Test Time Optimization: A Novel Staggered-capture Architecture Using A Token-passing Architecture

Khushboo Agarwal (PMTS, AMD)
Ravindra Mathanker (MTS, AMD)
Ari Shtulman (Director, AMD)
Ahmet Tokuz (Director, AMD)
Ivan Marinov (SPW, AMD)
Kristian Yordanov (SPW, AMD)
Miroslav Marev (SPW, AMD)

AMD
together we advance_

[AMD Official Use Only - General]

Agenda

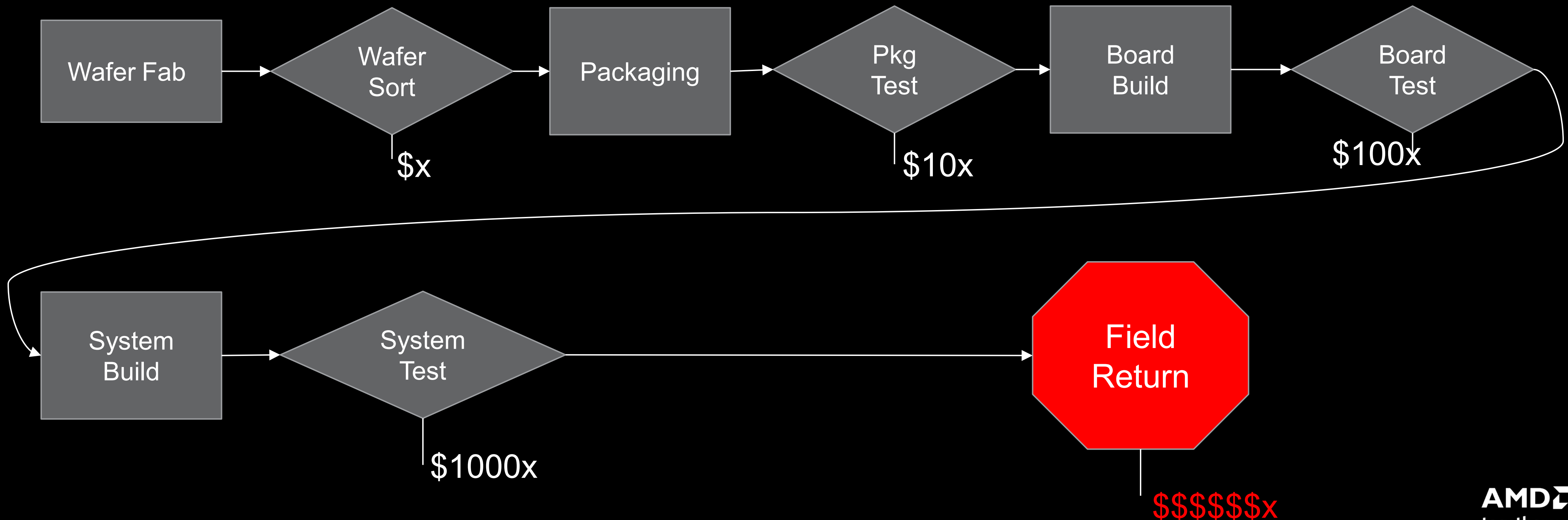
- Problem Statement
- Staggered Capture scheme - Introduction
- Introduction to OCC architecture
- Several OCCs in chain(s) – for clock staggering via token passing
- Modified OCC architecture to support clock staggering
- Additional features supported in this proposed architecture
- ATPG Results
- Conclusions
- Future Work

AMD
together we advance_

[AMD Official Use Only - General]

Problem Statement

- Large SOCs -> multiple scan test modes -> lot of patterns needed for robust testing!!
- Patterns might be pruned due to tester memory limitations – leading to low quality test -> customer returns!! ☹
- Multi-clock designs with several interacting asynchronous clock domains – pulsed in a one-hot manner!!
- Can we combine several clock domains in the same pattern?? – Timing aspect??
- Clock pulses need to be spaced far apart to avoid timing violation which can lead to silicon failures
- If this can be achieved reliably, we can significantly reduce test time and improve test quality: Lower DPPM!! ☺

