Testing of a SoC with a Pre-tested and Pre-verified Silicon

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 2, Issue No. 2, 80–94

https://journals.mirlabs.in/index.php/ijcsnt
DOI-10.18486/ijcsnt/2.2.024
IJCSNT

Prokash Ghosh[1], Sandip Ghosh[2]

Abstract

Due to increase in SoC complexity, the cost of production testing on automated testing equipment (ATE) is increasing, which directly impact the gross margin and viability of any SoC. These pushes SoC maker to look for an alternative method of testing. This paper proposes an alternate scheme for production testing whereby major part (I/O AC specification testing, functional testing, loopback of Serializer/De-serializer (SERDES) testing, frequency testing, Built-In-Self-Test (BIST) etc) of the SoC testing except wafer testing (stuck-at fault, AC scan testing, DC) can be done with a pre-tested and pre-verified custom made SoC. It will help, to minimize the dependencies of Automatic Test Equipment (ATE) for SoC testing on production significantly. Hence production testing cost per device will be reduced. Also this scheme will enable to test many of the functional features of SoC on silicon, which were not possible to test using ATE due to high test cost. In this paper the micro architectural details of the scheme, system requirements, design challenges and case study conducted on different SoCs are presented.

Keywords

ATE, SoC Production Testing, Delay Measurement, I/O AC Timing Testing, Testing of I/O Interfaces, SoC Testing, Loopback Test, Serdes Testing, RF Testing.

Received: 04 April 2013; **Revised:** 12 July 2013; **Accepted:** 18 August 2013;
Published: 31 August 2013;

[1,2]Freescale Semiconductor Inc, Plot-18, Sector-16A, Noida, India.

Corresponding author:
Email: prokash.ghosh@freescale.com

Introduction

Today's SoC is integrated with many functions which in-turn increase the silicon die size and correspondingly increases the complexities of testing on ATE[8]. As we engage ATE to test many different features of a SoC, eventually we compromise on gross margin of any product. International Technology Roadmap for Semiconductor (ITRS), Test and Test Equipment[1] shows different drivers of Test cost. And which is going to increase with shrinking technologies to come. ATE capital expenditure is one of the significant contributors to the overall test cost. Several approaches[2–6,16,19–21,25,27] are available, which talks about different strategies to reduce overall test cost of SoC. But none of the approach talks about testing a SoC without ATE.

Different type of testing is performed on ATE on production, which include DFT testing, input/output (I/O) timing specification (DC and AC) testing, critical functionality testing, loopback testing for SERDES/RF circuits, BIST, power binning, frequency binning etc for a typical SoC. In recent times, due to high integration of SerDes/RF circuits on device, the loopback testing[18,22] is adopted as suitable alternative method on production testing to cover digital and most of the analog circuitry. Paper[17] describes one of the most efficient ways of testing IOs without relying on high cost ATE by providing internal loopback configurations. ATE has many features (for e.g. shmooing etc) which are needed mostly during the development phase of SoC testing, where comprehensive testing is done to check the health of silicon across process, voltage and temperature (PVT). Once SoC passes through qualification process, it goes to production for mass manufacturing. During mass manufacturing SoC is mainly tested for DFT, I/O timing, critical functionality for high speed interfaces, loopback testing, CBIST[23] etc at their specified limit, mentioned in the hardware specification document of SoC. Therefore, for GO-NO-GO program or to decide pass or fail of a chip we don't need many of the ATE's expensive features. Therefore we can think of a testing mechanism, whereby only a subset of ATE subsystem, is required to test a SoC. In this paper, we propose an alternate mechanism whereby silicon testing can be done with a pre-tested and pre-verified SoC on production. This pre-tested and pre-verified SoC can be called as ATE Replacement Chipset (ARC). The ARC can be thought of as very simplified architecture of an ATE, similar to one shown in figure 2 of[7].

Here we are mainly targeting testing AC specification of different I/O pad, functional features testing, loopback testing, CBIST, frequency and power binning. The ARC consists of different IPs like processor Core, Ethernet Controller, DDR memory responder, eSDHC controller etc. Along with different IPs, it has calibration logic, delay unit for compensating Process-Voltage-Temperature (PVT) variation. There will be a load board designed with design under test (DUT) and ARC along with traditional ATE components[8] like handler etc. There will be test software running on a computer connected to load board and handler to control the testing flow. I/O AC timing specification can be measured by exchanging predefined data with proper I/O timing using delay unit between DUT and ARC. The respective data is captured and compared with golden data to make sure that corresponding test passes or fails. Similarly loopback, CBIST, functional testing can be done using this scheme.

