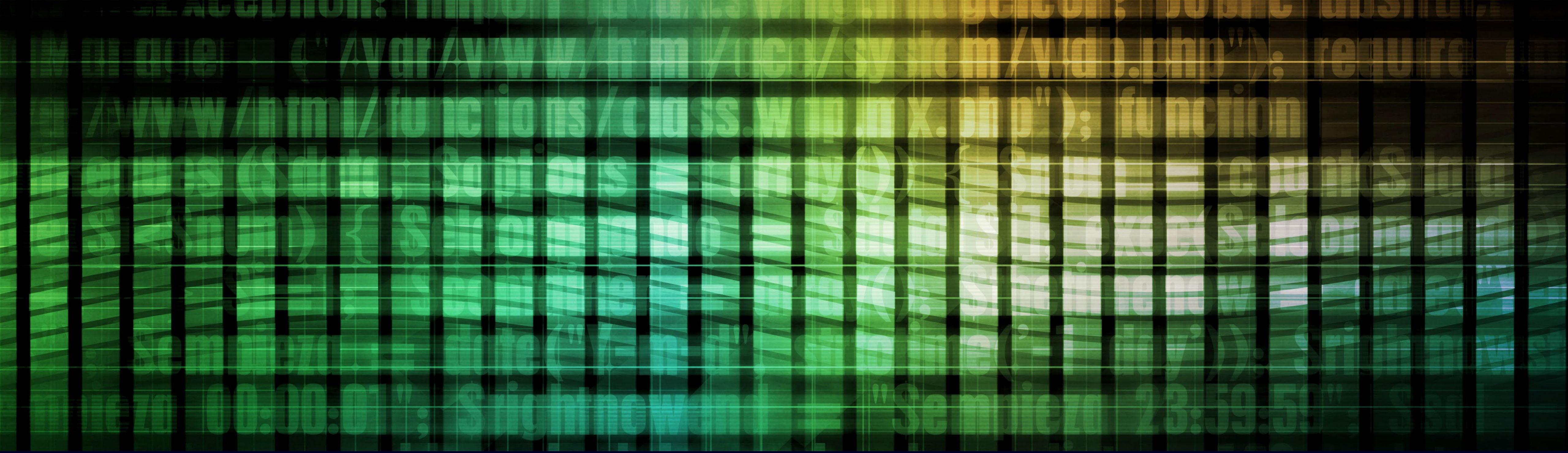


Streaming Scan Network (SSN) Post silicon features (Burn-in and LVI/LVP)

Peter Orlando

Tessent® DFT Product Management | Tessent® Silicon Lifecycle Solutions

www.siemens.com/eda

SIEMENS

Burn-in

SIEMENS

Silicon aging techniques : Burn-in patterns

- Silicon aging technique
- Burn-in patterns to predict silicon aging part IC manufacturing process
  - Commonly used during HTOL testing
- Historically EDT used to drive pseudo random pattern into chains
  - Toggle EDT_update and drive the channels
- Custom observability used proof of life / activity
  - Counter or path to edge through GPIO

How it works

- SSH hardware configured in infinite shift mode
  - No Capture
- Packetized deliver of burn-in payload to all SSH’s (default)
  - Optionally target specific SSH
- Full scan word delivered to each SSH
  - Equal to number of input_channels / scan chins
- Payload sequence 1100 (default – four packets)
  - Packet #1, #2 drive all bus inputs to 1’s, packets #3, #4 drive all bus inputs to 0’s
  - Triggers pseudo random data out of the EDT

Payload

- Configurable with new command (optional)

```
set_burnin_options [ -ssh_icl_instances <ssh_icl_instances> ]
                   [ -sequence <sequence> ] [ -repetitions { load_chains | <int> } ]
```

-ssh_icl_instances	target specific SSH ICL instances (**default all**)
-sequence	arbitrary binary sequence (**default 1100**)
-repetitions	(1)*load_chains* writes verification testbench or <int> times to repeat payload (**default 1**)

(1) See limitations slide for details on limited functionally of burn-in verification testbench

Creating burn-in patterns (Recommendations)

