ZIP-ATE: Zero-to-Infinity Pins ATE Using Packet Switched Network

M. Nourani and S. Vengatachalam
Center for Integrated Circuits & Systems
The University of Texas at Dallas
Richardson, TX 75083
{nourani,sxv022000}@utdallas.edu

***Abstract*— We present an Automatic Test Equipment architecture that is not limited by the number of pins it can serve. To achieve this, we introduce the idea of using Packet Switched Network as the mode of communication between ATE and the VLSI chip under test. We show that our architecture which we refer to as ZIP-ATE (Zero-to-Infinity Pin) reduces the complexity and time involved in testing tens of chips at a time by a Multi-Site test philosophy. To increase the ATE utilization and available communication bandwidth, we distribute main portion of ATE's task of signature verification to the test-heads. These test-heads are no more dumb probes but are capable of applying patterns and verifying signatures produced by the chip to which they are connected. We analyze various factors that determine the performance of our architecture. Our analysis and empirical results indicate a speedup of 4 to 10 by using existing off-the-shelf components and network infrastructure.**

Keywords: Automatic Test Equipment, Medium Access Control, Multi-Site Testing, Network Layer Protocol, Packet Format, Packet Switched Network, Test Area Network, Test Head.

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]. The need of the hour is the capability of the ATE (Automatic Test Equipment) to test several devices without introducing enormous cost or complexity [1]. Every manufactured VLSI (Very Large Scale Integration) chip needs to be tested by placing it in the ATE testhead, applying the generated patterns (usually a combination of 1s and 0s for digital Integrated Circuits) and verifying the signatures (responses of the Integrated Circuits for the applied patterns) produced. Figure 1 portrays this practice. The major task of an ATE is thus pattern generation and signature verification. For large chips, millions of such patterns are applied and the test time typically is several seconds. Current multi million transistor SoCs (System on Chip) are far more complex to test than manufacture and hence take a greater ATE time to be tested. Most of the test cost is accounted to the huge cost of the ATE, as literally every millisecond of ATE time counts. Additionally ATE has a 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 has multiple test-heads and more than one IC can be tested at the same time. The motivation behind this idea is that it is far cheaper to add test-heads than to deploy multiple ATEs.

Fig. 1. Testing a VLSI Chip

The communication mechanism between the ATE and the DUT (Device Under Test) in Multi-site testing is a tightly coupled one and every connection between the ATE and a DUT is called a Channel. Each channel is formed by a fixed number of pins to which an ATE is supposed to serve. Tight coupling places the burden of managing the complexity involved on the ATE software and hence limits the number of pins, eventually reducing the number of DUTs that can be tested by the ATE at the same time. ATE utilization is key factor for the growing concern of reducing the manufacturing test cost and speeding up time to market. In this paper we introduce a mechanism that allows the ATE to be loosely coupled to the DUTs, i.e. a packet switched local area network. The immediate advantage is less complex interface, more ATE utilization and ability to support tens of test-heads simultaneously.

A. Prior Work

Test time reduction is a goal which has been approached in various angles. One approach of reducing test time is by reducing the number of patterns required to test a circuit. This is the most visited approach and is very old too. 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 lower the data communicated between the ATE and DUT by using compression techniques [4], [5], [6], [7]. To an extent, this improves the utilization of ATE. Recent years have shown a lot of interest towards Multi-site testing which offers raw parallelism in terms of reducing test cost [8] and complexity. Efficient resource utilization for Multi-site testing is described in [9]. Another idea is to reduce the test cost by improving tester utilization which is presented in [10].

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 [11]. The scenario in which tens of devices need to exchange data with a single central master is a common application in networking and can be readily mapped to client-server architecture. 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. Local Area Networks (LAN) have been evolved phenomenally. The prices of

ready to use LAN devices (such as Ethernet cards) are always shrinking while their performance and reliability are ever improving. Use of networking for industrial applications has been proposed in the literature [12], [13]. A detail of how Ethernet can be used for such applications is presented in [13]. The author of [12] proposes Ethernet to replace regular bus communication for data acquisition systems. Employing a packet switched technology for an industrial purpose such as IO acquisition or VLSI testing improves scalability, performance and reduces complexity.

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.

