TAN: A Packet Switched Network for VLSI Testing

S. Vengatachalam, M. Nourani and M. Akhbarizadeh
Center for Integrated Circuits & Systems
The University of Texas at Dallas, Richardson, TX 75083

***Abstract*—We introduce the idea of using Packet Switched Network as the mode of communication between Automatic Test Equipment and the VLSI Chip under test in a Multi-site ATE architecture. We show that our architecture which we refer to as Test Area Network reduces the complexity and time involved in testing tens of chips at a time. To increase the ATE utilization, we distribute a portion of ATE's task of signature verification to the intelligent test-heads which are now capable of applying patterns and verifying signatures produced by the chip being tested. Our analysis and empirical results indicate a speedup of 4 to 10 by using existing network infrastructure.**

I. Introduction

While the cost per transistor follows Moore's law, test costs do not show a similar behavior. If the same trends continue, it is expected that the cost to test a transistor would become greater than the cost to manufacture it in near future [1]. Every manufactured VLSI (Very Large Scale Integration) chip needs to be tested by placing it in the Automatic Test Equipment (ATE) test- head, applying the generated patterns (usually a combination of 1s and 0s) and verifying the signatures produced in response to the patterns. The major task of an ATE is thus pattern generation and signature verification which may be several seconds for large chips. Most of the test cost is accounted to the huge cost of the ATE. Additionally ATE has limited number of pins which forms another basis for its pricing. This suggests that ATE utilization and hence effective test time reduction becomes a major concern in VLSI testing [1].

A significant improvement in test time can be achieved by testing multiple ICs (Integrated Circuit) in parallel. This is conventionally referred to as Multi-site testing [2]. In this method, the ATE's test pins are shared by more than one IC which can be tested at the same time. The communication mechanism between the ATE and the test-heads in Multi-site testing is tightly coupled and does not scale well. This limits the number of test-heads that can connect to conventional Multi-Site ATE.

A very old approach for test time reduction is by reducing the number of patterns required to test a circuit. In [3] authors present a series of techniques that exploit the inherent parallelism available in test patterns and thus use them to test multiple modules inside a core. Another idea is to reduce the data communicated between the ATE and DUT (Device Under Test) by using compression techniques [4], [5], [6]. Recent years have shown a lot of interest towards Multi-site testing which offers raw parallelism [7]. Efficient resource utilization for Multi-site testing is described in [8]. Another idea to reduce the test cost is by improving tester utilization which is presented in [9].

Packet switched networks have evolved over years and have proved their advantages over circuit switching. A multitude of protocols in various layers have solved problems related to communication [10]. It is an intuitive step to take from master-slave bus systems to a packet switched network when the distributed system's degree of complexity grows. A detail of how Ethernet can be used for such applications is presented in [11]. The author of [12] proposes Ethernet to replace regular bus communication for data acquisition systems. Due to the advancements in network speeds in range of gigabits per second, it now becomes feasible to use them for throughput hungry applications like VLSI testing. With a well structured network and properly designed methodology, highly scalable and resource conscious communication architecture can be developed for ATE based VLSI testing. This was the motivation behind our architecture.

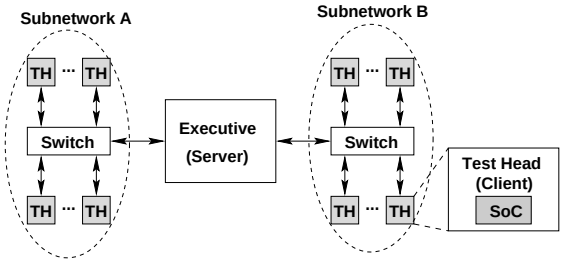

Fig. 1. TAN Architecture

In this paper, we present a parallel architecture for ATE based testing. By using packet switching as the mode of communication between the ATE and test-heads, we make the coupling between them more flexible and hence reduce the communication complexity. This is a problem of significant concern when the number of devices to be tested in parallel increase to more than a handful. By having tens of DUTs tested in parallel we show a dramatic reduction of test time per chip. Another significant novelty in our paper is the high utilization of ATE achieved by distributing a portion of its job to the test heads to work on. The test heads are now capable of verifying the signatures obtained after applying patterns to the SoC.

II. Test Model and Architecture

A. Test Model for TAN

We detail our architecture based on a simple test model which we use throughout this paper. Since our architecture does not depend on the actual test mechanism used, a generic test model is considered. The DUT is modeled to have N_s pins that are used to test it. A test vector is a sequence of bits applied to a pin. If we consider a vector to be L bits long, then a test pattern can be considered as bits of data of length $N_s \cdot L$. There are several DUTs being tested simultaneously and all of them are copies of same circuit. Each DUT takes $N_s \cdot L$ bits of data and produces signatures which are collected simultaneously by the test heads that hold them. The test head has resources to store patterns in a local buffer if necessary and apply them in a burst in case a high speed testing is required.