The paper content is organized as follows. Section II discusses top level block diagram of SoC testing with ARC. Section III talks about detail I/O AC timing measurement scheme for Ethernet interface and DDR Interface. Section IV discusses SERDES loopback testing using ARC. Section V discusses different structural testing like LBIST/FBIST/CBIST by using ARC. Section VI talks about the different implementation challenges. Section VII provides several case studies on test time saving and scope of

additional features testing with this new architecture on various SoCs. Finally, section VIII concludes the paper.

SoC Testing Sub-system with ARC

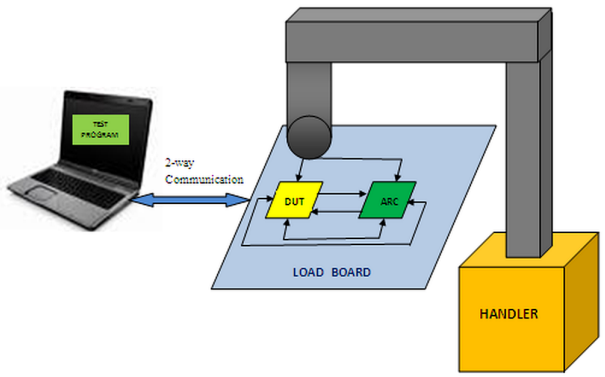

Figure 1. Top Level Block Diagram of SoC Testing Using ARC

Figure 1 shows how a SoC can be tested using an alternate approach. The basic infrastructure of testing a SoC without ATE consists of a load board containing mainly DUT and ARC, 2-way communication channel, a testing software and a handler. The testing software contains the following 1. Test stimulus for the test to be performed 2. Golden data to be used for determining pass or fail of any test. 3. Calibration program of ARC for determining delay across process-voltage-temperature (PVT) corners. 4. Configuration of load board (if any). 5. Handler program. To measure an I/O AC timing parameter of any interface, DUT & ARC will be configured by test stimulus of testing software. The test stimulus contains configuration of different peripherals, and data pattern to be used for the measurement of I/O AC timing, and output data comparison wrt golden data for the respective peripherals. After DUT and ARC configuration, DUT will perform a read transaction from ARC to measure input AC timing. The incoming data from ARC will be given programmable delay such that it will reach to DUT with intended setup and hold timing parameter to be measured. DUT captures the incoming signals (data and control) with applied timing by ARC. The received data will be stored in an internal memory of DUT. The testing software reads the internally stored data and compares with the golden data. If it matches then corresponding test will be considered as passed for the applied input timing. Similarly output AC timing can be measured by sending write transaction to ARC by DUT and ARC captures the incoming data after delaying data wrt clock such that intended timing relation is maintained. The captured data are stored in internal memory of ARC and later read out and compared by testing software to determine pass or fail of corresponding test. Before performing any test, ARC's delay line logic will be calibrated to find out the per unit buffer delay in a particular PVT condition. This delay line logic is used for applying a configurable delay between data/control and clock, so that I/O setup and hold timing can be measured across PVT. Testing software controls the handler movement for replacing the DUT from the socket of load board.

I/O AC Timing Measurement using ARC

Ethernet Interafce

Figure 2 shows the data flow that occurs between ARC and DUT for the measurement of I/O AC timing specification of an Ethernet[9] interface. It can be in any mode like reduced gigabit media independent interface(RGMII), reduced ten bit interface(RTBI), gigabit media independent interface(GMII), media independent interface(MII) etc. The magenta color line shows typical data flow for the measurement of output AC timing for the transmit interface signals. The test stimulus of testing software triggers processor core of DUT to transmit frames from Ethernet interface of DUT to ARC. The incoming signals after reaching to ARC passes through the Ethernet measuring unit (EMU), which delays the incoming signals with intended timing. The delayed signals reach to the Ethernet controller of ARC. Now Ethernet controller of ARC receives these data/control signals and finally stores inside the internal memory like L2 cache etc after processing. The stored data are read out by the testing software via debug port (e.g. JTAG[24]) connected in the 1st universal serial bus (USB) port of laptop via power tap. The power tap is an external bus translator which translates jtag transactions to USB in both directions. Now testing software compares the data with golden data to determine pass or fail condition.

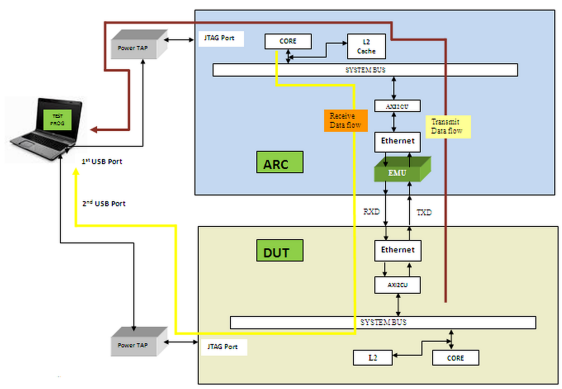

Figure 2. Transmit & receive data flow between DUT & ARC for AC spec measurement.

