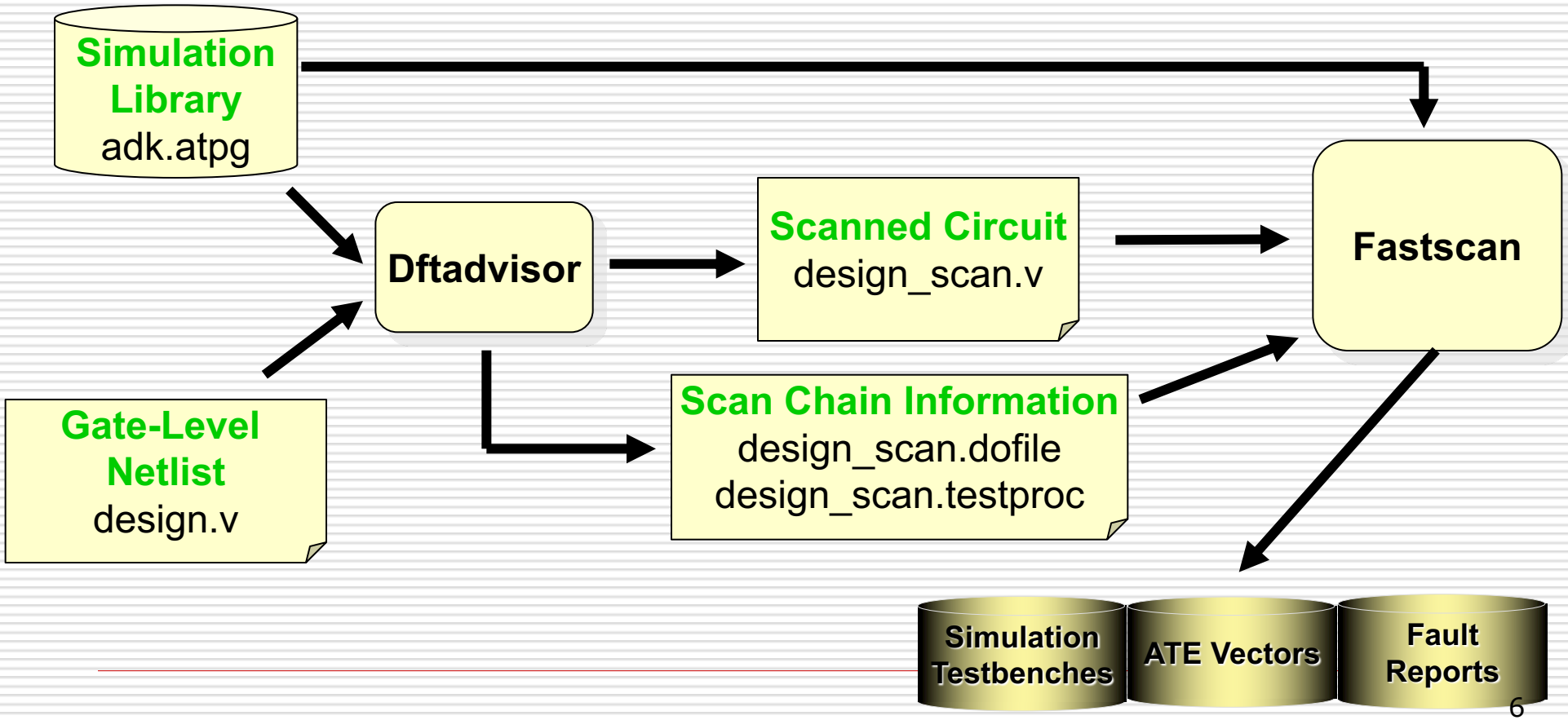
Introduction

- This lab focuses on ATPG with tools from different EDA vendor
  - Synopsys
  - Mentor Graphics
- Dftadvisor is used to insert scan chain (basically replace FF with scan FF)
- Fastscan is used to do ATPG and fault simulation

Insert Scan and ATPG Flow



Input/Output Files



Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

Invoke DFTADVISOR

- Read in Verilog source file and assign ATPG library file
  - $ dftadvisor pre_norm_noscan.v -verilog -lib l90sprvt.atpg -nogui
- Default system mode is “SETUP”
  - SETUP>