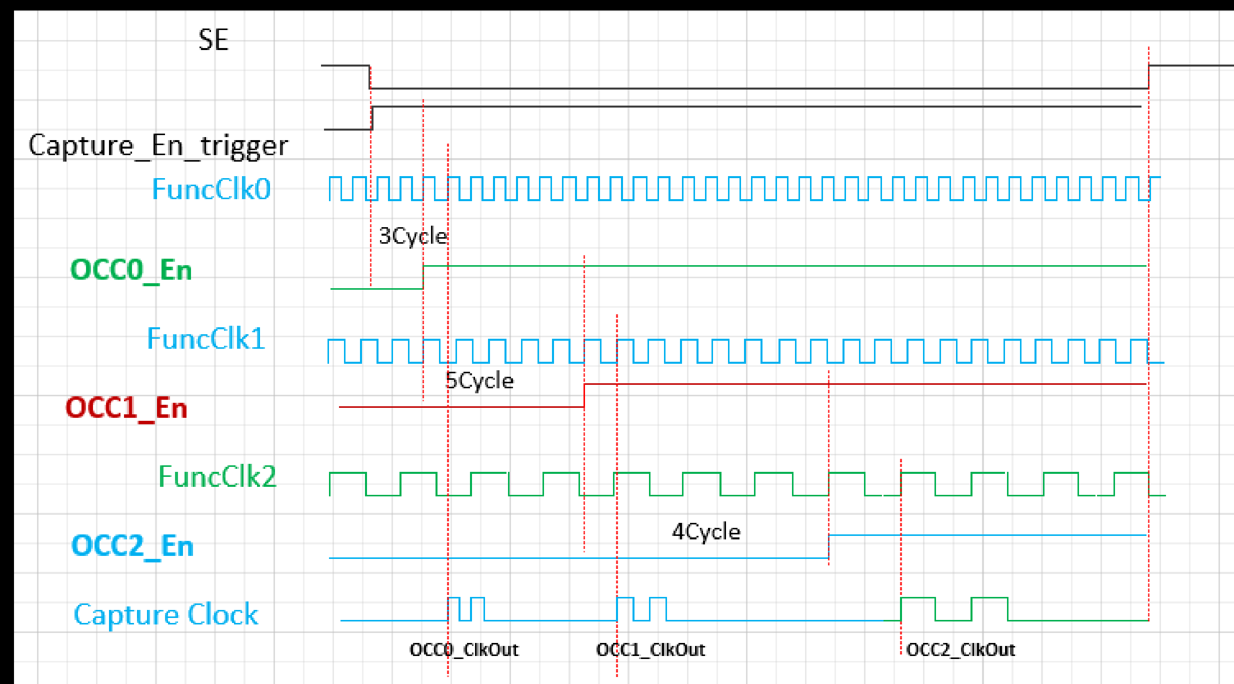


AMD
together we advance_

[AMD Official Use Only - General]

Staggered Capture scheme - Introduction

- Pulsing Capture clock (stuck-at/at-speed) for multiple **interacting** clock domains in the same pattern
- Pulses should be spaced far enough so as to not create a timing violation – paths between these domains are false/asynchronous in timing
- This scheme improves coverage per pattern – by packing more detected faults
- Overall scan pattern count should be reduced (for similar coverage)
- Less test time!!