The yellow color line shows typical data flow for the measurement of input AC timing of the receive interface signals. The test stimulus of testing software triggers processor core of ARC to transmit frames from Ethernet controller (of ARC). The transmitted frames pass through the EMU which are delayed with intended timing and it finally reaches to DUT. The Ethernet controller of DUT receives the frames and processes it further. Finally the data are stored into the internal memory like L2 Cache etc of the DUT. Now these data are read out by the testing software via JTAG port and compared with golden data to determine pass or fail condition.

1) RGMII (1Gigabit) AC timing parameter specification

Figure 3 depicts the I/O AC timing diagram of the RGMII mode of Ethernet. Here GTX_CLK is same as TXC (transmit clock) of RGMII specification[9].

2) Transmit section of EMU in RGMII Gigabit mode

Table 1. RGMII AC Timing Specifications

Parameter	Symbol	Min	Typ	Max	Unit
Data to clock output skew (at transmitter)	t_{SKRGT_TX}	-500	0	500	ps
Data to clock input skew (at receiver)	t_{SKRGT_RX}	1	-	2.6	ns

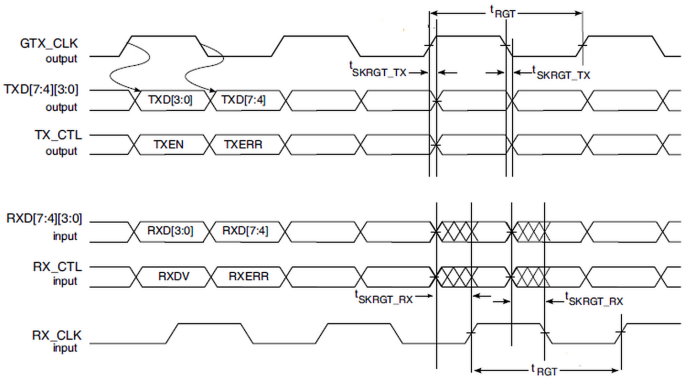

Figure 3. RGMII AC timing and Multiplexing Diagrams[9,10]

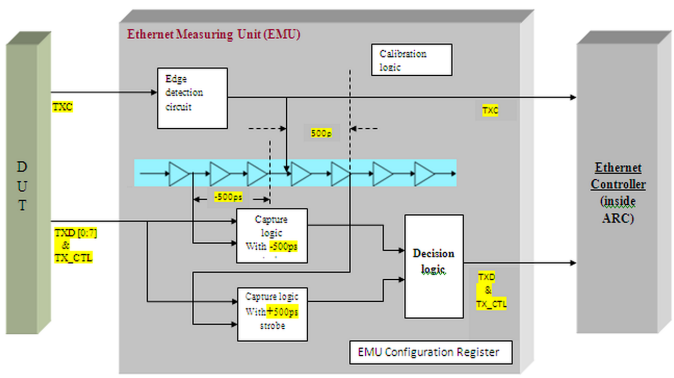

Figure 4. Block diagram of transmit path of EMU

Different phases of transmit I/O AC timing measurement by EMU of ARC in Ethernet -RGMII mode are given below:

a) Reset Phase

In this phase mainly, Phase Lock Loop (PLL) configuration of EMU, delay line calibration and EMU registers programming are done.

b) I/O AC timing measurement steps of Tx path of DUT

i) Firstly edge of output GTX_CLK (125MHz) is detected by edge detection circuit of EMU.

ii) As the gtx_clk is periodic in nature, therefore using a delay chain two strobes at 3.5ns & 4.5ns will be generated w.r.t previous edge to capture the incoming transmit data (TXD) to ARC. The strobes are gtx_clk_3p5 & gtx_clk_4p5 as shown in figure 5.

iii) The TXD either will be delayed or advanced at a particular PVT corner, therefore to sample the data, we will have to do the following –

1. Data pattern sent on transmit line are captured with strobe at gtx_clk_3p5 (inside EMU).
2. Capture data are in-turn passed through Ethernet Controller. The Ethernet controller decodes the incoming frames and stores the frame content in L2 Cache. Once frame is stored, the testing software reads the stored data via USB port which is connected with JTAG port via power tap.
3. If capture data matches with the expected data then this passes for -500ps skew (between gtx_clk and TXD) and skip step 4.
4. If the captured data doesn't match with expected data using gtx_clk_3p5 then same data pattern has to be resent for capturing data with strobe gtx_clk_4p5 i.e. to check if data arise after +500ps w.r.t clock.
5. Test will be considered to be passed if either of steps 3 & 4 is passed. If none of steps 3 & 4 are passed then, it is considered to be failed.
6. As the TXD data on the negative edge arrives from a different flop, the skew relation with gtx_clk will be different wrt positive edge of gtx_clk. Therefore we will have to run the pattern four times to make it pass. If anyone of the four skew combination mentioned in Table 2 is passed the test will be considered as passed. In table 2, "ON" denotes test need to be run with corresponding strobes selection and "NA" means corresponding strobe doesn't participate in AC timing measurement.