B. Contribution

In this paper, we present a parallel architecture for ATE based testing. We refer to test-head as an interface to the DUT. It is capable of processing information sent by the ATE's *Executive*. Executive is the section of the ATE that is capable of generating patterns and controlling the overall test activity. By using packet switching as the mode of communication between the Executive and test-heads, we make the coupling between them more flexible and hence reduce the complexity involved in testing multiple chips. 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. At present we propose this architecture only for digital functional testing. We hope to expand this idea to other methods of testing like IDDQ and parametric test [2] in near future.

C. Paper Organization

Following the Introduction, Section II describes the overall architecture in detail. Section III selects key-portions of the architecture that contribute to the novelty and performance and describes them vividly. Section IV describes the realization of ZIP-ATE by using two configurations. Section V analyzes various factors that govern the performance of our architecture and derives a relation among them. Typical performance improvements are also indicated. Section VI concludes the paper by summarizing the advantages and providing ideas for expansion.

II. Test Model and Architecture

A. Test Model

We detail our architecture based on a simple test model which we use throughout this paper. Since our architecture presents an efficient communication mechanism between the Executive and the test head, it does not depend on the actual test mechanism (eg. scan) used. The DUT is modeled to have

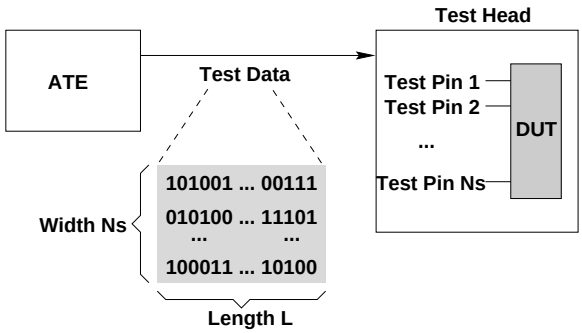

Fig. 2. Test model

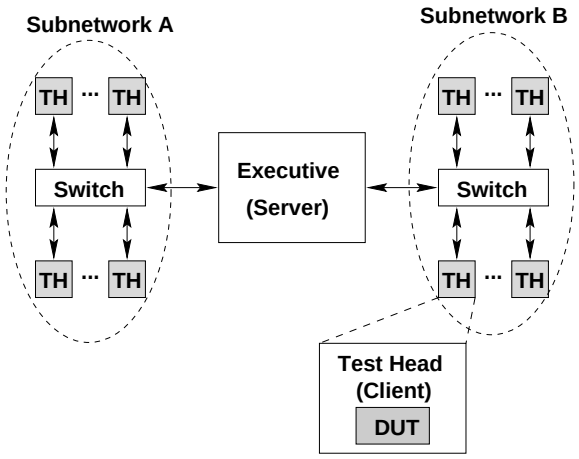

Fig. 3. TAN Architecture

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 total 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 back by the test head. Figure 2 shows these details. In case a high speed testing is required, the test head can have a local buffer to store patterns and apply them in a burst.

If Scan based testing [2] is performed, the following could be an interpretation for a pattern set. Every DUT is assumed to have N_s scan chains. Every input pin feeds one scan chain with the test data. Thus, if a scan chain is L in length, a test vector is made up of L bits. Once the scan chains are filled, every further input produces an output bit and is captured by the test head.

B. Conventional ATE Architecture

Conventional ATE generates patterns and applies them to the DUT which plugs itself into the test-head. The DUT produces signatures which are captured back to the ATE and are compared with the expected results. State of the art ATE can do a variety of tests on the chip including analyzing the input/output waveforms on every pin. In a conventional ATE, a test head is merely a socket that is capable of applying signals to the DUT. In Multi-site testing, pins in the test head are shared by more than one DUT and the signals that appear on the pins are controlled by the ATE software. Note carefully that multi-site

testing is not as easy as extending a shared bus and connecting multiple DUTs for parallelism. The ATE is required to be very powerful i.e., additional buffering, protection and isolation circuitries need to be devised, and the software must be very robust, to handle multiple DUTs in parallel.