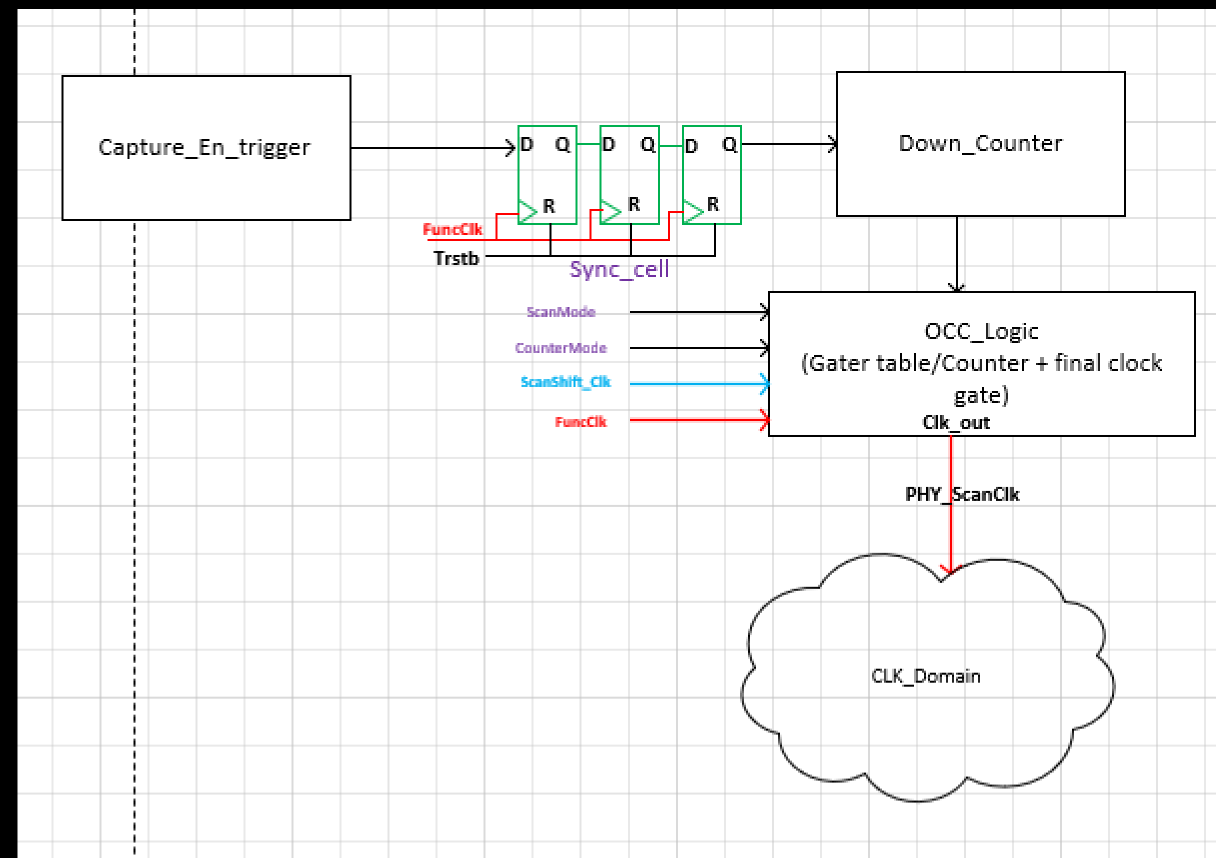


AMD
together we advance_

[AMD Official Use Only - General]

Introduction to On-Chip Clocking (OCC) architecture

- Capture_En trigger is the trigger for each OCC and comes from ATE (broadcasted to all OCCs)
- After trigger comes, it is synchronized to the functional clock domain
- Synchronized trigger enables the down counter (initial value of this counter can be set during scan)
- Once counter expires, it enables the OCC logic to start emitting the capture pulses
- Enabling all OCCs at the same time can cause pulses ending up fairly close/overlapping
- Manual calculation of counter for each clock domain is error prone and fails in case of shmooing



AMD
together we advance_

[AMD Official Use Only - General]