Table 2. Different combination of skew to be tested

Sl no	gtx_clk_pos_3p5	gtx_clk_pos_4p5	gtx_clk_neg_3p5	gtx_clk_neg_4p5	State Combination
Test1	ON	NA	ON	NA	00
Test2	NA	ON	NA	ON	11
Test3	ON	NA	NA	ON	01
Test4	NA	ON	ON	NA	10

The Decision Logic decides which strobes to be used for capturing incoming data (TXD, TX_CTL), which in-turn goes to Ethernet controller of ARC.

Figure 6 shows flow chart for output AC timing measurement of Ethernet interface. First step is to calibrate the delay chain in a particular PVT corner where output AC timing specification measurement is to be performed. The next step is to detect the clock (gtx_clk) inside ARC and generate four strobes as shown in Fig 5. Now DUT sends write transaction (i.e. transmit Ethernet frames) to ARC. The ARC captures the incoming frame with selected strobes and passes to Ethernet controller inside the ARC. Finally the frame content stored to internal memory (e.g. L2 Cache) of ARC. The stored data are read

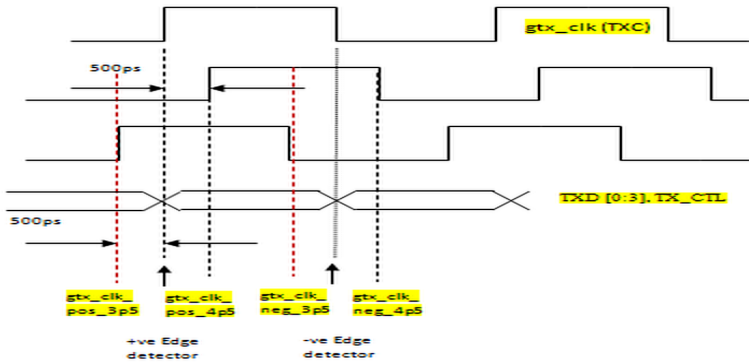

Figure 5. Timing diagram for AC spec measurement with different strobe

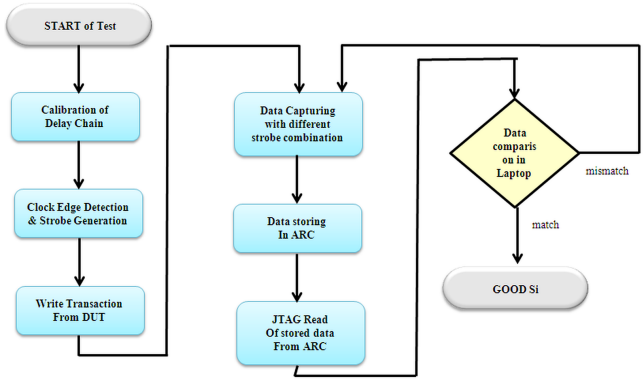

Figure 6. Flowchart for output AC timing measurement of Ethernet

out by laptop and compared with the golden data. If test passes atleast with two strobes as given in table 2, then the corresponding test can be declared as passed, otherwise it will be considered as failed.

3) Receive section of EMU in RGMII (1Gigabit) mode

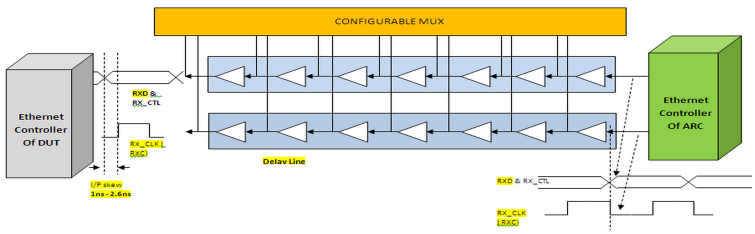

Figure 7. Block diagram of input AC spec testing of Ethernet RGMII

Figure 7 shows data from the Ethernet controller of ARC arrive at the delay line of EMU with edge align wrt RX_CLK (RXC). As the input AC timing specification between RXD/RX_CTL and RX_CLK is 1ns – 2.6ns as given in table 1 (i.e. minimum skew is 1.0ns and maximum skew is 2.6ns), therefore we have to test for both the minimum & maximum skew limit. Here configurable delay line is being used to provide the adequate delay i.e. 1.0ns and 2.6ns between RXD/RX_CTL & RX_CLK in two separate run of same test.