C. Network Topology

Interconnection of intelligent test heads and the ATE Executive is termed Test Area Network (TAN). TAN adopts a client-server model in which the Executive acts as a server as shown in Figure 3. Each DUT compares to a client and is connected to the network through a switch. A switch takes care of receiving the frames it receives from the Executive and sends them to the appropriate client. A switched network is chosen because it avoids collisions and hence frame delivery is quickened. A TAN may have multiple sub networks branching from switches and the switches in turn connect to the Executive. The Executive does TDM (Time Division Multiplexing) in addressing these sub networks. This is possible due to the availability of the Executive after applying patterns to one sub network and allowing the clients in that sub network to apply patterns to their DUTs to get their responses.

D. Client-Server Architecture

The entire ATE architecture can be mapped into the Client-Server model in which the Executive is the Server and has the complete command and control over the test heads (that includes DUTs) which are the Clients. A top level architecture for the Server and Client is presented in Figure 4.

Server can be modularized as follows:

- A conventional pattern generator - Pattern generation is unmodified and is adopted from the conventional ATE. The modular nature of the networking implemented in our architecture presents a transparent interface for pattern generation.
- Packet wrapper - Patterns are the payload for the packets which have a header with the network control information. The control information is provided by the ATE control software.
- A unit that takes care of scheduling and queuing the packets to various clients.
- MAC (Media Access Control) and Physical layer network interface components.
- ATE Control Software which monitors and controls various test activities.

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 holds the chip. It is capable of interpreting the commands in the packet to apply them to the chip

The idea behind having a modular architecture is to reuse the existing ATE architectures by making minimum changes.

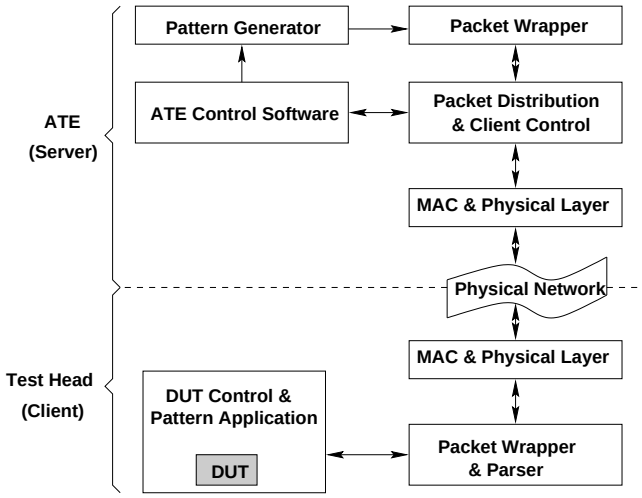

Fig. 4. Client and Server architecture

E. Distributed Testing Scheme

While multi-site testing offers test time reduction using parallelism, it requires the ATE to manage all tasks involved in testing all the DUTs. As we know that packet switched network does come with a header overhead, it is necessary to keep the traffic between the Executive and the test heads minimal in order to maintain high utilization of the Executive. Our proposal addresses this problem to a large extent by following principles. They are:

1) Broadcast the patterns to all the clients
2) Allow the clients to apply the patterns to the DUT and capture the signature in response.
3) Broadcast the signatures to the clients and hence make them verify the obtained result with the expected result.
4) When necessary, allow peer-peer client communication.

These principles can be explained in two phases necessary in testing, i.e., pattern broadcast and response verification.

1) Pattern Broadcast: Our architecture suggests a network that has all clients in the same state in every point in time during testing. By this it is conveyed that the testing process takes place in a synchronized manner. In a conventional testing process, the patterns are applied one at a time and the response from the DUTs is captured by the ATE. The philosophy of applying patterns in our architecture takes a different approach. The patterns are made into packets at the Server 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. A 'stop-upon-failure' policy is used in which the moment an unexpected response turns out, the testing is stopped and the device is declared 'Failed'. On the other hand, our architecture allows the ATE to dispatch hundreds of patterns to the client before a testing is started at the client units. Following are the advantages in this method.

- Forward bandwidth utilization is high due to reduction of packet overhead. Since multiple patterns are transferred in a burst in a single packet, a very small overhead for packet header data is required.
- After a set of patterns are broadcasted to a sub-network A, the Executive is free to attend another sub-network B

until the signatures are ready in sub-network A. This increases the ATE utilization significantly and hence reduces the overall test cost.