B. Network Model and Architecture

Test Area Network (TAN) is comprised of ATE and intelligent test heads interconnected. TAN adopts a client-server

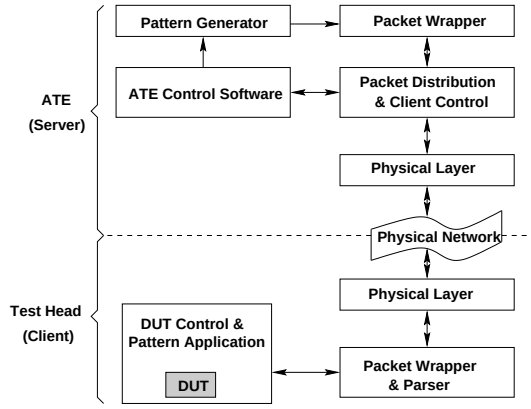

Fig. 2. Client and server architecture

model in which the ATE acts as a server as shown in Figure 1. Each DUT compares to a client and is connected to the network through a switch. A switched network is chosen because collisions and hence delay in frame delivery is not tolerable in our application. A TAN may have multiple sub networks branching from switches and the switches in turn connect to the ATE. The ATE does TDM (Time Division Multiplexing) in addressing these sub networks. This is detailed in Section.

The entire ATE architecture can be mapped into the Client-Server model in which the ATE is the Server and has the complete command and control over the DUTs which are the Clients. A top level architecture for the Server and Client is presented in Figure 2. Server can be modularized as follows:

- A conventional pattern generator - Pattern generation is unmodified and is adopted from the conventional ATE.
- Packet wrapper - Patterns are the payload for the packets which have a header with the network control information.
- A unit for scheduling the packets to various clients.
- MAC and Physical. layer network interface components.

Each Client has the following modules:

- MAC and Physical layer network interface components.
- A unit that parses the packets to extract the commands and patterns from it. It would also be able to wrap response packets back to the server.
- A DUT controller unit that is capable of interpreting the commands in the packet to apply them to the chip.

C. Distributed Testing Scheme

We distribute the testing process between the ATE and clients in order to keep the complexity of ATE low. We try to reduce the communication between the ATE and the DUTs by using the following key principles.

1) Broadcast the patterns to all the clients and let the clients apply the patterns and collect signatures.
2) Broadcast the expected signatures to the clients and hence make them verify the result of test.
3) Let the Clients communicate with the ATE minimally as this dedicates the network for one Client and the ATE.

The philosophy of applying patterns in our architecture takes a different approach. The patterns are made into packets at the Server ATE and are broadcasted to all the clients in the network. By doing this, the Server eliminates the need to address each client to deliver the patterns to be tested for. Our architecture allows the ATE to dispatch hundreds of patterns to the client before a testing is started at the client units. The advantages being reduced packet header overhead and increased ATE utilization which is due to the availability of ATE to service other subnetworks when one is busy applying test procedure.

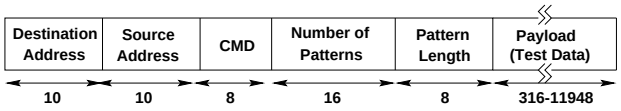

Fig. 3. TAN Protocol header (length shown in bits)

It is completely possible to allow multiple clients to send back the responses they generate to the ATE for verification. But this would require the ATE to be extremely powerful to handle responses for multitude of clients. Moreover, response communication needs to be in an individual basis (unlike patterns which are broadcasted) which loads the network heavily. This can substantially slow down the overall testing process and hence would offset the advantages offered by parallelism. Just as a broadcast allows the ATE to send information to all clients in one burst, we make the ATE to broadcast the expected responses too. The comparison of obtained responses and expected responses happen in the clients individually. If a client finds a difference in this comparison, it indicates the ATE about this condition which in turn would isolate it from network.

III. The TAN Protocol

A. Networking Methodology

Since the application requires a very high speed link between the Server and Client, it is essential that the bandwidth available be utilized most effectively. Packet switched networks have a bandwidth overhead contributed by headers and the need for retransmissions due to collision. A simple protocol for effective communication between the ATE and the clients is developed and named TAN protocol. In its simplest form, the TAN protocol works as a network layer protocol over the MAC layer of the Ethernet LAN in the 7 layer OSI (Open Systems Interconnect) model [13] [14]. The ubiquitous presence of Ethernet, the availability of low cost and high speed (Gigabit) solutions for Ethernet networks were the reasons behind selecting Ethernet as the network infrastructure. We keep the work close to the physical layer to reduce the hardware/software overhead.