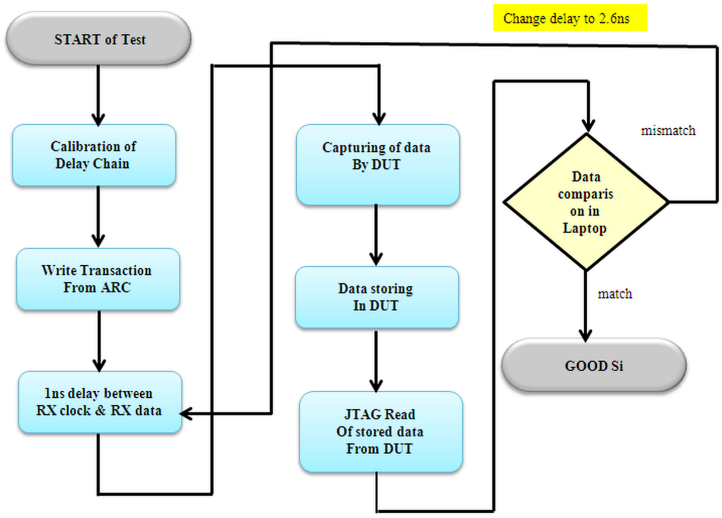

Figure 8. Flow Chart of input AC timing measurement of Ethernet

Figure 8 shows flow chart for the measurement of input AC timing specification. Again the first step is to calibrate the delay chain in the particular PVT corner where measurement/testing to be performed. ARC sends write transaction (i.e. transmit Ethernet frames) to DUT. The next step is to provide the 1ns delay between RXD/RX_CTL and RX_CLK. These delayed signals are captured by Ethernet controller of DUT. The data are further processed and finally the frame content is stored to internal memory (e.g. L2 Cache) of DUT. The stored data are read out via JTAG port of DUT and compared with golden data. If received data matches with golden data then the corresponding test is considered to be passed for that given input AC timing. If test is failed for 1ns input AC timing then same test to be re-run with 2.6ns delay. If it passes with 2.6ns delay then corresponding test can be considered as passed else test is failed.

DDR Interface

Figure 9 shows DDR measuring unit (DDRMU) with different functional blocks required for the measurement of I/O timing AC specification parameters for DDR interface [12]. There is calibration logic which determines per unit buffer delay in a particular PVT corner for the delay chain. There is a delay control unit which sets the delay in delay chain required for different interface signals of DDR like DQS, DQ, and ADDR/CMD etc. Each interface signals passes through the delay chain whose delay is controlled by the delay control unit.

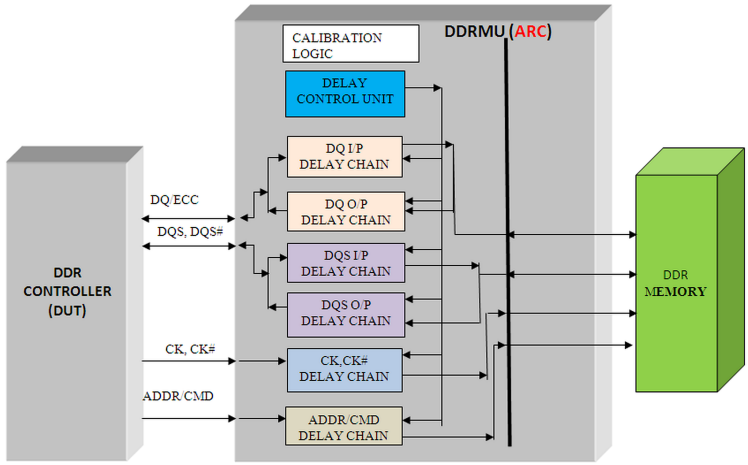

Figure 9. Block diagram of DDR measuring Unit depicting different functional blocks

1) Input AC timing measurement

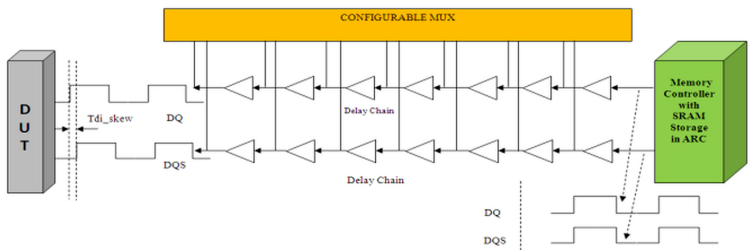

Figure 10. Logic involved in I/P AC timing measurement for DDR

Figure 10 shows the logic involves for the measurement of input AC timing of DDR interface. The mechanism is similar to input AC timing measurement of Ethernet interface.