2) Response Verification at Client: A conventional ATE captures the signature produced in response to a pattern applied and compares it with the expected output. This operation takes place in a pin by pin basis. It is possible to extend this method to multiple clients by making them 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 creates a huge latency. Having these implemented might offset the advantages obtained by parallelism. Just as a broadcast allows the Executive to send information to all clients in one burst, we make the Executive 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 Executive about this condition. It is now the responsibility of Executive to isolate this salve from network. Since this method distributes a major portion of the Executive's load to clients, a considerable reduction in complexity at the Executive in handling responses from multiple clients is achieved.

III. TAN Protocol Methodology

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 [14] [15]. 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. By keeping the work close to the physical layer, the additional hardware/software overhead is reduced.

B. TAN Packet Format

The application layer of the TAN sits over the MAC layer. We propose a new protocol in which:

- Source and destination address fields are 16 bits each, although the Ethernet LAN can support maximum 1024 attached units. This makes these fields byte aligned.
- 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 in next section.
- 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.

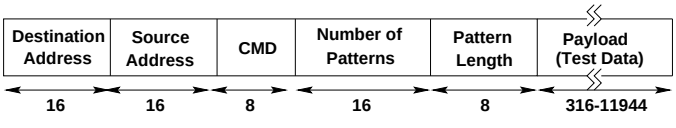

Fig. 5. TAN Protocol header (length shown in bits)

- The next byte indicates the length of each pattern in number of bits. This allows 2^8 = 256 bit long patterns.

It can be observed that TAN protocol header is 64 bits wide. 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 to a maximum 1500 bytes and at least 46 bytes. Hence, the maximum payload of TAN protocol should be restricted to 1492 bytes to avoid fragmentation at the MAC layer.

C. TAN Commands

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 although not all of them are defined now. We define the following commands that are basically needed for a TAN:

- BRP - The broadcasted data is patterns, usually by the Executive
- BRS - The broadcasted data is signatures, usually by the Executive to allow the clients to verify the generated responses.
- PAS - In response to BRS, the clients that passed the test send this as the acknowledgment.
- FAI - In response to BRS, the clients that failed the test send this as the acknowledgment. Optionally, the payload from these frames is transferred to the Executive for further processing.
- STP - In response to FAI, the Executive issues STP (Stop) to the failed client. The failed client is required not to interfere in the network and not to send any more frames until it sees a RST (Reset) being broadcasted by the Executive.
- RST - Upon completion of a complete testing for a batch of DUTs, the Executive broadcasts RST (Reset) and thus commands the clients to reset all their counters and registers to start testing a fresh batch of DUTs.
- SYN - This command is broadcasted before a new test sequence is started. All clients are required to initialize their registers and prepare for the test sequence.
- ERR - In its simplest form, any unit in the TAN can issue this command to alert the Executive of a potential problem that it is not able to recover from. In response, the Executive isolates the client by issuing an STP to it.
- ALR - This is an Alert signal that the Executive sends to a particular client that failed to respond in expected manner.

D. TAN Protocol Execution

VLSI Testing being a real-time communication between Executive and the test heads, it is required to keep the delay due to packet delivery under control. A switched network topology provides us with a virtually collision free communication.

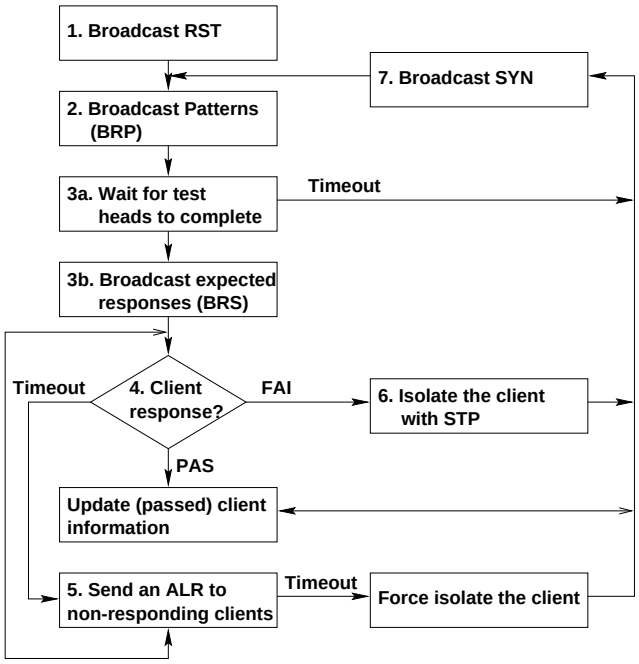

Fig. 6. FSM of TAN Protocol Execution

Moreover, TAN is a network that spreads out in a very short range and the communication is deterministic. The susceptibility of the network to packet errors can be minimized as the network is built in an environment which has high reluctance to noise. With all this in place, we use a connection-less protocol that would keep the delay low as well as reduce the overhead. Sequence of operations that occurs 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 6.