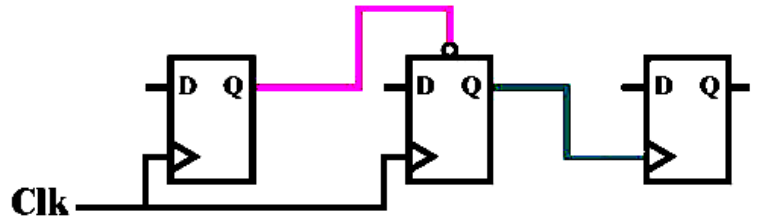
Specify Clock

- Clocks are primary input signals that synchronously change the state of sequential logic elements
  - SETUP> add_clocks 0 clk ← primary input to design

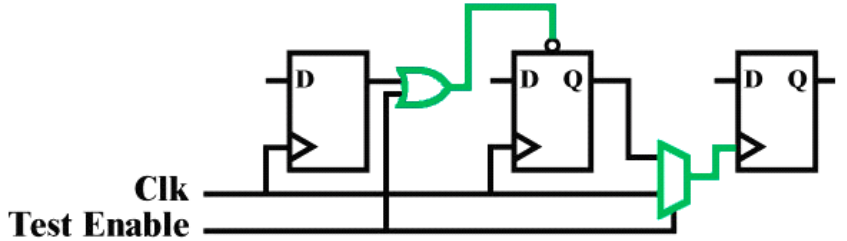
    positive edge-triggered signal
- Check the clock list
  - SETUP> report_clocks

Setup Test Logic Configuration

- Set scan style design
  - Mux_scan : mux-DFF
  - Lssd : level sensitive
  - Clocked_scan : clocked-signal
    - SETUP> set_scan_type m
- Test logic options make clock lines controllable to get a scannable design
  - SETUP> set_test_logic -clock on -reset on
- Verify with report_environment



Non-scannable



Scannable after test logic insertion

Enter Scan Insertion System Mode

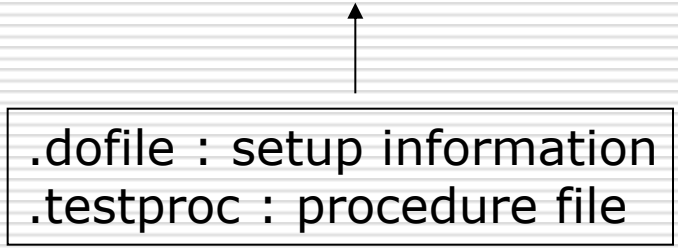
- Enter Scan Insertion system mode (DFT) and perform scan identification
  - SETUP> set_system_mode dft
- Report detailed statistical report of scan identification
  - #Sequential instances
  - #Scannable instances
    - DFT> report_statistics

Insert Scan-Chain and View Report

- Set # of scan-chains to insert and do so
  - DFT> insert_test_logic -number 10
- Report scan-chain information
  - DFT> report_scan_chains
  - DFT> report_test_logic

Output Scanned Design for ATPG

- Write out files and exit DFTADVISOR
  - DFT> write netlist pre_norm_scan.v -verilog -replace
  - DFT> write_atpg_setup pre_norm_scan -replace
  - DFT> exit



Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

Invoke Fastscan

- Specify scanned Verilog file and ATPG library file
  - $ fastscan pre_norm_scan.v -verilog -lib l90sprvt.atpg -nogui

Read Setup Information

- Read setup information from Dftadvisor
  - SETUP> dofile pre_norm_scan.dofile
- Enter ATPG mode
  - SETUP> set_system_mode atpg
- Select fault type: stuck, IDDQ, toggle, transition, path_delay, bridge, etc
  - ATPG> set_fault_type stuck

Generate Patterns (1/2)

- Use “-auto” option to
  - Suggest the best settings possible to generate the most compact patterns with the highest coverage within the lowest time
  - ATPG> create_patterns -auto

Generate Patterns (2/2)