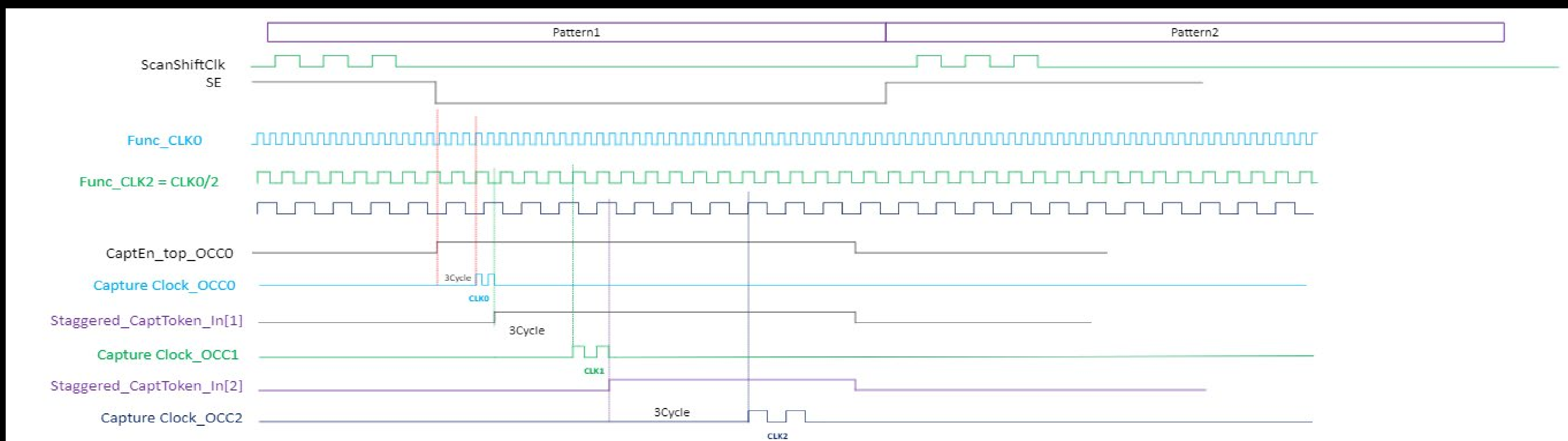
Several OCCs in chain(s) – for clock staggering via token passing

- Built-in hardware solution for clock staggering : daisy chaining of OCCs and token/trigger passing!!

OCC_0 – 625MHz
AcScanMode
AcCaptEn
ScanShiftEn
Staggered_CaptToken_In[0]
Tdr_Staggered_CaptToken[0]
1'b0
ScanShiftClk
FuncClk0
(Free Running)
CaptEn_GEN_LOGIC
Staggered_CaptToken_In
Tdr_Staggered_CaptToken
Staggered_CaptToken_Out[0]
OCC0_CapTokenOut
OCC0_ClkOut
Domain0_625

OCC_1 – 500MHz
Staggered_CaptToken_In[1]
Tdr_Staggered_CaptToken[1]
1'b1
FuncClk1
(Free Running)
Staggered_CaptToken_Out[1]
OCC1_CapTokenOut
OCC1_ClkOut
Domain1_500

OCC_2 – 400MHz
Staggered_CaptToken_In[2]
Tdr_Staggered_CaptToken[2]
1'b1
FuncClk2
(Free Running)
Staggered_CaptToken_Out[2]
OCC2_CapTokenOut
OCC2_ClkOut
Domain2_400