1) A test session always starts with an RST command broadcasted by the Executive. Upon the receipt of this command, the clients are expected to reset their registers and start operation afresh. This is required to ensure that all clients are in a "ready" condition and those clients that were isolated in a previous test session are included for this session.
2) Executive then starts the first sequence with a packet of patterns broadcasted to the clients on sub network A using the BRP command. After this, the Executive can switch to another sub network B and start or continue operation on that. This scheme is adopted to keep the Executive busy when a sub- network is applying patterns and collecting signatures.
3) 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 Executive switches back to sub network A to broadcast the signatures with BRS command.
4) In response to BRS those clients that have the DUT passing the test respond with PAS and others with FAI.
5) If no response is received from a client, then the Executive interprets it as a communication failure. This may happen due to frame dropping in the network. In this scenario, the Executive sends an ALR command to the non responding client to give it another chance to recover. If the client received this frame, it responds with ERR frame in which the first byte of the payload corresponds to the last successful command it received. Upon receiving this, the Executive may continue sending the remaining frames to complete the test sequence. If the client did not respond to ALR, the Executive updates its information about that client as isolated and this client would no more be attended in the current test session.
6) The Executive then isolates the failed clients with STP which is addressed to each client. These clients remain isolated for the remaining test sequences in the current test session.
7) After updating its information about the passed clients, the Executive then broadcasts a SYN to indicate the active clients to prepare for the next test sequence. The SYN does the same as RST except that isolated clients do not respond to SYN.

Between any two successive packet transmissions, the Executive waits for a time greater than round trip delay (t_r) of a frame in the network. This delay allows a failed client to respond with an ERR to the Executive.

IV. ZIP-ATE REALIZATION

The previous sections gave an in depth idea of how the TAN works. Conceptually, the TAN does not depend on the underlying physical network medium because it is essentially a Network layer protocol. This gives us the freedom of using different physical media depending on the bandwidth and implementation requirements. Widespread availability of Fast Ethernet and Gigabit Ethernet gives us a low cost medium that has sufficient bandwidth to support a 100 DUTs as can be observed from Figure 9. Another manifestation of TAN could be using the popular 802.11 wireless as the physical medium. An implementation of the schemes would result in an ATE that would either get rid of pins (Zero pin configuration) or have the capability to serve any number of pins (Infinite pin configuration)

A. Zero pin ATE

Figure 7 presents a configuration for the TAN architecture in which the Executive and the DUTs use the popular 802.11 wireless network as the physical medium. This medium allows a gross bandwidth of 11 Mbps in 802.11b [16] and 54 Mbps in 802.11a [17] standards. In this method, the DUT does not rely on the test head for the test data. Each DUT is built with the necessary hardware to perform communication directly with the Executive. Distribution of test data and collection of signatures from various circuit segments inside the DUT is taken care of by the Test Controller that resides inside the DUT. Each DUT still needs to be provided with power supply and passive components to complete the hardware necessary for wireless