- Retargeting flow (option 1)
  - Large designs that follow Tessent hierarchical flow
  - Benefits
    - Requires least amount of machine memory
    - Improved run-time when compared to running flat
- Flat ATPG flow (option 2)
  - Designs that run flat ATPG
  - Benefits
    - Reuse same setup as ATPG

Writing burn-in patterns

- STIL patterns
  - Write setup pattern files (apply once)

    write_patterns tstsetup.stil –stil **–test_setup only** –pattern_set **burnin_loop**

    write_patterns ssnsetup.stil –stil **–ssn_setup only** –pattern_set **burnin_loop**

  - Write payload pattern file (loop on ATE)

    write_patterns payload.stil –stil **–test_payload** –pattern_set **burnin_loop**

- Verilog testbench

  write_patterns vtestbench.v -verilog -pattern_sets **burnin_loop**

SIEMENS

LVI / LVP (LVX)

SIEMENS

Highlights

Functionality

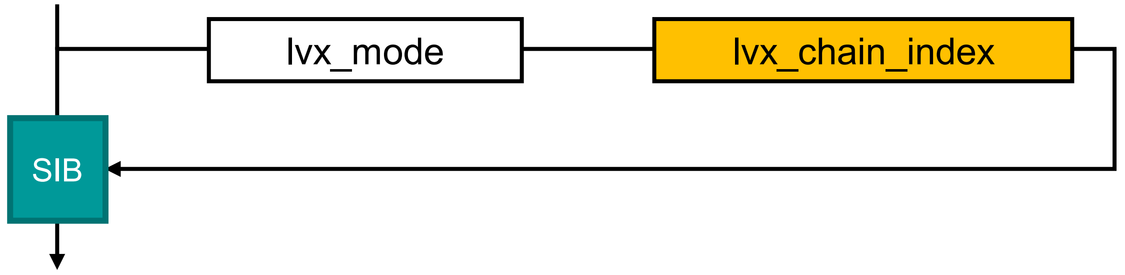
- Enable scan chain failure analysis using Laser Voltage Imaging (LVI) and Laser Voltage Probing (LVP)
- Ability to shift from the tester any repeating sequence (e.g., “1100”) through SSN → EDT → chain(s)
- Sequence can be broadcast to all chains of an EDT, or 1 specific chain (selectable on tester by patching setup)
- After applying setup sequence, tester loops on payload pattern set (lvx_loop) to apply desired sequence on internal chains with no interruptions. No ssn_end applied between iterations as needed for ATPG patterns.

Generating EDT with the LVX hardware

- Specify the access code for the EDT_LVX feature
- Use the following DFT Spec wrapper/properties to generate EDT LVX hardware:

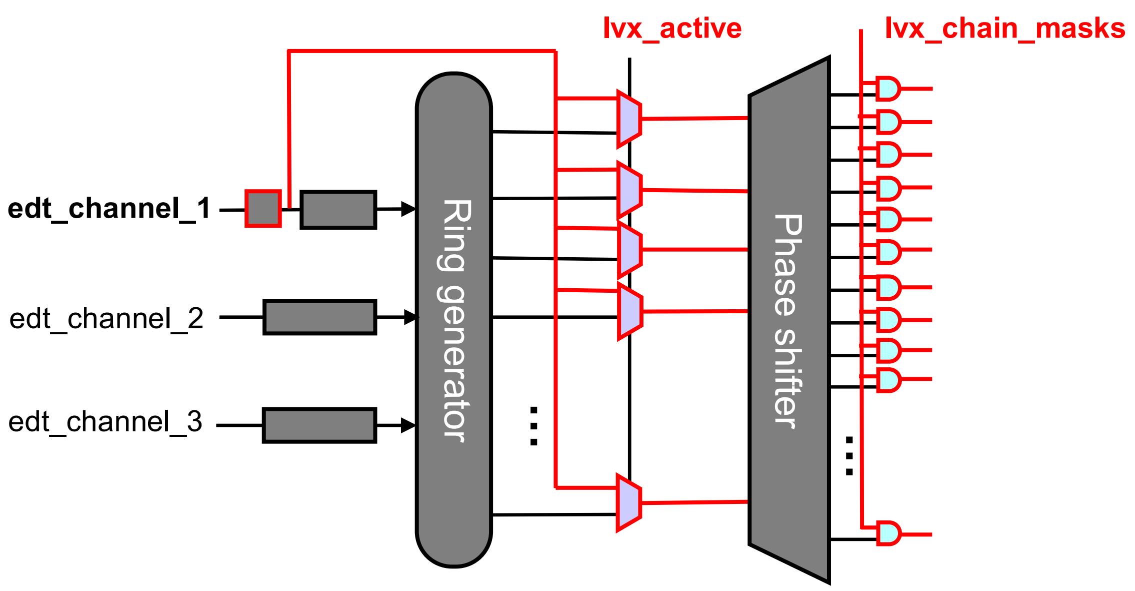
```
DftSpecification(module_name, id) {
   EDT {
      Controller {
         LVxMode {
            present: on | off | auto
            enable_one_chain: on | off
         }
      }
   }
}
```