B. TAN Packet Format

The application layer of the TAN sits over the MAC layer. We propose a new protocol (see Figure 3) in which:

- Source and destination address fields are 16 bits each. Although Ethernet can support maximum 1024 nodes, 16 bits are selected to keep these fields in the byte boundary.
- An 8 bit field is used for commands and would be referred as CMD field in TAN. This field essentially identifies the type of the frame that is being sent. A list of frame commands has been discussed in detail.
- A 2 byte field indicates the number of patterns that are packed in the payload. This allows up to 2^{16} = 65536 patterns to be packed in a single frame.
- The next byte indicates the length of each pattern in number of bits. This allows 2^8 = 256 bit long patterns.

Following the header is the test data (patterns/expected responses). Although the TAN header allows for larger number and longer patterns, the MAC header restricts the payload size

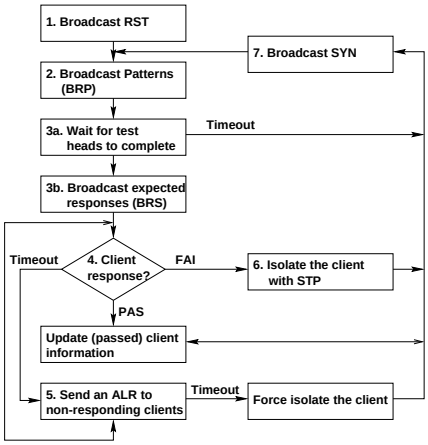

Fig. 4. FSM of TAN Protocol Execution

to a maximum 1500 bytes and at least 46 bytes. Hence, the maximum payload of TAN protocol could be 1492 bytes. This avoids the associated Fragmentation and Re-assembly overheads that would be introduced otherwise.

Since in our design, the test heads are more than dumb probes and that they are intended to reduce the workload of ATE, they can be controlled by ATE using the CMD field of the header. This is an 8 bit field and hence supports 256 different commands. We define the following commands in a TAN:

- BRP - Broadcasted data is patterns, usually by the ATE.
- BRS - Broadcasted data is signatures, usually by the ATE.
- PAS - Clients that pass the test acknowledge with this.
- FAI - Clients that fail the test alert the ATE with this.
- STP - In response to FAI, ATE issues STP (Stop) to the failed client. The failed client is supposed not to interfere in the network until it sees a RST (Reset).
- RST - Upon completion of a test session for a batch of DUTs, the ATE broadcasts RST (Reset) to start testing a fresh batch of DUTs.
- SYN - Broadcasted before a new test sequence is started.
- ERR - Issued by clients to alert the ATE of a problem.
- ALR - This is an Alert signal that the ATE sends to a particular client that failed to respond in expected manner.

C. TAN Protocol Execution

Sequence of operations that occur during a typical testing scenario in our architecture is outlined here. A test session on a batch of DUTs is a series of pattern application and response verification activities until all tests are completed. A test sequence is defined as a cycle in which a set of test patterns are applied to the clients and the produced signatures are verified. A set of clients that failed the test sequence are isolated. Thus, a test session is made of multiple test sequences. FSM of the test execution using the TAN protocol is presented in Figure 4.

A test session starts with an RST command broadcasted by the ATE for which the clients are expected to reset their registers and start operations afresh. A packet of patterns is then broadcasted to the clients on sub network A using the BRP command. After this, the ATE can switch to another sub network B and start operation on that. Since the test clock speed and number of patterns is known, the time required for pattern application and signature collection can be predicted. After this time period, the ATE switches back to sub network A to broadcast the signatures with BRS command. Those clients that have the DUT passing the test respond with PAS and others with FAI. If no response is received from a client, then the ATE interprets it as a communication failure. In this scenario, the ATE sends an ALR command to the non responding client to give it a chance to recover. If the client responds with ERR frame the ATE may continue sending the remaining frames to complete the test sequence. Else, the ATE marks the as isolated does not attend it in the current test session. Each failed client is isolated with STP which remain isolated for the remaining test sequences in the current test session. The ATE then broadcasts a SYN to indicate the active (passed) clients to prepare for the next test sequence. The SYN does the same as RST except that isolated clients do not respond to SYN.

