

A Packet Switching Communication-Based Test Access Mechanism for System Chips

Mohsen Nahvi[‡] and André Ivanov[‡]
Department of Electrical and Computer Engineering
The University of British Columbia
2356 Main Mall, Vancouver, B.C. V6T 1Z4, Canada
Telephone: [‡]1 – 604 – 439 9763, [‡]1 – 604 – 822 6936
{mohsenn,ivanov}@ece.ubc.ca

ABSTRACT

In this paper, a *Test Access Mechanism (TAM)* architecture based on a packet switching communication network is presented. The basic goal is to develop a modular, generic, and configurable TAM. Core access time and interconnect length models and simulation results for the proposed architecture are also presented and compared to that of a bus-based TAM. The proposed architecture provides a modular TAM that provides two levels of scalability, i.e. *design-version scalability* and *multi-level scalability*. It is also shown that the proposed TAM, unlike a bus-based architecture, can function without any restriction on the number of primary input/output pins. Furthermore, it is estimated that the total wire-length in our architecture will be less than one percent of the total wire-length of the chip, compared to more than one hundred percent in the case of a bus-based TAM architecture.

Topic: Test of Embedded Cores and System-on-Chip

A Packet Switching Communication-Based Test Access Mechanism for System Chips

ABSTRACT

In this paper, a *Test Access Mechanism* (TAM) architecture based on a packet switching communication network is presented. The basic goal is to develop a modular, generic, and configurable TAM. Core access time and interconnect length models and simulation results for the proposed architecture are also presented and compared to that of a bus-based TAM. The proposed architecture provides a modular TAM that provides two levels of scalability, i.e. *design-version scalability* and *multi-level scalability*. It is also shown that the proposed TAM, unlike a bus-based architecture, can function without any restriction on the number of primary input/output pins. Furthermore, it is estimated that the total wire-length in our architecture will be less than one percent of the total wire-length of the chip, compared to more than one hundred percent in the case of a bus-based TAM architecture.

1 INTRODUCTION

1.1 Background

Using the large number of transistors available on a chip, designers have already managed to put entire systems on a single chip, referred to as *Systems on Chip* (SoC) or *System Chips*. As the number of these transistors increases, approaching 4000 million by the year 2014 [1], more complex systems, utilizing hundreds of embedded *cores*, will be placed on a single chip. Apart from the availability of large transistor counts, there are many reasons for the increasing use of system chips: SoCs are suitable for portable low-weight products, they use less power per function, and they cost less when compared to systems on a board [8].

However, due to the *productivity gap*, it is becoming harder to utilize all the transistors available on the chip effectively and keep the time to the market of a design to a minimum [1]. To bridge the *productivity gap*, the only viable solution put forward so far has been for designers to *re-use* their pre-designed/verified *Intellectual Properties* (IP), or those of others, as embedded *cores* [2,3].

IP-based design, however, is creating a new style of design with its own characteristics, requirements,

benefits, and challenges. One particular challenge is how to test these *core-based* systems. Experts believe that the idea of *re-use* must be applied in testing SoCs such that the *core-provider* be responsible for the development of the cores' test strategies and providing all the test knowledge to the *core-user*. The *core-user* is the party that designs a system's DFT scheme, where individual core test schemes are integrated, and hence, enabling the integrated cores to be tested [4-7].

One important challenge in testing SoCs is accessing the embedded cores from the chip boundary. Challenges associated with this *test accessing mechanism* include:

- Using the limited number of primary I/O pins for test;
- Testing different types of cores with compatible methodologies;
- Testing the User Defined Logic (UDL) and interconnects;
- Testing at-speed;
- Integrating different core test schemes into a system DFT;
- Limiting test time, power, area overhead, etc.;
- Having an access mechanism allowing new cores to be integrated into the chip without incurring core design changes;
- Designing the system DFT such that the chip will be re-usable hierarchically.

Zorian et al. in [4] propose a generic conceptual test access architecture for embedded cores, the components of which are: *Source*, *Sink*, and *Test Access Mechanism* (TAM). The TAM is the physical mechanism that connects the source and the sink with the core. It also refers to the controlling signals needed for this connection.

1.2 Prior Work

There are many proposed TAM architectures in the literature. These can all be grouped into three main categories: multiplexing, serial connection, and bus-based connection.

1.2.1 Multiplexing

In this category, multiplexing is used to access the cores. The simplest method in this category multiplexes the test pins to the primary inputs/outputs (I/O) such that a direct path is established during test [12]. A second method modifies the cores such that each core has a *transparent* mode for testing [9,13]. A third method in this category suggests using the available functional communication medium on the cores, such as the processor bus [14]. A recent fourth method tries to find a systematic methodology by providing *transparent* paths and modeling the TAM as an Integer Linear Programming (ILP) problem to minimize the overall test time and overhead area [15].

