TESSOLVE
Delivering Excellence in Semiconductor Engineering

Volume - 4
Issue - 1
Jan - 2016

FIRST BIN

a newsletter for the semiconductor engineering community

Contents

Printed and published on behalf of

Tessolve Semiconductor Pvt. Ltd.
Plot # 31, Electronic City,
Phase 2, Bangalore 560 100,
Karnataka, India.
Tel: +91 80 4181 2626
Your kind enquiries / feedback solicited at,
sales@tessolve.com / news@tessolve.com
www.tessolve.com

FROM THE PRESIDENT'S DESK

Srinivas Chinamilli

Dear Customer,

Wish you all Health, Happiness and Prosperity in the New Year!

It has been a tough year with a lot of consolidations in the semiconductor space. We are poised to grow 15% this fiscal year in spite of the slowdown. Thanks for your support! We are continuing to invest in hiring quality engineering resources and building our capacity.

We have added ESD &Latch up capability at Tessolve. The ESD & Latchup testers will be part of our characterization lab and are fully capable of testing ESD for HBM (Human body model), MM (Machine model) and CDM (Charge data model).I am proud to say that ours is the only facility with ESD test capabilities in India. I welcome you to work with us to address your ESD test needs.

We have just announced concessional rates for using our Characterization lab. Please contact our sales if you need more details. We have also gone a step further to allow free access to our Characterization lab for educational and research institutes for addressing their test needs. We would definitely like to do our part in encouraging the semiconductor eco system in India.

We have just incorporated Tessolve Malaysia and opened an office in Penang. We are in the process of expanding our operations there to better serve our customers in that region.

I would like to congratulate D Vijayakumar for his paper presentation on Library and Data Management at CDN Live India 2015. I would also like to congratulate Banukumar on his paper presentation "MEMS Testing of gyros and accelerometers" at NI Day conference at Singapore.

We look forward to working with you a lot more and strive even harder to provide value add services for you in 2016.

TESSOLVE SHOWCASE

Pattern Delivery Automation (PDAT) Tool Introduction

Jagan Mohana Rao & Sooraj K V - Test Engineer, Basavaraj - Software Engineer

The Pattern Delivery Automation is an offline checking tool for ATPG patterns; this tool is an extension of the Silicon bump-out pattern delivery checklist, which is used to validate ATE worthiness of DFT provided bins targeted towards the Advantest 93K test platform. This tool is helpful to check the Test Setup and Scan Operation for every ATPG vector. The pins of interest will be Scan Clock Pins, Scan In and Scan out Pins, Scan Enable Pin, JTAG or other pins (Cxo, mode_0, mode_1, Srst_n, resin_n).

How this tool works?

ATPG Vectors will consist of 2 parts Test Setup and Scan Operation.

- Test Setup
  During Test setup, the vector environment is setup (like path/chain access). Test setup will be done through JTAG programing. During Test setup the Scan chains won't have any operation/access. After Test Setup is done, Scan Operation will start.
- Scan Operation
  During Scan operation, loading-unloading of scan chains + capture activity commences.

Check Done during Test setup

TDI check for one or more toggles during test setup and make sure TDI toggle count is greater than TMS otherwise report it as an ERROR. Check for toggle and report toggle count. For TCK clock toggling is to be checked

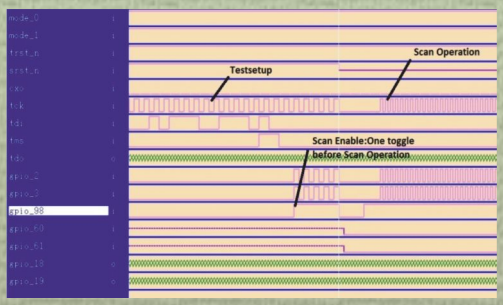

Fig 1. ATPG Vectors with Test Setup and Scan Operation

up. The Resin, Trst_N and Srst_N should have one low pulse and stays high (RH pulse) till test setup ends. Resin (in this design) will start toggling as a scan input pins after test setup. Cxo-CLK toggles checked. The Mode should have one RH/RZ pulse and stays high/low till end of pattern. For Scan-Enable the logic should be low during test setup; check for one RZ pulse before test setup ends. Scan clock/Scan In should not have any toggle activity during Test setup. Scan Out should not contain any compares during Test setup.