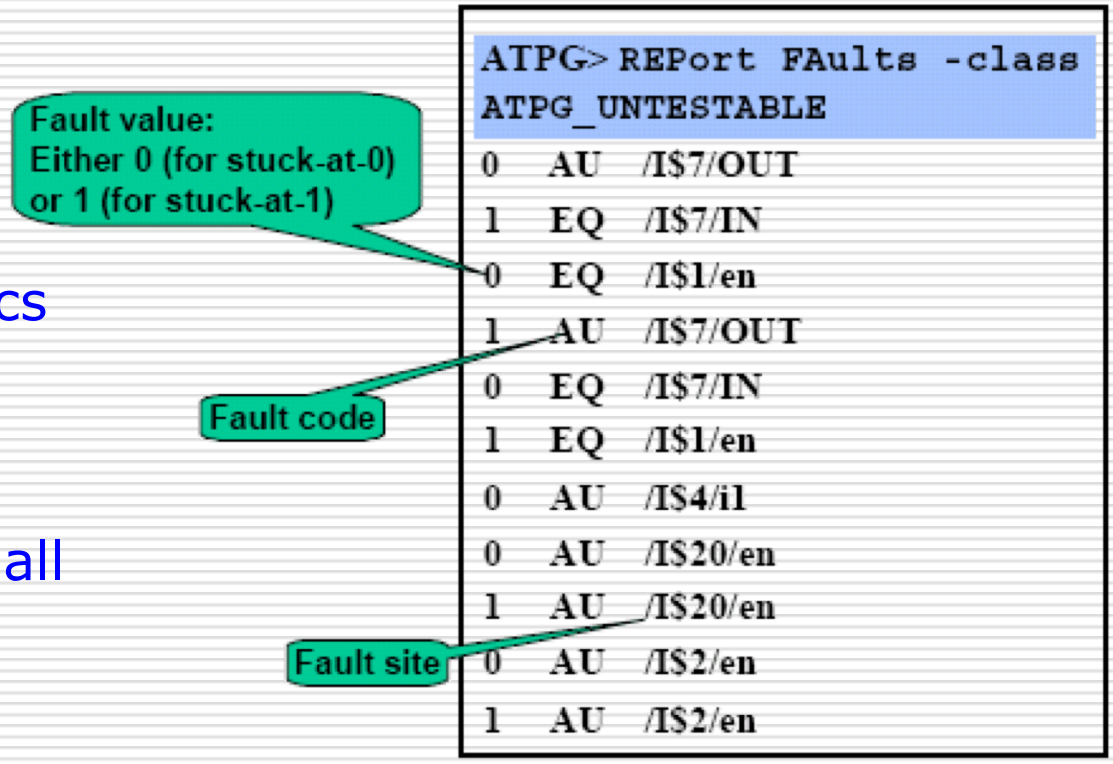
- Without the "auto" option, you can specify your own configurations using these commands
  - ATPG> set_atpg_limits
    -cpu_seconds [integer]
    -test_coverage [real]
    -pattern_count [integer]
  - ATPG> set_abort_limit [integer]
  - ATPG> create_patterns

During ATPG

```
// Simulation performed for #gates = 27550  #faults = 110115
// system mode = ATPG    pattern source = internal patterns
// ------------------------------------------------------------------------
// #patterns  test        #faults  #faults    # eff.     # test      process     RE/AU/abort
// simulated  coverage  in list    detected   patterns  patterns    CPU time
// deterministic ATPG invoked with comb/seq abort limit = 300/100
//  ---       ------     ---       ---        ---       ---         0.10 sec     224/0/0
//  32        82.90%     20861     89030      32        32          0.15 sec
//  ---       ------     ---       ---        ---       ---         0.25 sec     1423/0/0
//  64        91.30%     10501     9161       32        64          0.26 sec
//  ---       ------     ---       ---        ---       ---         0.40 sec     2342/0/0
//  96        95.67%     5187      4395       32        96          0.41 sec
//  ---       ------     ---       ---        ---       ---         0.59 sec     3192/0/0
//  128       98.43%     1887      2450       32        128         0.60 sec
//  ---       ------     ---       ---        ---       ---         0.67 sec     3609/0/0
//  160       99.22%     954       516        10        138         0.68 sec
//  ---       ------     ---       ---        ---       ---         0.68 sec     3609/0/0
//  192       99.69%     357       597        32        170         0.69 sec
//  ---       ------     ---       ---        ---       ---         0.69 sec     3609/0/0
//  224       99.98%     24        333        32        202         0.70 sec
//  ---       ------     ---       ---        ---       ---         0.70 sec     3609/0/0
//  256       100.00%    0         24         3         205         0.70 sec
```

View Report

- Report simulation result and faults
  - ATPG> report_statistics
- Display fault information
  - ATPG> report_faults -all