IV. Performance Estimation

Based on the test model described in Section II, we list the following parameters:

$$\begin{cases} \text{Width of a pattern} & = & N_s \text{ bits} \\ \text{Number of patterns required per test} & = & N_p \\ \text{Number of DUTs being tested in the TAN} & = & N \\ \text{Minimum frame PDU size} & = & F_{min} \text{ bits} \\ \text{Maximum frame PDU size} & = & F_{max} \text{ bits} \\ \text{Number of DUTs that succeed the test} & = & Y \\ \text{Header overhead per frame} & = & H \text{ bits} \\ \text{Total bandwidth of the Ethernet LAN} & = & T \text{ bps} \\ \text{Available bandwidth for testing} & = & B \text{ bps} \end{cases}$$

Let us denote $N_s \cdot N_p = k \cdot F_{min}$, which tries to represent the total test data size as a multiple of the minimum frame size. A conventional ATE that tests one DUT at a time uses all the available bandwidth with much less control overhead. In its most simplest form:

$$\begin{cases} \text{Test clock} & = & T \text{ Hz} \\ \text{Equivalent bandwidth} & = & T \text{ bps} \\ \text{Total test data (pattern size)} & = & N_s \cdot N_p = k \cdot F_{min} \text{ bits} \\ \text{Test time per DUT in seconds} & = & k \cdot F_{min}/T \end{cases}$$

A. Available Bandwidth for Testing

$$\begin{cases} \text{Header overhead per frame} & = & H \text{ bits} \\ \text{Fraction of bandwidth utilized } (B) & = & \frac{T \cdot k \cdot F_{min}}{H + k \cdot F_{min}} \end{cases}$$

The total header size of the MAC protocol including the idle time, preamble, is 38 bytes.

$$\begin{cases} \text{Frame size} & = & \text{Header size + Payload size} \\ \text{Max. frame size} & = & 38 + 1500 = 1538 \text{ bytes} \\ \text{Min. frame size} & = & 38 + 46 = 84 \text{ bytes} \end{cases}$$

The TAN protocol itself has just 8 bytes overhead. The overall header overhead can be computed to show that larger frames use the bandwidth more efficiently.

$$\begin{cases} \text{Total TAN protocol overhead} & = & 38 + 8 = 46 \text{ bytes} \\ \text{Header overhead (1538 byte frame)} & = & 2.99\% \\ \text{Header overhead (84 byte frame)} & = & 54.76\% \end{cases}$$

Although no collision is definitely unreal, collisions could be practically neglected in our architecture which uses a switched topology. Additionally, queuing in switches can alleviate potential contention at the link from the switch back to the ATE

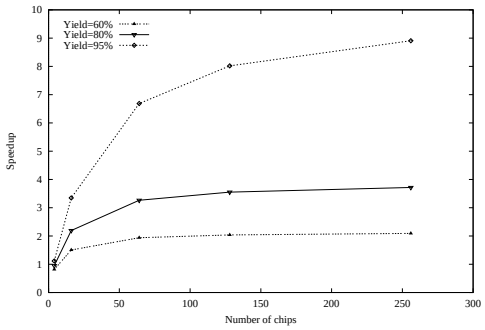

Fig. 5. Plot of speedup vs number of chips under test

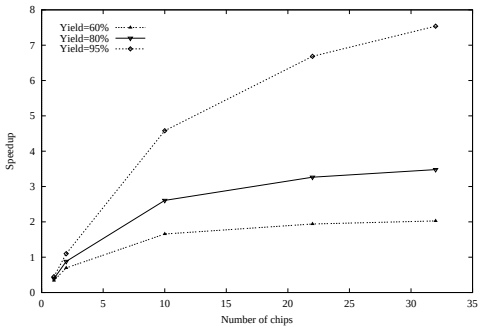

Fig. 6. Plot of speedup vs k

when more than one DUT tries to send a frame to the ATE. In the analysis that follows, this has been addressed by treating all responses as bits of data that are to be transported in a serial fashion. Since the number of frames that travel from clients to ATE are a small fraction of the overall frames transported, the queuing latency has been neglected.

B. Effective Test Time per DUT

Based on Figure 4, an analysis has been done which assumes that there are no network failures. The number of bits required for various commands are: BRP/BRS ($N_s \cdot N_p$), PAS ($Y \cdot F_{min}$), FAI ($(N-Y)\cdot(N_s \cdot N_p)$), SYN/RST ($F_{min}$), STP ($(N-Y)\cdot F_{min}$). The total number of bits of data exchanged in the TAN for one set of N_p patterns is, $(Y+1)\cdot F_{min} + (3+N-Y)\cdot(N_s \cdot N_p) + (N-Y)\cdot F_{min}$. With the available bandwidth, the time t in seconds required to test N_p patterns on N DUTs is:

$$t = (H + k \cdot F_{min}) \cdot (1 + N + k \cdot (3 + N - Y))/(T \cdot k)$$

Having t, we have: *Time per DUT* $=t/N$.