Scan Operation Check Points using tool

- **Scan Clk**
  Toggles on all scan-clocks.

- **Capture Pulses**
  There should be at least one capture pulse among scan clock pins during capture period (Scan Enable pulsing to logic Low). During scan chain loading check 'Scan Clock' & 'Scan In' are toggling. During 'Scan Enable Low' capture pulse, all clocks are DEAD (driving 0). There should be a capture pulse in one or more clock pins. During first scan loading, if there is any compare on the scan-outs, then report it as error. After second capture pulse, check for Compares throughout the pattern on SO pins (catches dummy patterns with no compares).

Advantages of the tool

- No more manual sanity checks on SAF bins needed. All sanity checks automated (ex. no-compares, CLK toggle, reset toggle).
- 200+ scan SAF (stuck at faults) patterns generated by DFT
- TE's check 1 bin/core manually using timing diagram with this tool, all SAF bins can be checked up.
- Need not login to ATE or offline simulators, Can be run on a standalone LINUX Xterm.
- Reduces a day of manual effort on limited set of bins to 1.5-2 hour automated effort on all SAF bins.

Software Development Process Model

Ajayvignesh Manonmani Velumani - Sr Test Engineer

There would be days when you are developing your own small scale software on your own way. You would be expert in delivering the software in whatever way the requirements are put through. You would zip-zap the delivery in any zig-zag way you like. Everything would go fine and all deployments would be so successful to you. Suddenly there may come a day when you are put-up with a large scale software development task. And while developing the new project, you will start feel like things aren't going fine as it used to be!

What would have gone wrong there?

It would be the usual answer from experts of software development. "You own method / model of software development doesn't fit in large scale application. You need better process model to develop your large scale software".

What is a better process model?

Well, a good process model would help you work on any large project and with any large team, letting you focus mainly on your goals instead of worrying about other unnecessary activities of the project; it would contain best practices of software development stages to achieve your goal.

Could you give example of such?

There are numerous of such models are in practice and people choose the one which fits them the best. Few popular process models are

1. Waterfall
2. Agile
3. Spiral
4. RAD Model etc..,

Jargons? What are those?

Sure I will give you a description of those. You can choose the one which may fit your needs...

1. Waterfall model

Waterfall model lets your project flow through well-known sequential order as requirement gathering & analysis à system design à implementation à testing à deployment of system à maintenance. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements

2. Agile Model

Agile model is a type of Incremental model. Software is developed in incremental, rapid cycles. The cycle usually completes in 2-3 weeks' timeframe. This result in small incremental deployments with each release built on previous functionality. Each release is thoroughly tested to ensure software quality is maintained.

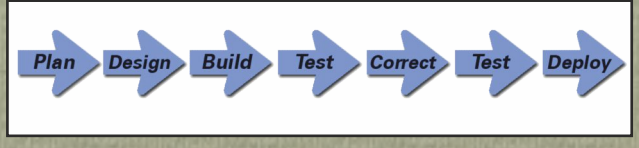

Fig 1. Waterfall Model

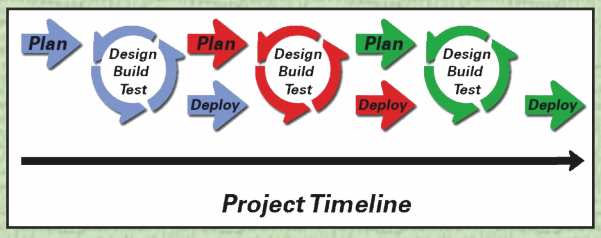

Fig 2. Agile Process Model

3. Spiral model

Spiral Model is also an incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spirals, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spiral is built on the baseline spiral.

4. Rapid Application Development model (RAD)

RAD is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

Do these fit only for large scale projects?

Not necessarily! Even a small scale projects can benefit from such process models. As usual, most small scale project will later scale up to large scale and you shouldn't worry about not following the better process models later point of time.

Do you follow any such models?

We the Software Development Team at Tessolve follow Agile model for our S/W development, where we run a sprint for every 2 weeks and deliver a well-tested feature by end of every 2 weeks. And on daily basis, we call for a stand-up to get updates from team and keep the team focus on their goals.

Sprint, stand-ups..?! What are those? Could you explain in more detail?