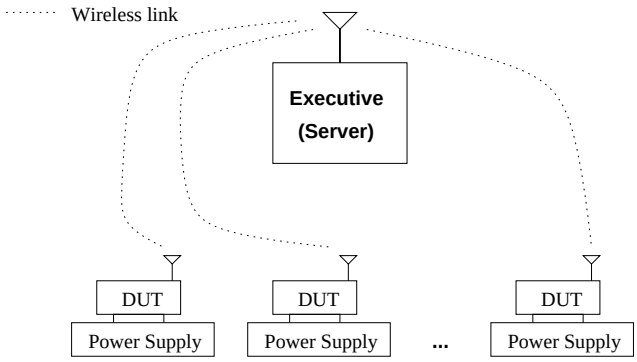

Fig. 7. Zero-pin confi guration

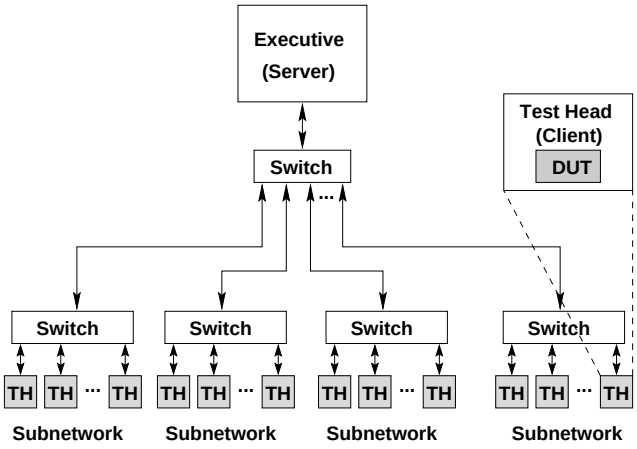

Fig. 8. Infi nite-pin confi guration

network. Thus the test data exchange is no more limited by number of pins.

Advantages of Zero pin configuration:

- The testing procedure is not limited by the number of pins present in the DUT or the ATE.
- Test setup is much simpler with much less test hardware.
- The additional circuitry in the chip needed for wireless communication is a well established one and is much cheaper when compared to cost of complex DUTs.
- As 802.11 uses the Ethernet protocol in the MAC layer, there is no need for modification of TAN protocol.

Limitations of Zero pin configuration:

- The gross bandwidth of Wireless network is less than conventional wired Ethernet network and hence this would support lesser number of DUTs at a time.
- Every manufactured chip would have additional logic circuitry to provide for wireless communication. But this is not a serious limitation due to high integration densities available in today's State-of-the-art SoCs.

B. Infinite pin ATE

Figure 8 presents a configuration for the TAN architecture in which the Executive and the DUTs use the regular wired Ethernet network as the physical medium. This medium allows a gross bandwidth of 100 Mbps in Fast Ethernet and 1 Gbps in Gigabit Ethernet standards. The requirement of this configuration is high speed testing capable of supporting many DUTs in parallel. The inherent nature of Ethernet network is the ready scalability. Clients can join or leave the network at any time (though this would be limited by the TAN to the start of a test session). By adopting a star topology using switches, maximum bandwidth utilization can be achieved.

Advantages of Infinite pin configuration:

- The testing procedure is not limited by the number of pins present in the DUT. Any number of pins can be served by the Executive as long as the DUT is present in the Test Area Network.
- Test heads use the cheap and readily available Ethernet hardware.
- Unlike Zero pin configuration, this does not need a any special hardware inside the DUT.

Limitations of Infinite pin configuration:

- Although the name suggests that this configuration can serve any number of DUTs, the performance is limited by the bandwidth provided by the medium. Moreover, a very high number of DUTs would increase the control overheads and hence degrade performance.

V. Performance Estimation

Since the architecture uses the network using a two layer protocol structure, the performance estimation too would be done in a hierarchical manner. Thus, the goal is to:

1) determine the available bandwidth provided for sending and receiving test data.
2) determine the effective test time required per DUT in a TAN network.

Based on the test model described in section II.A, we list the following parameters:

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

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,

Test clock	=	T hz
Bandwidth equivalent of test clock	=	T bps
Total test data (pattern size)	=	$N_s \cdot N_p$ bits
	=	$k \cdot F_{min}$ bits
Test time per DUT in seconds	=	$k \cdot F_{min}/T$

A. Available Bandwidth for Testing

Total test data (pattern size)	=	$N_s \cdot N_p$ bits
Header overhead per frame	=	H bits
Fraction of bandwidth utilized by test data: B	=	$\frac{T \cdot N_s \cdot N_p}{H + N_s \cdot N_p}$
	=	$\frac{T \cdot k \cdot F_{min}}{H + k \cdot F_{min}}$

TABLE I

ETHERNET HEADER SIZE

Header Field	*Size[Byte]*
Idle Time	12
Preamble	8
Source Address	6
Destination Address	6
Type	2
CRC	4
Total	38

The total header size of the MAC protocol is computed from the Table I as follows:

Frame size = Header size + Payload size
Max. frame size = 38 + 1500 = 1538 bytes
Min. frame size = 38 + 46 = 84 bytes