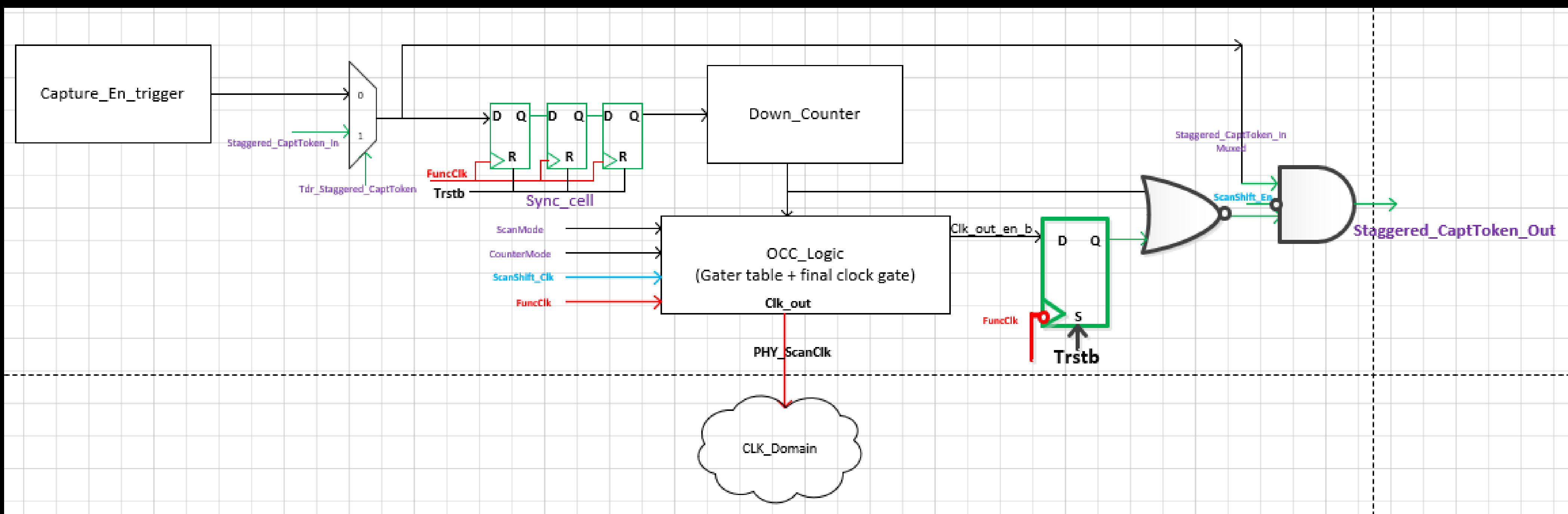
AMD
together we advance_

[AMD Official Use Only - General]

Modified OCC architecture to support clock staggering

- Mux on Capture_En_trigger – first OCC gets trigger from ATE, subsequent OCCs in the chain get a token from the previous OCC as a trigger

- The next OCC is triggered ONLY after all of the following conditions are met:
  - ScanShift_En has been de-asserted (we are in capture phase)
  - Down Counter from previous OCC has expired
  - Previous OCC finished generating clock pulses



AMD
together we advance_

[AMD Official Use Only - General]

Additional features supported in this proposed architecture

- In RTL one long chain of OCCs can be created (randomly)

- During ATPG, this chain can be broken down into several short segments using TDR programming – gives full flexibility during ATPG on how many OCCs we want to keep in a chain
  - We could even decide to NOT use staggering function at all and trigger each OCC independently

- If any given OCC (in a chain) is bypassed (not used) during pattern generation, it will immediately pass the incoming token to next OCC (does NOT waste any tester cycles for token passing)

- Counter values in each OCC can be programmed to create configurable amounts of delay before the OCC is triggered even after the arrival of token from previous OCC

- This scheme can be used for both stuck-at as well as at-speed pattern generation in a staggered fashion

[AMD Official Use Only - General]

ATPG results – At-speed scan

- Results were collected across 4 test circuits with varying numbers of clock domains and pattern distributions across those domains

Test Circuit	Longest Scan Chain	Slow Clock Frequency (Shift / Capture)	# of Clock Domains (OCCs)	Maximum Functional Clock Frequency	Total Scan Flop Count	Dominant Clock Domain as % of Total Pattern Count (Stuck-At)	Dominant Clock Domain as % of Total Pattern Count (At-Speed)
Circuit A	250	200 MHz	34	810MHz	67,007	33.95%	43.28%
Circuit B	240	200 MHz	19	1250MHz	59,919	38.06%	47.76%
Circuit C	130	200 MHz	5	570MHz	37,957	63.64%	66.67%
Circuit D	194	200 MHz	9	1000MHz	117,172	63.65%	84.59%

- OCCs were divided into equal-sized “staggered chains”
- Each test pattern had an expanded capture window to allow for staggered capture across all OCCs in a given “staggered chain”
- All 4 circuits showed improvement in pattern count and test time in varying amounts
- The biggest contributor to variance was the presence of a “dominant” clock domain