ATPG Result

Statistics report

Fault Classes	#faults (total)
FU (full)	6078
UO (unobserved)	9 (0.15%)
DS (det_simulation)	5393 (88.73%)
DI (det_implication)	540 (8.88%)
UU (unused)	16 (0.26%)
RE (redundant)	118 (1.94%)
AU (atpg_untestable)	2 (0.03%)

Coverage

test_coverage	99.81%
fault_coverage	97.61%
atpg_effectiveness	99.85%

#test_patterns	180
#simulated_patterns	224
CPU_time (secs)	3.7

Fault Classes - FU: Full

- FU = TE + UT
- TE: Testable
- UT: Untestable
  - Faults which no pattern can exist to either detect or possible-detect
  - Cannot cause functional failures, so they are excluded from test coverage calculation

Fault Classes - Testable (TE)

- DT: Detected
- UD: Undetected
  - Faults that cannot be proven untestable or ATPG_untestable
  - Initial class for testable faults
- AU: Atpg_untestable
  - Due to pin constraint or insufficient sequential depth placed on Fastscan
- PD: Possible-detected
  - Faults with good-machine value being either 0 or 1 and faulty machine value being X in simulation

Fault Classes - Untestable (UT)

- UU: Unused
  - Faults not connected to any circuit observation point
- BL: Blocked
  - Faults blocked by logic on all paths
- TI: Tied
  - Point of the fault value is always same (e.g. AND2 with complementary inputs)
- RE : Redundant
  - Faults undetectable after exhausting all patterns and need special analyses to verify redundancy
  - ATPG> identify_redundant_faults

Test Coverage Formula Comparison

- TetraMAX

possible detected

default 50%

default 0

$$\text{test\_coverage} = \frac{DT + (PT * \text{posdet\_credit})}{all\_faults - (UD + AU * \text{au\_credit})}$$

- Fastscan

default 50%

$$\text{test\_coverage} = \frac{DT + (PD * \text{posdet\_credit})}{\text{testable}} * 100$$

$$\text{fault\_coverage} = \frac{DT + (PD * \text{posdet\_credit})}{\text{full}} * 100$$

$$\text{ATPG\_effectiveness} = \frac{DT + UT + AU + PU + (PT * \text{posdet\_credit})}{\text{full}} * 100$$

Testable=DT+PD+AU+UD Untestable=UU+TI+BL+RE

Save Patterns

- Save patterns just generated
- Various format including binwgl, ctl2005, stil2005, stil999, Verilog, VHDL, wgl, zycad, tstl2, utic
  - ATPG> save patterns pre_norm_scan.pat -verilog –proc –replace
  - ATPG> save patterns pre_norm_scan_tstl2.pat -TSTL2 –replace
  - ATPG> exit