2) Output AC timing measurement

Steps for output AC timing (setup/hold) measurement:

1. The incoming MDQS signal is being delayed by KHDS/KLDS delay chain as shown in figure 12 to generate pulse for KHDS & KLDS measurement.
2. The incoming signal is delayed by KHDX/KLDX delay chain as shown in figure 12 to generate pulse for KHDX & KLDX measurement.
3. The complete setup has to be run twice to make all the setup and hold measurement.
4. The control logic shown in figure 12 controls which test to be run.

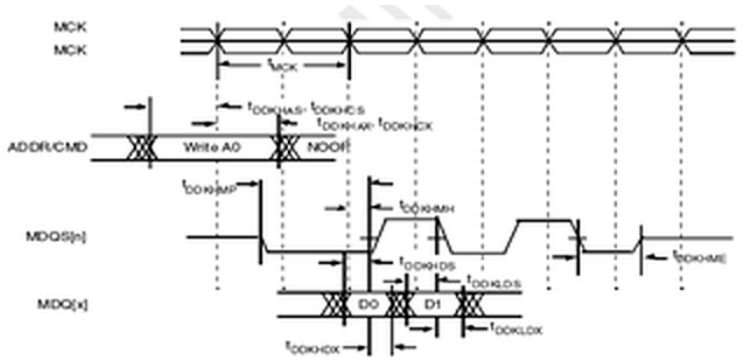

Figure 11. Timing diagram of output AC Timing[12] of DDR

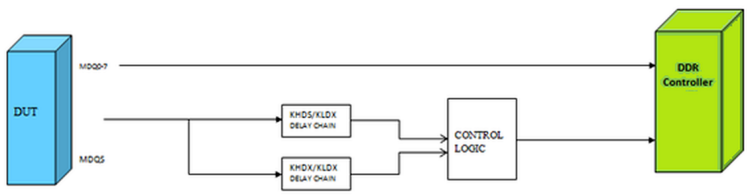

Figure 12. Logics involved for output spec measurement of DDR

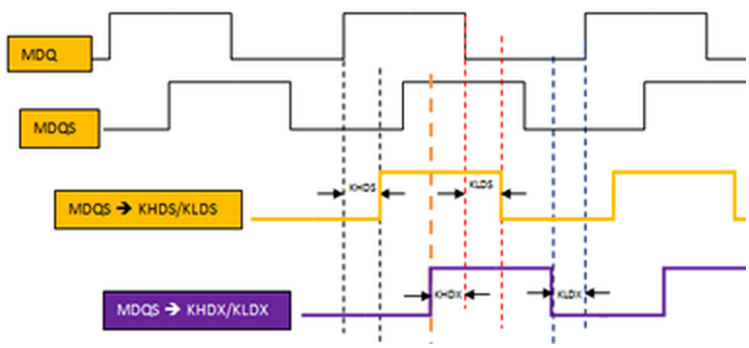

Figure 13. Timing Diagram of setup (KHDS & KLDS) & hold (KHDX & KLDX) measurement

5. The captured data by ARC is stored in internal memory.

6. Using JTAG data can be read out for comparison, which determines PASS or FAIL of AC timing.

In similar fashion AC timing of address & command signals can be measured using appropriate delay between clock and address/command signals in ARC logic.

In a similar way, micro architecture of measurement unit of other peripherals (e.g. USB, SPI, DIU etc) can be done and integrated into a single SoC to build the ARC.

SerDes and RF Interface Testing using ARC

In[17], the most efficient ways of testing most of the functionality of IOs without relying on the tester is discussed and it is done by providing internal loopback configurations. This AC IO loopback are broadly comprised of 1) setup the test. 2) Execute the test algorithm. 3) Checking the results. The above flow can be easily executed by programming appropriate configuration through debug port like JTAG or test port. Similarly typical loopback testing path of serdes is shown in figure 2 of[23]. Here the loopback testing[18,19,22] of Serdes/RF is consist of broadly above steps, which can be executed by JTAG/debug port. Once the DUT is placed on the ARC setup, the test loading, execution and results checking can be done by ARC using its JTAG port.

Circular/Logic/Functional BIST using ARC

BIST is well known and one of the most adopted strategy for structural test for different components of design[26] including memory, complex logic, microprocessor, serdes, etc in place of pure functional testing. In[16], the authors have shown how FBIST can be used for testing microprocessor functional part with high coverage. In[23], authors have shown how circular BIST logic can be used to exhaustively test the digital part of high speed serdes. In general most of the BIST execution involves configuring the BIST controller, wait for the execution of the selected algorithms and finally checking the results with expected values. These all can be done via JTAG port or any debug port. Once associated BIST controller gets implemented in design, the programming & result checking can be done by ARC as it involves sequence of JTAG transactions.