Sure, I will explain those, in the next newsletter.

TP Integrity for Test Program Security

Sudhakar Rajamanickam - Test Lead

When TE releases a test program, there is a possibility that accidentally changes may happen to the key setup (e.g. timing, level) during production run. These changes have a huge impact to Quality and incur cost on Retest especially on the fused parts. TP-Integrity ensures TP setup remains un-changed after Test program is released to production. TP-Integrity comprises of a tool to generate test program checksum in a database file and a test method code to be used in a testsuite, which compares the runtime Checksum file of the loaded setup to checksum in database file during release.

With the help of the tpi tagging tool, checksum for each test program file is created and stored in a database file. During testing, "TPIntegrity" testsuite compares checksum of loaded setup against the database checksum file when it was released - bin out as FAIL when there is a mismatch in comparison.

On First Run/TD since it generates and compares the checksum file against the database it will consume ~ 120ms. On Subsequent Runs/TD, it will just datalog consuming only 1.5ms.

Tool parses in testflow name as an input parameter; based on the content of program setup files listed below, generates 1 encrypted code for each file: setup files are Testflow, Pin config, Level, Timing, Vector, Test table (also manages multiple .csv files in MFH) & Analog Control. Encrypt this data and store it in the device directory with filename "<testflow.tf>".checksum file. Check for TPIntegrity testsuite existence by checking the testmethod used in testflow. Check and ensure the TP integrity testsuite is NOT bypassed or group bypassed or set PASS.

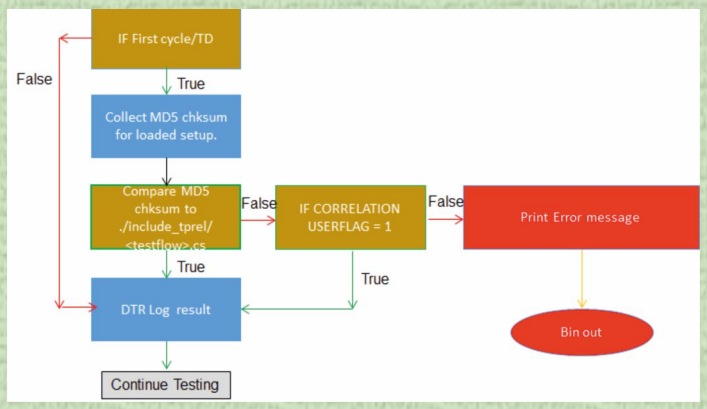

Fig 1.

Testsuite "TP-Integrity" using the specific TML is used to match encrypted data. Users are responsible to add the "Run and Branch" testsuite to the testflow. If a mismatch happens, TM will check the status of "CORRELATION" User Flag controlling the Fuse blow. If CORRELATION == 1 (non-production version where Fuses are not blown), testflow will pass & still proceed with execution. This allows in-house and remote debugging (except QFPROM debugging). If CORRELATION == 0 (production version where fuses are blown), TM will set function as fail and bin out as per test table setting with ERROR message in the UI report (for intial testing, testflow continues).

TESSOLVE ENGINEERING CHALLENGE - CASE STUDIES

Image Grab Algorithm Creation Using VHDL: An avant-garde move towards Image Capture

Arjun Gupta - Test Engineer & Sajin K - Test Lead

It has always been a challenging job to carry out image capture through high speed LVDS lines and it becomes far more complex when the same includes post processing while capturing the image.

Here in this particular project that was executed; we had a challenge to read the Image sensor memory (2112x512) bit. After reading it, convert the pixel data that is read to the corresponding resolution current (16 bit each) to which the sensor responds and finally store it as the actual Image. This task when being executed using a normal method (via a PC) would have lead to a much higher Test Time and also would have resulted to memory breakdown with each iteration due to storage of a multiple images. The normal method of image capture on the tester would have been impossible with XP based slower PC and we have to find out a solution outside the tester capability.