Since the TAN protocol itself has just a 64 bits (8 bytes) overhead, the overall header overhead is computed as,

Total TAN protocol overhead =38(Eth.)+8(TAN) =46 bytes
Header overhead (1538-byte frames) = 100 × 46/1538 = 2.99%
Header overhead (84-byte frames) = 100 × 46/84 = 54.76%

It is clear that larger frames use the bandwidth more efficiently. The typical bandwidth available on a 100Base-T Ethernet LAN, considering maximum frame size and "no collision" assumption is:

$$B = 100000000 \times (100 - 2.99) = 97.01 Mbps$$

Although no collision is definitely unreal, collisions could be practically neglected in our architecture because it uses a switched Ethernet topology.

B. Effective Test Time per DUT

From section III.D a test on a TAN takes the following stages. The data in the parenthesis represent the number of bits required for the action. This analysis assumes that there are no failures due to network problems. This assumption is reasonable and was explained in section III.D.

1) ATE broadcasts a SYN/RST (F_{min})
2) ATE broadcasts patterns using BRP ($N_s \cdot N_p$)
3) ATE broadcasts signatures using BRS ($N_s \cdot N_p$)
4) Succeeded DUTs respond with PAS frames ($Y \cdot F_{min}$) while failed DUTs respond with the signatures ($(N-Y) \cdot (N_s \cdot N_p)$)
5) ATE sends an STP to each failed DUT ($(N-Y) \cdot F_{min}$)

The total number of bits of data exchanged in the TAN for one set of N_p patterns is,

$$(Y+1) \cdot F_{min} + (3 + N - Y) \cdot (N_s \cdot N_p) + (N - Y) \cdot F_{min}$$

The total bits per test (N_{bpt}) will be:

$$\begin{aligned} N_{bpt} &= F_{min} \cdot (1 + Y + 3k + (N - Y) \cdot (1 + k)) \\ &= F_{min} \cdot (1 + N + k \cdot (3 + N - Y)) \end{aligned}$$

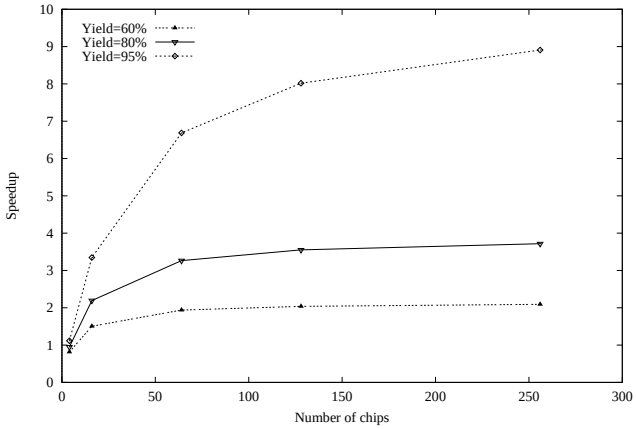

Fig. 9. Plot of speedup vs number of chips under test

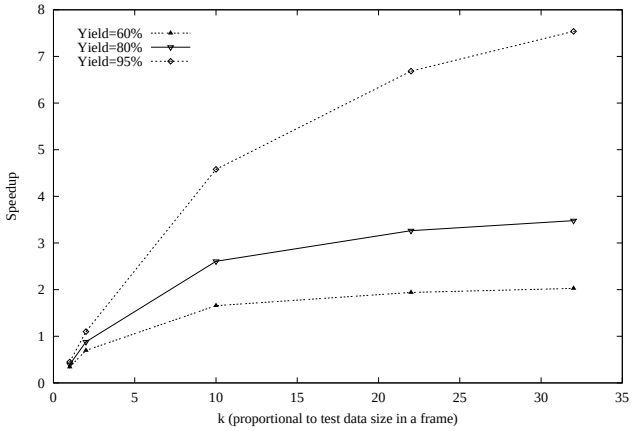

Fig. 10. Plot of speedup vs k

With the available bandwidth B, the time t in seconds required to test N_p patterns on N DUTs is,

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

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.