- New TDRs:
  - "**lvx_mode**" and "**lvx_chain_index**"
  - EDT Controller will have an IJTAG scan interface
  - Both TDRs represented by **iWriteVars** in the PDL so they are annotated in the pattern set and are patchable on the tester

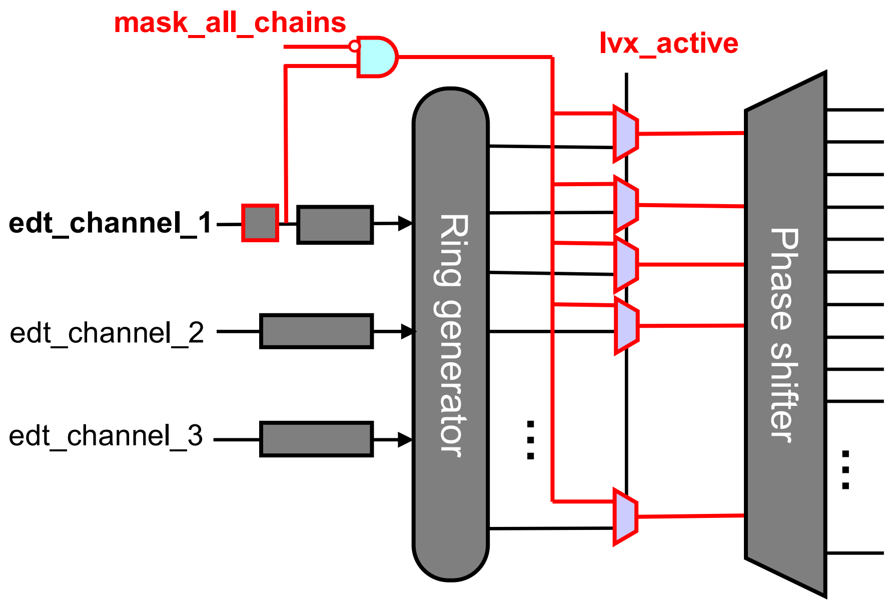


⬜	Mandatory data registers
🟨	Optional data registers (enable_one_chain = "on")

LVX hardware for EDT channel input



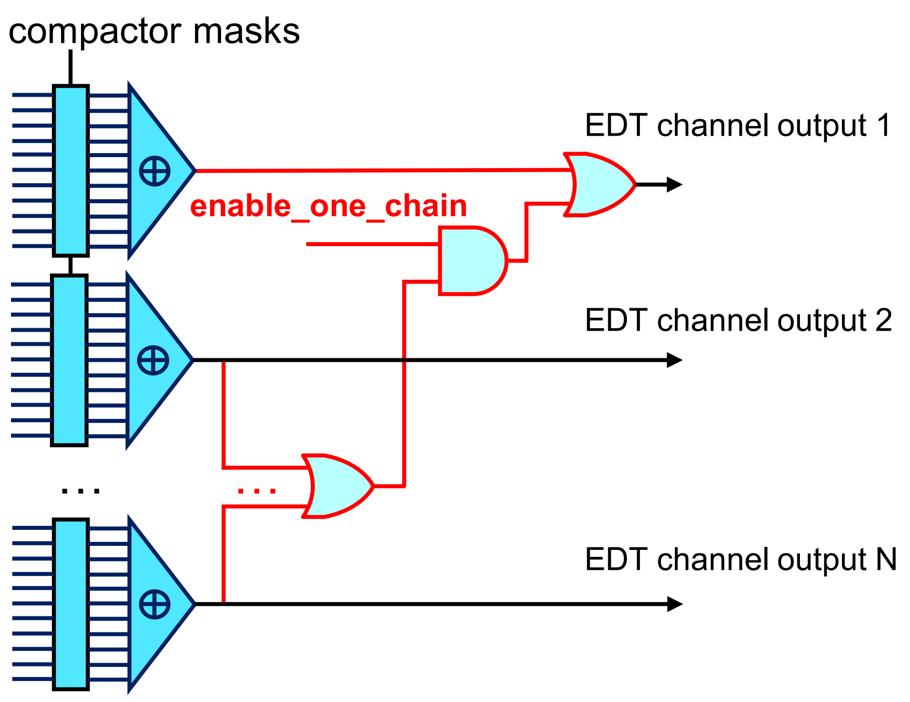
LVX hardware **with** “enable_one_chain”