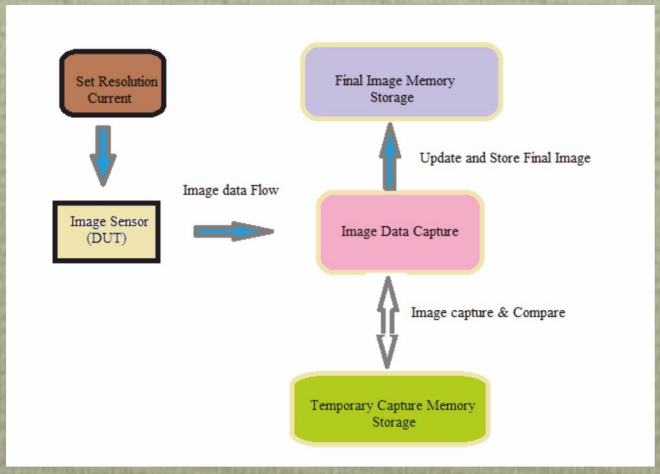

Fig 1. Image construction Algorithm Using VHDL

This required a more innovative approach which can overcome the above limitations and prove to be more efficient in terms of both Test Time and Memory utilization. So keeping that in mind, the whole process is now executed using VHDL (Cyclone III FPGA) which not only produced desired performance but further reduced the test time to a great extent.

Test Procedure

Step1: Set Image sensor with initial Resolution Current.
Step2: Initialize the Final Image memory with initial Resolution Current and trigger Image sensor to throw image data.
Step3: Transfer the captured data to Temporary capture memory.
Step 4: Set next Resolution current and capture the image data.
Step5: Read the Temporary capture memory (256 bits) and compare with the current Image data.
Step6: Compare the each capture data bit for binary '0' or '1'

- If the captured data is 0, update the Final image memory with reference resolution current value.
- If the captured data is 1, retain the same value of resolution current in Final image memory.

Step7: Repeat the above steps from 4 to 6 until all the resolution current range is covered.
Step8: Save the final Image.

Though the implementation of the same using VHDL was itself a challenge, since the data need to be continuously written and read from the memory, there were three other major challenges that we had faced. Below are the challenges,

Test Challenges

- To test the VHDL implementation in simulation mode without the Image sensor, we had to develop an internal Data Generator module and a memory that mimics the sensor memory in the real time.
- In order to synchronize the image data capture and update of resolution current for each iteration we had to develop an external block in VHDL which operates in parallel to capture block and works to update the resolution current with each capture.
- Reading the entire 2112x512 will lead to the shortage of FPGA slices. To overcome this, we have provided an option to read one memory location (256 bits) at a time.

Final image in the memory contains the original pixel data with object

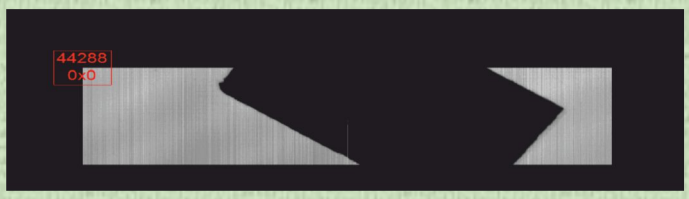

Fig 2. Final image captured from the image sensor with the image following the shape of the object

High Current Load Board Design – Consideration & Challenges with signals routing in dual stripline environmental for dual site

Selvaraj S - PCB Design Engineer

Scope of work

This case study describes the High current design consideration and challenges while designing up to 60amp current rating with signals routing in dual strip line environmental without cross talk interferences.

Challenges encountered during high current power domain designing

- While designing the core power 2 Oz copper thickness for this particular power domain is used in order to carry over up to 60amp current rating.
- 3 mil core was used for core power with 2 Oz copper thickness in both sides. (One side having power plane and another side having gnd plane)
- As the application area for placement accommodation is too less and real estate for routing also very less, this layer was poured with copper plane to entire area (14.25"X11") and splited in center of the board since this is dual site (Left side goes to site 1 and right side goes to site 2)
- Via size was calculated and multiple vias were used for Bulk caps and other de caps in order to improve the power loss Characterization.