Test Circuit	# of staggered OCCs / chain	Orig. Patt. Count	Orig. Test Time	Orig. Test Cov.	New Patt. Count	New Test Time	New Test Cov.	Patt. Count Delta (%)	Test Time Delta (%)	Cov. Delta (%)
Circuit A	8, 8, 7, 7, 4	26,736	121,907,802 ns	74.79%	15,807	80,140,500 ns	75.73%	(40.88%)	(34.26%)	0.94%
Circuit B	6, 5, 4, 4	14,016	59,426,895 ns	70.65%	9,329	42,323,825 ns	70.48%	(33.44%)	(28.78%)	(0.17%)
Circuit C	5	6,902	23,682,374 ns	84.02%	4,862	17,555,674 ns	84.04%	(29.55%)	(25.85%)	0.02%
Circuit D	5, 4	20,930	90,826,440 ns	80.09%	18,208	84,474,060 ns	79.78%	(13.00%)	(6.99%)	(0.21%)

AMD
together we advance_

[AMD Official Use Only - General]

ATPG results – Stuck-at

- The same 4 test cases were used for stuck-at data collection

Test Circuit	Longest Scan Chain	Slow Clock Frequency (Shift / Capture)	# of Clock Domains (OCCs)	Maximum Functional Clock Frequency	Total Scan Flop Count	Dominant Clock Domain as % of Total Pattern Count (Stuck-At)	Dominant Clock Domain as % of Total Pattern Count (At-Speed)
Circuit A	250	200 MHz	34	810MHz	67,007	33.95%	43.28%
Circuit B	240	200 MHz	19	1250MHz	59,919	38.06%	47.76%
Circuit C	130	200 MHz	5	570MHz	37,957	63.64%	66.67%
Circuit D	194	200 MHz	9	1000MHz	117,172	63.65%	84.59%

- Circuits A and B still see a significant pattern count and test time improvement
- Circuit C has pattern count and test time improvement, but significant coverage loss.
- Circuit D has neither a test pattern or test time improvement in addition to significant coverage loss.
- Suspect that the presence of a very dominant OCC in circuits C and D complicated test cube generation resulting in lower coverage.
- Circuits C and D have much shorter scan chains than Circuits A and B, resulting in a larger penalty for increasing the capture window.

Test Circuit	# of staggered OCCs / chain	Orig. Patt. Count	Orig. Test Time	Orig. Test Cov.	New Patt. Count	New Test Time	New Test Cov.	Patt. Count Delta (%)	Test Time Delta (%)	Cov. Delta (%)
Circuit A	8, 8, 7, 7, 4	25,560	103,004,460 ns	87.22%	11,254	52,215,500 ns	87.13%	(55.97%)	(49.31%)	(0.09%)
Circuit B	6, 5, 4, 4	12,685	52,822,885 ns	88.17%	10,120	44,700,465 ns	88.15%	(20.22%)	(15.38%)	(0.02%)
Circuit C	5	6,131	20,032,952 ns	92.03%	4,902	17,737,834 ns	91.31%	(20.05%)	(11.46%)	(0.72%)
Circuit D	5, 4	8,444	34,357,320 ns	91.07%	8,729	38,400,730 ns	90.63%	3.26%	10.53%	(0.44%)

AMD
together we advance_

[AMD Official Use Only - General]

ATPG results – Stuck-at – remove dominant OCC from stagger

- The same 4 test cases were used for stuck-at data collection

Test Circuit	Longest Scan Chain	Slow Clock Frequency (Shift / Capture)	# of Clock Domains (OCCs)	Maximum Functional Clock Frequency	Total Scan Flop Count	Dominant Clock Domain as % of Total Pattern Count (Stuck-At)	Dominant Clock Domain as % of Total Pattern Count (At-Speed)
Circuit A	250	200 MHz	34	810MHz	67,007	33.95%	43.28%
Circuit B	240	200 MHz	19	1250MHz	59,919	38.06%	47.76%
Circuit C	130	200 MHz	5	570MHz	37,957	63.64%	66.67%
Circuit D	194	200 MHz	9	1000MHz	117,172	63.65%	84.59%

- Reconfigured the staggering to capture the dominant OCC on its own named-capture procedure
- Staggered the remainder of the OCCs
- The coverage loss in Circuit C is now completely recovered, but at the cost of pattern count and test time
- Circuit D sees pattern count and test-time improvement. Test coverage now in-line with baseline numbers
- Removing dominant OCC from staggering appears to have eased the strain on the test generator