The Different Implementation Challenges

In the previous sections we have discussed about the micro architecture of measuring units that can be coupled with individual controller while designing ARC. There are various challenges that need to be taken care of during the implementation of ARC.

Calibration Logic

The calibration logic should be designed in such a way that it can calculate per unit buffer delay in a particular PVT corner with an accuracy such that different I/O AC timing measurement is possible.

Delay Chain Length

Delay chain consists of series of buffers and it should have sufficient number of buffers such that it can be used to measure different I/O AC timing parameters of future SoCs. With shrinking technology node and with increase in speed of different IPs, I/O AC timing is getting reduced day by day. Therefore, designing a delay chain, keeping in mind, future technology node and future version of IP will be a challenge.

Precise Edge Detection Circuit

The edge detection circuit is needed to detect the edge of clock or data depending on whether clock is input or output. And edge detection circuit should have high precision to provide more accuracy while performing at-speed AC timing specification testing.

Case Study

ATE Test Time saving of SoC P1025

Table 3 shows different interfaces of a product[10] and their corresponding ATE test time when the product is tested on production line with Verigy 93k tester[12]. We have not shown other test like the BIST/parametric in this case. Here we have shown interfaces which are eating up test time significantly e.g. ETSEC & QE interface consumes 5500 & 9500 millisecond (msec) respectively. The ATE test time & ATE tester resources are very expensive and every product is designed with a budgeted test time to have a targeted gross margin and that limits the scope of testing on ATE. The product P1025[10] has different functional interfaces/protocols like eLBC, DDR, 60X (processor core test mode), ETSEC (Ethernet Controller), QE (Quicc Engine), JTAG, LYNX, Platform AC (i.e. SDHC, USB, SPI, I2c, UART, etc) and SCAN[15].

Table 3. ATE test time break up of SoC P1025[10]

Interface	Test Time (msec)
eLBC	427
DDR	7823
60X (Core Exposed mode)	2100
Platform AC (Low Speed interfaces)	600
ETSEC	5500
QE	9500
SCAN	1434
JTAG	320
LYNX (SERDES)	3200
Total	30904

Table 3 shows that functional testing (at-speed + I/O AC timing) consumes 85% of ATE test time. Therefore we can conclude that I/O AC timing and feature testing consume most of the ATE test time[11]. If the proposed ARC based strategy is followed, the test time incurred due to I/O AC specification testing, functional testing, power binning, frequency binning etc can be reduced and significant cost can be saved per device[14].

ATE Test Time saving of SoC MPC8548

Table 4 shows test time break up for another SoC MPC8548[13]. Here also I/O AC timing and other functional testing, consumes more than 80% of ATE test time, which can be reduced if we adopt ARC based testing.

Table 4. ATE test time break up of SoC MPC8548[13]

Interface	Test Time (msec)
JTAG	755
SCAN	2569
Platform AC/Speed	4197
CORE Functional/Speed	4602
DDR	7614
HSSI Function	4086
LBIU	5545
LYNX (SERDES)	5927
ETSEC	3633
Parametric/Overhead	3033
Memory	2289
Total	44250

Production Cost Saving Analysis

In modern days production testing cost contributes almost 50% of the product cost. On mass production with this scheme major portion of the device production testing cost ac be saved, it will improve gross margin of the device. In this approach test time is not a cause of concern. Therefore it gives enormous flexibility to SoC makers to test many critical features to increase the quality of silicon, hence reduces customer return[24].

Conclusion

In this paper an alternate methodology of testing a chip with pre-tested and pre-verified SoC has been proposed. The basis of this new idea is to reduce ATE test time incurred due to I/O AC timing measurement, functional features testing, BIST, loopback testing, etc in production. This methodology will help to improve the gross margin of any SoC by reducing the ATE test time significantly. In our knowledge, this is the first of its kind of proposal of testing an IC with pre-tested and pre-verified SoC[27].

References

[1] THE INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS: 2011 EDITION, TEST and TEST EQUIPMENT, http://www.itrs.net

[2] Poehl, F. Demmerle, F. ; Alt, J. ; Obermeir, H. “Production test challenges for highly integrated mobile phone SOCs-A case study”, Test Symposium 2010, ETS, pp. 17–22, Date-24-28 May 2010.