Challenges encountered during digital signals routing in dual stripline environmental for dual site

- Totally 7 power plane need to be done for each site. These power plane layers need to be used as reference to the impedance control traces.
- All signals were routed in horizontal and vertical axis with minimum no of transition vias. During routing stage we cross verified the signal routing and maintain the parallelism also, so this was useful to reduce the maximum rework during cross talk analysis.
- One signal can be routed to tester from dut region with max 7 transition vias whereas signals are noncritical, second priority signals can have max 3 transition vias and high priority signals should be routed with max 1 via also stub drill process need to be called out in order to reduce the unwanted stub.
- The first and second priority signals are routed in horizontal and vertical patterns with max allowable cross talk (DC limits)
- Maintaining at least 5X spacing or more for all first priority signals in order to escape from aggressive nets (both neighbour and parallelism signals).
- This design was completed with 30 layers stack up formation (15 signal layers and 7 power plane layers).
- In order to remove the unwanted stub and control the actual signal reflection the High priority transmitter signals were to be routed in micro strip line environmental and receiver signals in strip line environmental
- Length match was done for all first and second priority signals in order to obtain the same propagation delay in terms of different routing environmental such as Micro strip line, Strip line, and dual strip line. These Rx signals needed to isolate at least 100 mil away from any other entity like Via, traces copper and etc. Also above signals were involved into cross talk analysis with max 10mv threshold (DC).

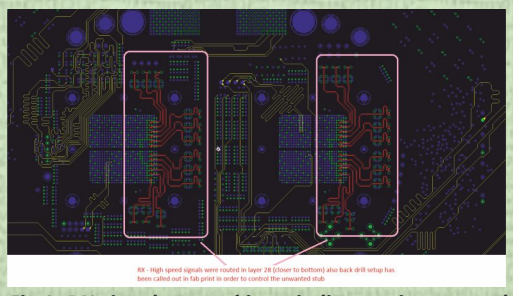

Fig 1. RX signals routed in strip line environmental

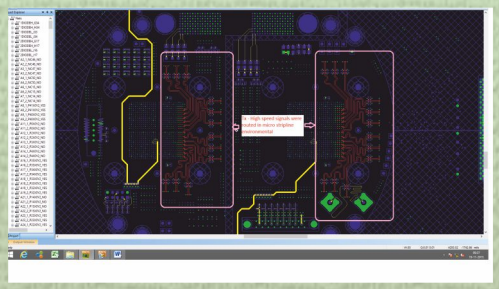

Fig 2. TX signals routed in micro strip line environmental

High current Power domain

This is a core power which works at 6.3V/60A max transient current rating. Special requirements for this power supply is to have 2 Oz copper in single power layer. In order to achieve the plane impedance for multiple GND layers with high current application We have used no of LOW ESR capacitors to have high capacitance.

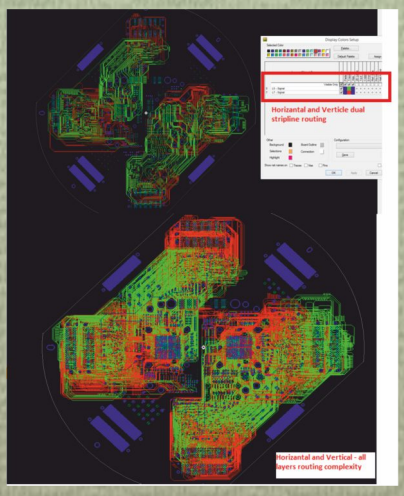

Fig 3. Routing complexity with dual strip line concepts

Tantalum Caps – Sin wave placement

We placed the tantalum capacitors in sin wave type. Integrating the tantalum capacitors vias maximize mutual inductance coupling between

alternating vias which will cancel H-field flux. As a result this will lower overall net inductance of vias.

Signal Integrity analysis

Cross talk analysis has been performed successfully to make sure whether the (high) first priority signals have escaped from aggressor nets with max 10mv (DC) cross talk threshold limit and second priority signals with 150mv (DC) cross talk threshold limit.

Conclusion

This design had a lot of challenges while placing the component, Critical signals routing, Power/GND plane, signal impedance achievements with dual strip line environmental and Cross analysis. Lesson learnt from this design, we can follow the similar concept of placement and routing with dual strip line for all upcoming design where no of layers need to be controlled during pcb designs in order to minimize the production cost.