C. Factors Affecting Performance

The expression for test time (t) derived in the previous section, depends in a non linear way on N (number of chips tested in parallel) and $k(= N_s \cdot N_p/F_{min}$, a factor that is directly proportional to size of test data sent in a frame). A decision on these factors could be effectively made if we determine their effects on speedup. The process yield (Y/N) of a VLSI manufacturing process is defined as the percentage of fault free parts among all parts that are fabricated. As the yield decreases, a greater percentage of bandwidth is used for handling failed clients and hence the overall test time increases. Common factors in these computations are:

$$\begin{cases} \text{Header size} & = & 46 \times 8 = 368 \text{ bits} \\ F_{min} & = & 46 \times 8 = 368 \text{ bits} \\ T & = & 100,000,000 \text{ bps} \end{cases}$$

The curves in Figure 5 drawn with $k = 22$, convey the speedup versus number of chips per test. Increasing the number of test heads does not give a linear performance improvement. At lower yield levels, a significant percentage of bandwidth is utilized for handling communication between failed clients in a one-one basis and so results in increased average test time. On the other hand, when the yield level is high (above 90% as often is the case), a better increase in speedup can be achieved.

The curves in Figure 6 drawn for $N = 64$, shows the speedup versus k (proportional to test data size). They indicate that as the test data size (proportional to k) per frame increases, the bandwidth is used more efficiently and hence a better speedup can be achieved.

TABLE I

RESULTS FOR ISCAS89 CIRCUITS

Metrics	*S13207F*	*S38584F*	*S35932F*
Bit Count	165672	199376	28240
Test Time (Conv.)[sec]	1.060	1.267	0.180
Test Time (TAN)[sec]	0.151	0.182	0.028
Speedup	6.98	7.0	6.42

Table I shows the results for actual test time by running the formulation for t derived in the previous subsection on the ISCAS89 benchmark circuits [2]. With all other factors remaining same and $N = 64$ and 90% yield level, different k values were computed for the data size. At least 85% reduction is test time can be clearly seen in each case.

REFERENCES

[1] The International Technology Roadmap for Semiconductors, "Test and Test Equipment", 1999, http://public.itrs.net/fi les/1999_SIA_Roadmap.

[2] Bushnell, M. L., Agrawal, V. D., "Essentials of Electronic Testing", Massachusetts: Kluwer Academic Publishers, 2000.

[3] Ravi, S., Lakshminarayana, G., Jha, N.K., "Reducing test application time in high-level test generation", Proc. Int. Test Conf., pp. 829-838, 2000.

[4] Jain, V., Waicukauski, J., "Scan test data volume reduction in multi-clocked designs with safe capture technique", Proc. Int. Test Conf., pp. 148-153, 2002.

[5] Nourani, M., Chin, J., "Testing High-Speed SoCs Using Low-Speed ATEs", Proc. VLSI Test Symposium, pp. 133-138, 2002.

[6] Jas, A. , Ghosh, J., Touba, N.A., "Scan vector compression/decompression using statistical coding", Proc. VLSI Test Symposium, pp. 114-120, 1999.

[7] Volkerink, E.H., Khoche, A., Rivoir, J., Hilliges, K., "Test Economics for Multi-Site Test with Modern Cost Reduction Techniques", Proc. VLSI Test Symposium, pp. 411-416, 2002.

[8] Iyengar, V., Goel, S.K., Marinissen, E.J., Chakrabarty, K., "Test Resource Optimization for Multi-Site Testing of SoCs Under ATE Memory Depth Constraints", Proc. Int. Test Conf., pp. 1159-1168, 2002.

[9] Khoche, A., Kapur, R., Armstrong, D., Williams, T., Tegethoff, M., Rivoir, J., "A New Methodology for Improved Tester Utilization", Proc. Int. Test Conf., pp. 916-923, 2001.

[10] Stallings, W. "High-Speed Networks and Internets Performance and Quality of Service", New Jersey: Prentice Hall Inc., 2002.

[11] Swales, A., Gray, C., "Transparent factories through industrial internets", Canadian Conf. on Electrical and Computer Engineering, pp.931-936, 1999.

[12] Potter, D., "Using Ethernet for Industrial I/O and Data Acquisition", Proc. of Instrumentation and Measurement Tech. Conf., pp. 1492-1496, 1999.

[13] Spohn, D. "Data Network Design", McGraw-Hill, 1997.

[14] IEEE 802.3: CSMA/CD Access Method. http://standards.ieee.org/ getieee802/download/802.3-2002.pdf