Hence, we plot Speedup for various yield levels at the test time. It is very important to note that the speedup computation is limited to the use of bandwidth between the ATE and the DUT. Common factors in these computations are:

Header size = $46 \times 8 = 368$ bits
F_{min} = $46 \times 8 = 368$ bits
T = $100,000,000$ bps

- **Speedup vs Number of Chips Per Test**

The curves in Figure 9 drawn with k = 22, convey that in-

TABLE II

TEST TIME FOR ISCAS89 CIRCUITS USING 100BASE-T ETHERNET NETWORK

Metrics	*S13207F*	*S38584F*	*S35932F*
Bit Count	165672	199376	28240
Test Time (Conv.)[sec]	0.1060	0.1267	0.0180
Test Time (100Base-T TAN)[sec]	0.0151	0.0182	0.0028
Speedup	6.98	7.0	6.42

TABLE III

TEST TIME FOR ISCAS89 CIRCUITS USING 802.11A WIRELESS NETWORK

Metrics	*S13207F*	*S38584F*	*S35932F*
Bit Count	165672	199376	28240
Test Time (Conv.)[sec]	0.1060	0.1267	0.0180
Test Time (Wireless TAN)[sec]	0.0281	0.0337	0.0052
Speedup	3.77	3.76	3.46

creasing the number of test heads does not give a linear performance improvement. At lower yield levels, a significant percentage of bandwidth is consumed 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.

- **Speedup vs k (Proportional to Test Data Size)**
  The curves in Figure 10 drawn for N = 64, 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.
- **Actual Test Time**
  Table II shows the results for test time by running the formulation on the ISCAS89 benchmark circuits [2] for 100Base-T Ethernet. Table III shows the test time and improvement for the TAN using 802.11a [17] wireless as the network medium. With all other factors remaining same and N=64 and 90% yield level, different k values were computed for the data size. A 100 MHz test clock was used to determine the performance of a conventional ATE. At least 85% reduction in test time for the 100Base-T TAN and 70% reduction in test time for the wireless TAN can be clearly seen for each benchmark circuit.

VI. CONCLUSION

Testing of VLSI chips contribute a major portion of manufacture cost. To utilize the ATE more efficiently and hence reduce test cost, a distributed architecture was proposed. To allow testing many more devices in parallel when compared to today's multi-site testing, packet-switched network was introduced. By using well established, ubiquitous Ethernet protocol over LAN as the medium, a protocol was designed to facilitate testing. The deployment of wireless network for the TAN, to free the ATE off the pin count limitations was illustrated. Another configuration that makes the TAN a highly scalable testing solution was presented. A relation involving various factors that affect the performance of TAN network was computed. Using typical values, a simple comparison between conventional and ZIP-ATE architecture was presented to show the performance improvement.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation CAREER Award #CCR-0130513.

REFERENCES

[1] The International Technology Roadmap for Semiconductors, "Test and Test Equipment", 1999, http://public.itrs.net/files/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] Chandra, A., Chakrabarty, K., "Test data compression for system-on-a-chip using Golomb codes", Proc. VLSI Test Symposium, pp. 113-120, 2000.
[5] 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.
[6] Nourani, M., Chin, J., "Testing High-Speed SoCs Using Low-Speed ATEs", Proc. VLSI Test Symposium, pp. 133-138, 2002.
[7] Jas, A. , Ghosh, J., Touba, N.A., "Scan vector compression/decompression using statistical coding", Proc. VLSI Test Symposium, pp. 114-120, 1999.
[8] 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.
[9] 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.
[10] 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.
[11] Stallings, W. "High-Speed Networks and Internets Performance and Quality of Service", New Jersey: Prentice Hall Inc., 2002.
[12] Potter, D., "Using Ethernet for Industrial I/O and Data Acquisition", Proc. of Instrumentation and Measurement Tech. Conf., pp. 1492-1496, 1999.
[13] Swales, A., Gray, C., "Transparent factories through industrial internets", Canadian Conf. on Electrical and Computer Engineering, pp.931-936, 1999.
[14] Spohn, D. "Data Network Design", New York: McGraw-Hill, 1997.
[15] IEEE 802.3: CSMA/CD Access Method. http://standards.ieee.org/getieee802/download/802.3-2002.pdf
[16] IEEE 802.11b: IEEE Standard 802.11b http://standards.ieee.org/getieee802/download/802.11b-1999.pdf
[17] IEEE 802.11a: IEEE Standard 802.11b http://standards.ieee.org/getieee802/download/802.11a-1999.pdf