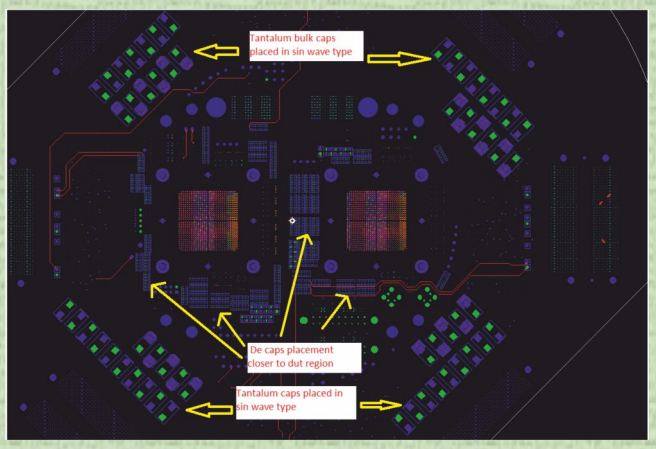

Fig 4. Sin wave type - Tantalum Bulk caps and decaps placement

VM or Vector Memory reduction challenges and approach

Abhishek Basava & Vamsi Krishna - Test Engineer

In two of our critical projects on conversion, there were multiple challenges like Vector Memory constraint, pattern conversions and debug, test time optimization up-to 50% of original platform, needless to say overall cost needs to be low too. Here we discuss how our team dealt with pattern conversions and vector memory constraints.

The legacy tester platform doesn't have Vector Memory limitations, and has all tests running at a constant frequency. Our first target was to convert the Legacy Tester vectors to Target Testers binary files and debug. Only after conversion of all patterns, the pattern depth can be calculated and the value was found to be >160M for one FPGA device and >250M for another FPGA device.

Target Tester Pin Scale architecture supports maximum 64M VM area while Smart Scale architecture supports only up-to 112M. So VM reduction was a crucial part for both our projects, without which production release was not feasible; the only means to achieve this with full test-coverage was to go with X4-mode conversions. Maximum possible X-Mode is X4. However, considering the fact that not all tests passing in X1 patterns necessarily pass in X4 patterns, and X4 doesn't efficiently reduce the VM size by 1/4th of X1, the target to drastically reach <64M was by far a challenge.

The purpose of X modes is to use memory more efficiently. After X1 patterns pass; again we need to reconvert Legacy Tester patterns to Target Tester binary with X4 mode and re-validate. It's like an additional effort to meet VM requirements to load complete program in production testing.
Negative impacts of high VM size include enormous testing cost, extra licensing for additional tester resources utilization, incapability to load complete program in production check-out due to license and hardware limitations, also greater test-time as vector memory contributes to test time as-well.

Approach and solution

- **X mode**

  This term refers to setup processes which can be used to fit multiple device cycles into a single test system cycle. The "X" implies multiplication ("X4 mode" means that there are 4 device cycles per test system cycle).

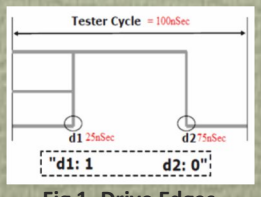

Fig 1. Drive Edges

We are using 2 drive edges to represent a clock format. Assume 100nSec is period. Rise at 25nSec & Fall at 75nSec. We have only 8 drive edges. Maximum possible X mode is "X4".

- **X2/X4 mode without padding**

  X2/X4mode will add extra padding cycles, whenever change in tests ends with odd cycle. This padding will add an impact on test time.

Example

X1 Cycle depth = (1+ No padding) + (1+ No padding) + (1+ No padding) + (8+ No padding) + (2+ No padding) = 13 Cycles
X2 Cycle depth = (1+ 1 padding) + (1+ 1 padding) + (1+ 1 padding) + (8+ No padding) + (2+ No padding) = 16 Cycles (16-13= 3 extra cycles)
X4 Cycle depth = (1+ 3 padding) + (1+ 3 padding) + (1+ 3 padding) + (8+ No padding) + (2+ 2 padding) = 24 Cycles (24-13=11 extra cycles)
In reality, if there is multiple timing set switch over, padding will be more, X4 VM gain is less, result in more test time and less X4-conversion efficiency. Although vectors will be compressed in X mode, efficiency is reduced due to padding of extra cycles.

A new STIL modifier tool was created for replacing the conventional STIL modifier which use T-Set to identify clocking. With this conversion method there is no padding vectors in the converted bin file.

Details

Regular STIL file converted from VCT (Legacy Tester patterns) is not of Industrial standard means it will have 'P' and 'C' characters other than 0,1,x which represent 0-1-0 as C and 1-0-1 as P for the clock pins as clock is of SBC format. So STIL modifier is a tool used to convert STIL file which are of non-Industrial standard to industrial standard STIL. It will replace C with 0 and P with 1 in modified STIL and adds a T-Set as a bookmark before that vector at which it encounters a P or C character and it will also create a header file with Clock Pin names.