Test Circuit	# of staggered OCCs / chain	Orig. Patt. Count	Orig. Test Time	Orig. Test Cov.	New Patt. Count	New Test Time	New Test Cov.	Patt. Count Delta (%)	Test Time Delta (%)	Cov. Delta (%)
Circuit C	4, 1	6,131	20,032,952 ns	92.03%	5,992	20,191,634 ns	92.12%	(2.26%)	0.78%	0.09%
Circuit D	4, 4, 1	8,444	34,357,320 ns	91.07%	7,756	33,555,620 ns	91.04%	(8.15%)	(2.33%)	(0.03%)

AMD
together we advance_

[AMD Official Use Only - General]

ATPG results – Stuck-at – remove dominant OCC shorten staggering to less OCCs per pattern

- The same 4 test cases were used for stuck-at data collection

Test Circuit	Longest Scan Chain	Slow Clock Frequency (Shift / Capture)	# of Clock Domains (OCCs)	Maximum Functional Clock Frequency	Total Scan Flop Count	Dominant Clock Domain as % of Total Pattern Count (Stuck-At)	Dominant Clock Domain as % of Total Pattern Count (At-Speed)
Circuit A	250	200 MHz	34	810MHz	67,007	33.95%	43.28%
Circuit B	240	200 MHz	19	1250MHz	59,919	38.06%	47.76%
Circuit C	130	200 MHz	5	570MHz	37,957	63.64%	66.67%
Circuit D	194	200 MHz	9	1000MHz	117,172	63.65%	84.59%

- Reconfigured the staggering to reduce the number of staggered OCCs for circuits C and D
- Dominant OCC still on its own named capture procedure
- Now, Circuit C shows improvement in pattern count and test time, the length of the capture window is now reduced sufficiently
- Circuit D, however, has an increased pattern count due to more NCPs and no longer results in a test time improvement.
- Therefore, the longer OCC staggered chains are better for Circuit D, but not Circuit C

Test Circuit	# of staggered OCCs / chain	Orig. Patt. Count	Orig. Test Time	Orig. Test Cov.	New Patt. Count	New Test Time	New Test Cov.	Patt. Count Delta (%)	Test Time Delta (%)	Cov. Delta (%)
Circuit C	2, 2, 1	6,131	20,032,952 ns	92.03%	5,714	19,124,958 ns	92.10%	(6.80%)	(4.53%)	0.08%
Circuit D	2, 2, 2, 2, 1	8,444	34,357,320 ns	91.07%	8,501	35,276,980 ns	91.05%	0.67%	2.61%	(0.02%)

AMD
together we advance_

[AMD Official Use Only - General]

Conclusion

- Staggering the capture pulses in at-speed scan shows significant benefits to pattern count and test time in all cases.
  - Additional reconfiguration of the staggering through software (TDR programming) could show additional benefits in some cases, but that was not attempted for this paper.
- Similar improvements in stuck-at are more dependent on the circuit under test.
  - Slower capture clock means that there is a larger penalty for increasing cycles in the scan capture window.
  - The length of the longest scan chain influences the impact of the increased capture window size.
  - The presence of a dominant clock domain will impact the pattern count reduction with this scheme.
- The software configurability of the token passing approach is a key benefit.
  - Reconfiguration of staggered capture through TDRs means users can get better results without design changes.
  - Through simple TDR programming, we were able to find a pattern count and test time improvement in every case.

AMD
together we advance_

[AMD Official Use Only - General]

Future Work

- Assess the impact of the order in which OCCs are staggered
  - In this experiment, we ordered the staggering arbitrarily, however the order in which a clock domain captures relative to another domain in the same pattern could impact the results.
- Leveraging AI
  - Use AI to help determine the optimal OCC staggering order and configuration (OCC chain length) for a given circuit.
- The impact on IR drop
  - The relative timing of the scan capture pulses within a pattern could impact IR drop
  - Would like to measure the IR drop relative to one-hot capture and analyze the impact

AMD
together we advance_

Copyright and disclaimer

- ©2022 Advanced Micro Devices, Inc. All rights reserved.
- AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.
- The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.
- THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION

AMD
together we advance_

AMD