Toshiba Standard Tester Interface Language 2

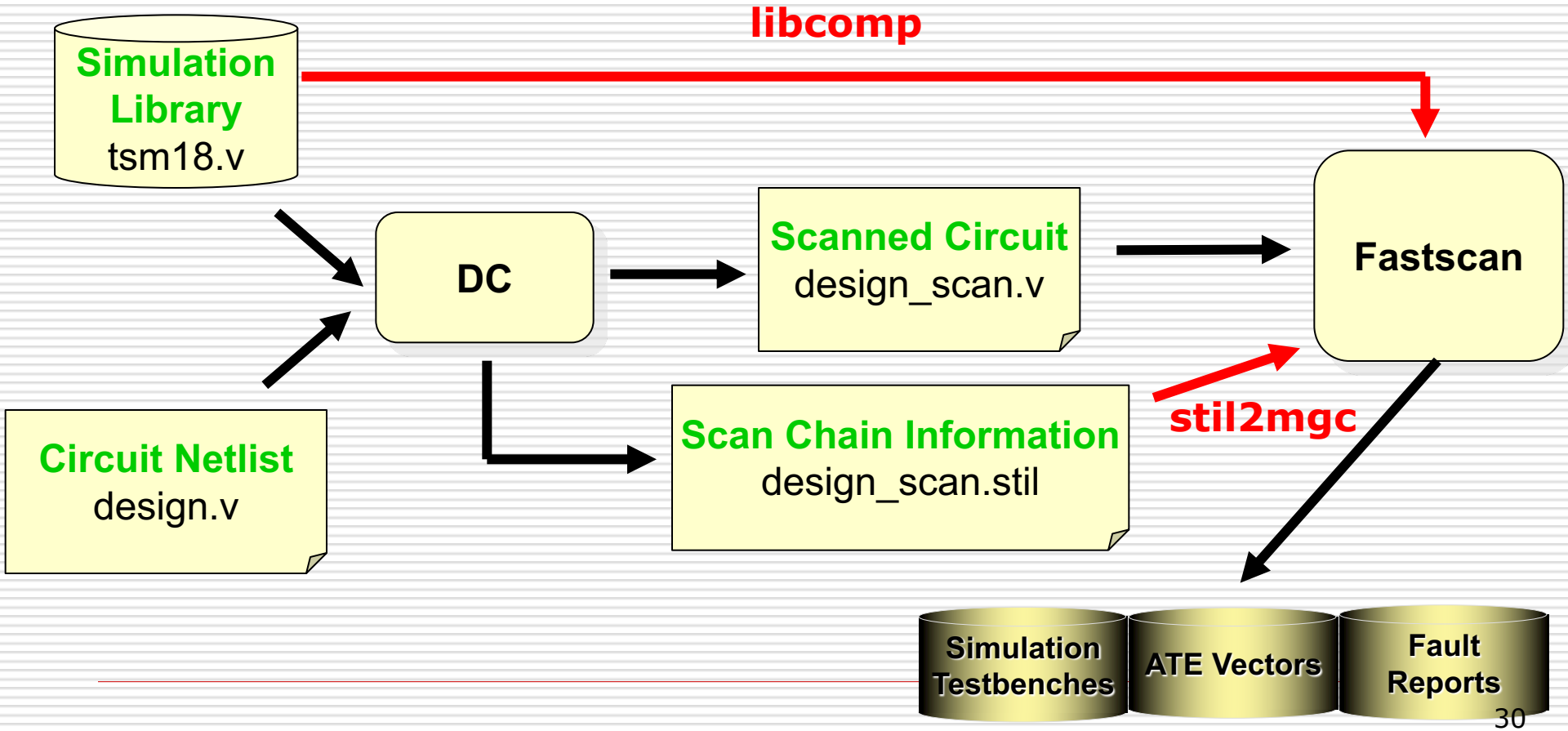
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

Mixed Flow

- Synopsys Design Compiler is better at mapping from RTL code to gate-level circuit
- Some cases in industry field also use
  - Design Compiler to synthesize gate-level netlist and insert scan-chain
  - Fastscan to perform ATPG

Input/Output Files



Input Files Required in Design Compiler's Flow

- Library file needs to be converted
  - From .v to .atpg
- The detailed information about scan chain needs to be converted
  - From .stil to .dofile
- Scanned design

Convert Library File

- Fastscan uses different library from TetraMAX, so use 'libcomp' command to convert l90sprvt.v to l90sprvt.atpg
  - $ libcomp l90sprvt.v
  - SETUP> add model -all
  - SETUP> set optimization on
  - SETUP> set learning on
  - SETUP> set sys mode tran
  - TRANSLATION> run
  - TRANSLATION> write lib l90sprvt.atpg -rep
  - TRANSLATION> exit

Convert STIL File

- Use command 'stil2mgc' to convert STIL file into dofile and test procedure file for Fastscan
  - $ stil2mgc pre_norm_scan.stil
    - It will generate pre_norm_scan.stil.do and pre_norm_scan.stil.proc

Perform ATPG using FASTSCAN

- Read scanned circuit from design compiler to perform ATPG
  - $ fastscan pre_norm_scan.v -verilog -lib l90sprvt.atpg -nogui
  - SETUP> dofile pre_norm_scan.stil.do
  - SETUP> set_system_mode atpg
  - ATPG> create_patterns -auto
  - ATPG> report_statistics

Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mixed Flow
- Lab

Lab Goal

- Compare the following during ATPG using the DC+TMAX, DFTA+FS and DC+FS flows
  - Total fault number
  - Test coverage
  - Pattern count
  - Run time
- Run on circuit “pre_norm_noscan.v”, and show a table like next slide

Result

Flow	#Faults	Test Coverage	#Patterns	Run time
DC+TMAX	71298	100%	138	0.83s
DFT+FS	122072	100%	201	0.66s
DC+FS	75208	100%	203	0.51s

References

- Mentor Graphics User Guide
- Synopsys TetraMAX User Guide