[3] Iijima, K. , Akar, A. ; McDonald, C. ; Burek, D. , “Embedded test solution as a breakthrough in reducing cost of test for system on chips”, Test Symposium. Page(s): 311 – 316 , Date: 18-20 Nov, 2002

[4] Deshayes H. , “Cost of test reduction”, Test Conference, 1998. Proceedings., International IEEE conference, Conference publications, Page(s):265 – 270 , Date of Conf: 18-23 Oct 1998.

[5] Maxwell P., "Principles and results of some test cost reduction methods for ASICs", Test Conference, 2007. ITC 2007. IEEE International, Page(s): 1 – 5, Date of Conference: 21-26 Oct. 2007.

[6] Sehgal, A., Iyengar, V. ; Krasniewski, M.D. ; Chakrabarty, K. "Test cost reduction for SOCs using TAMs and Lagrange multipliers", DAC-2003. Proceedings, Page(s): 738 – 743 , Date: 2-6 June 2003.

[7] Hamidreza Hashempour, Fabrizio Lombardi, Warren Necoechea, Rakesh Mehta, and Tim Alton "An Integrated Environment for Design Verification of ATE Systems" IEEE Tran. On Instrumentation and Measurement, VOL. 56, NO.5, OCT07, PP(s):1734–1743.

[8] B. Neblett. "Implementing reusable, instrument independent test programs in the factory". In: AUTOTESTCON '96, Test Technology and Commercialization. 1996, pp. 206 –212.

[9] Ethernet Standard availbale at http://www.ieee802.org/3/ & http://www.hp.com/rnd/pdfs/RGMIIv1_3.pdf

[10] P1025 product description, documentation & hardware specification http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P1025&fsrch=1&sr=1

[11] Verigy 93k : http://www1.verigy.com/ate/products/V93000/index.htm

[12] DDR2, DDR3, DDR4 SDRAM standdard available at http://www.jedec.org

[13] MPC8548E product description, documentation & hardware spec http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MPC8548E&fsrch=1&sr=19

[14] Truebenbach, E. "Instruments for automatic tests", Instrumentation & Measurement Magazine, IEEE Journals, Vol 8, pp. 27–34, March-05.

[15] Ochoa, J.A.; Porter, J.R, "Semiconductor test strategies" Instrumentation & Measurement Magazine, IEEE Journals and Magazines, Vol 6, pp. 20–25, Date : March 2003.

[16] Praveen Parvathala, Kailas Maneparambil, William Lindsay , "FRITS - A Microprocessor Functional BIST Method", International Test Conference(ITC), 2002, Page(s):590–598.

[17] M. Tripp, T. M. Mak and A. Meixner, "Elimination of Traditional Functional testing of Interface Timings at Intel", Proceedings IEEE International Test Conference, 2003.

[18] Iain Robertson, Graham Hetherington and Tom Leslie, Ishwar Parulkar and Ronald Lesnikoski, "Testing High-Speed, Large Scale Implementation of SerDes I/Os on Chips Used in Throughput Computing Systems", IEEE International Test Conference, 2005.

[19] William A Fritzsche, Asim E. Haque "Low cost testing of multi-GBit device pins with ATE assisted loopback instrument", IEEE International Test Conference, 2008.

[20] Onabajo, M. “Strategic Test Cost Reduction with On-Chip Measurement Circuitry for RF Transceiver Front-Ends - An Overview”, Circuits and Systems, 2006. MWSCAS ’06. 49th IEEE International Midwest Symposium (Volume:2), Page(s):643 – 647

[21] Hsiu-Ming, Chang, Kwang-Ting (Tim) Cheng, Wangyang Zhang, Xin Li, Kenneth M. Butler “Test Cost Reduction Through Performance Prediction Using Virtual Probe” Proc. of IEEE International Test Conference, 2011.

[22] A Halder, S. Bhattacharya, G. Srinivasan, A. Chatterjee, “A System-level Alternate Test Approach for Specification Test of RF Transceivers in Loopback Mode”, Proc. of 18th IEEE VLSI Design Conf., 2005.

[23] G. Hetherington and R. Simpson, “Circular BIST testing the digital logic within a high speed SerDes”, Proceedings of IEEE International Test Conference, 2003.

[24] IEEE. JTAG. URL: http://standards.ieee.org/findstds/standard/1149.1-2001.htm

[25] William A Fritzsche, Asim E. Haque “Low cost testing of multi-GBit device pins with ATE assisted loopback instrument”, Proc. of IEEE International Test Conference, 2008.

[26] Mike Mayberry, John Johnson, Navid Shahriari, Mike Tnpp, “Realizing the Benefits of Structural Test for Intel Microprocessors”, Proc. of IEEE International Test Conference, 2002.

[27] Nahar, A. Butler, K.M. ; Carulli, J.M. ; Weinberger, C. “Quality improvement and cost reduction using statistical outlier methods”, Proc. of ICCD, 2009, PP:64–69