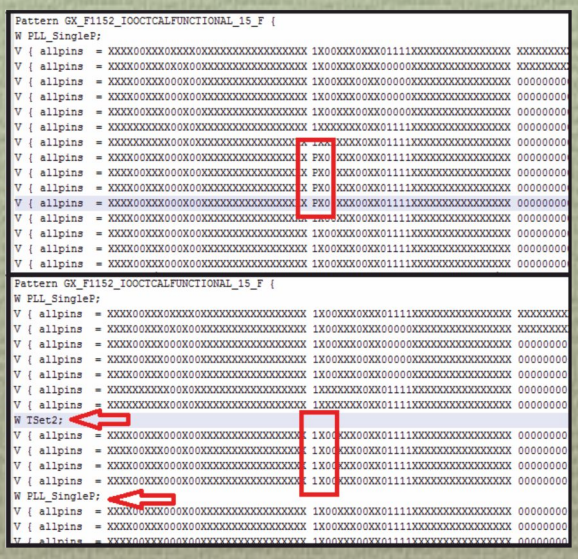

Fig 2. Original STIL file (above); Industrial standard STIL file (below)

This T-Set switchover will increase number of vectors by adding padding vectors. If it is X4 conversion the impact is high as the number of combinations for padding vectors increase from 2 to 16. This increase in vectors will increase the vector memory as well as Test time. We cannot avoid the T-Set switchover in F (functional) patterns as Clock pins change from subtest to subtest. But we observed that TP (setup) file will have same clock pins for all tests. This is the key for new method of conversion.

Clock pins are identified which will have either 1-0-1(P) format or 0-1-0(C) clock means if a clock pin is having 1-0-1 as the clock format, then it will not have any data as 0. In entire pattern corresponding pin will have either 1 as data or 1-0-1(P) as clock. The new STIL modifier tool will replace all P with 0 and all C with 1 to distinguish between clock and data unlike previous STIL modifier (0-1-0 as C and 1-0-1 as P) which use T-Set to identify clocking.

```
Pattern GX_F1152_IOOCTCALFUNCTIONAL_15_F {
W PLL_SingleP;
V { allpins  = XXXX00XXX0XXXX0XXXXXXXXXXXXXXXXX 1X00XXX0XXX01111XXXXXXXXXXXXXXXX XXXXXXXX
V { allpins  = XXXX00XXX0X0X00XXXXXXXXXXXXXXXXX 1X00XXX0XXX00000XXXXXXXXXXXXXXXX XXXXXXXX
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX00000XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX00000XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX00000XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXXXXXXXX00X0XXXXXXXXXXXXXXXXXX 1XXXXXXX0XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXXXXXXXX00X0XXXXXXXXXXXXXXXXXX 1XXXXXXX0XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 0X0XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 0X0XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 0X0XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 0X0XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X0XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX01111XXXXXXXXXXXXXXXX 00000000
V { allpins  = XXXX00XXX000X00XXXXXXXXXXXXXXXXX 1X00XXX00XX01111XXXXXXXXXXXXXXXX 00000000
```

Fig 3. Modified Industrial standard STIL file with no padding cycles and no T-Set switching

With this conversion method there is no padding vectors in the converted bin file. For example typical TP pattern having say 640560 vectors in x1.

EXAMPLE:

Vector depth in x1 is (file size =3.41MB)	=	6, 40,560 cycles
Vector depth with T-Set approach of conversion (default padding) for x4 (file size =1.98MB)	=	1, 93,160 cycles
VM gain from x1 to x4 in T-Set approach is [(6,40,560 - 1,93,160)/6,40,560]*100	=	69.84%
Vector depth with new approach of conversion (no padding) for x4 (file size =1.82MB)	=	1, 60,140 cycles
VM gain from x1 to x4 with new approach is [(6,40,560 – 1,60,140)/6,40,560]*100	=	75.00%

Overall there will be considerable gain in vector memory space using this approach and is very critical for our projects as we targeted to fit within 64M license. This approach could be extensively used in optimizing the overall test program and will help to save considerable cost in the production environment.