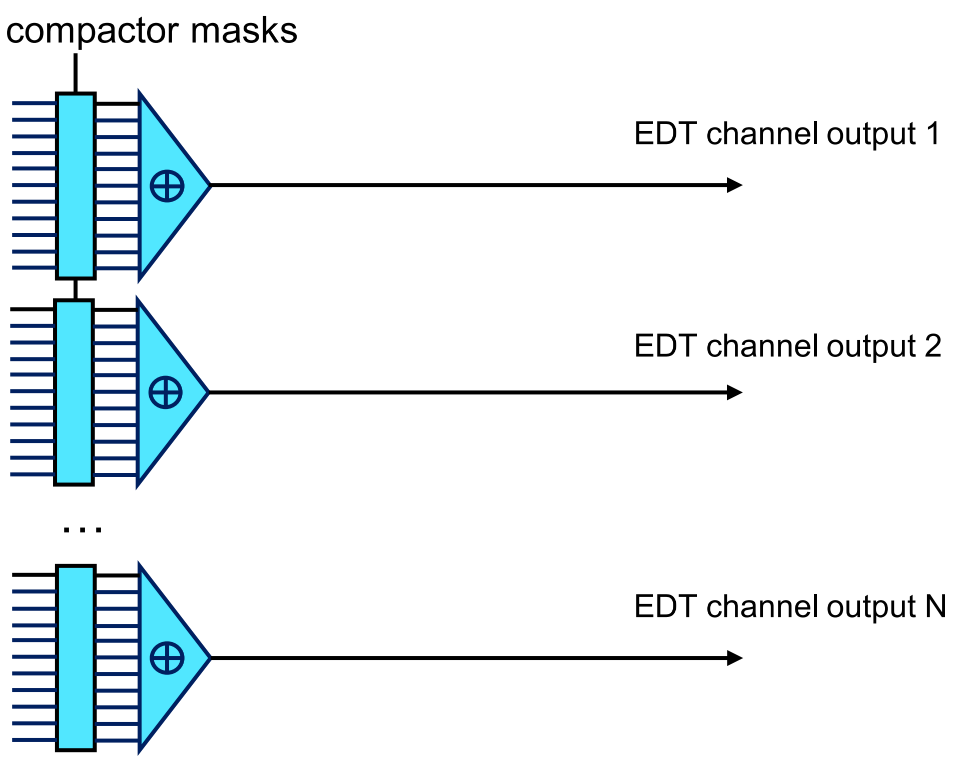
LVX hardware **without** “enable_one_chain”

SIEMENS

LVX hardware for EDT channel output



LVX hardware **with**
"enable_one_chain"



LVX hardware **without**
"enable_one_chain"

SIEMENS

LVX pattern generation

- When writing the LVX production patterns, separate the setup sequence from the payload on which you will loop
  - The setup sequence consists of **test_setup** + **ssn_setup** procedures:

```
write_patterns <lvx_loop_setup.stil> -all_setup_only -pattern_set lvx_loop -stil
```

  - The pattern set on which to loop:

```
write_patterns <lvx_loop_payload.stil> -test_payload -pattern_set lvx_loop -stil
```

Thank You

Disclaimer

Backup

SIEMENS