There are serious shortcomings in the above mentioned TAM methods, such as limited scope of use for future complex SoCs, severe overhead area, long test time, possible interference with functions of the cores, and non-scalability of the architecture.

1.2.2 Serial Connection

TAMs in this category use the established IEEE 1149.1 standard [11,16-18]. Whetsel, in [19], uses a hierarchical structure by introducing a *Tap Link Module* (TLM). An improvement on the TLM is presented in [20], where the *Test Access Protocol* (TAP) is kept unchanged from its original form, and hence, simpler TLM controls are designed.

For a few cores on an SoC, it may be possible to spend time transporting the test vectors serially to the cores. However, as the number and complexity of the cores increases, a serial solution based on the IEEE 1149.1 standard or its variants will prove too costly in terms of test time.

1.2.3 Bus-based Connection

A number of different variations of the bus-based connection schemes have been reported. Central [6] or distributed [21] controls have been suggested for controlling the required switches.

Varma et al. [6] suggest a structured architecture based on separate data and control buses. In their work, provision has also been made for using several such buses with different width. To simplify the control mechanism in the TAM architecture and provide scalability of the architecture through hierarchy, a multilevel bus structure connected in a tree topology has been suggested [10].

Marinissen et al. have suggested a topology where cores are connected in a rail configuration and buses can have different width, fan-in, and fan-out [7]. Each core can be bypassed if needed to access the next one in line and control is achieved by a serial connection. *Core Access Switches* (CAS), which are connected similarly to the rail configuration, have also been suggested to select P signals out of N bits of a bus [22].

Whetsel has suggested an addressable architecture in [23]. In this architecture each core is given an addressable *Test Port* (TP), which can serially be assigned with its appropriate address, to provide an intelligent distributed control mechanism.

In terms of trading-off increased overhead area for reduced test access time, bus-based architectures are the most efficient TAM schemes suggested to date. However, the conjecture here is that, due to a number of problems, bus-based strategies are not viable solutions for future system chips, since they do not provide sufficient scalability and configurability. In favor of this conjecture and from the standpoint of the functional interconnects for SoCs, a few problems of bus connections have been identified in [24] and a new generic interconnect template is proposed in that work.

The next section identifies more problems in bus-based TAM architectures apparent in terms of test access time and interconnects length. To overcome these problems and provide a more effective solution for TAM, a *Novel Indirect and Modular Architecture* (NIMA) for TAM is proposed in this work, where emphasis is placed on modularity, generality, and configurability of the architecture to exploit the advantages offered by the re-use paradigm.

2 THE NIMA TAM

2.1 Concept

The suggested TAM schemes in the literature all try to establish a *direct* path between the source, the cores, and the sink in one way or another. Instead, the key idea in our work is to establish an *indirect* digital communication path through packet switching connections, analogous to the technology used for the Internet. The proposed idea is illustrated in Figure 1, where on- and off-chip sources and sinks are shown and an extra unit denotes the limit on the chip primary I/O pins that emphasises the fact that all the communications need to pass through this limited channel. If the on-chip sources/sinks are not physically close to their appropriate cores for direct connection,

then they can communicate through the packet switch network as shown.

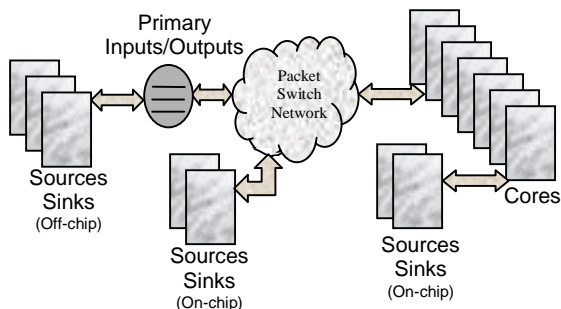


Figure 1: Conceptual TAM based on a Packet Switching Communication Network (NIMA).

Our preliminary investigations have established the feasibility of a TAM based on a packet switching digital communication network where test vectors are converted to packets before they *find* their way to the core they are intended for. Test results can also be transferred in the same way from the core to the sink. The following sections describe the concept in more detail.

2.2 Architectural Design

To simplify, manage, and provide modularity in a communication subsystem, the International Standards Organization (ISO) has adopted a 7-layered model where the entire communication subsystem is broken down into a number of well-defined layers. Using their example, NIMA TAM is designed such that the communication between source/sink and cores is achieved in a 3-layer hierarchical model as illustrated in Figure 2. A 3-layer model can provide the modularity required without complicating the architecture unnecessarily. Each layer, performing a well-defined function in the context of the TAM, has a pre-defined protocol and a well-defined interface with the other layer(s), and deals directly with its own *peer* through a *virtual link*. The physical connection, however, is only in the vertical direction of the model except in the *Physical Layer*, where there is no *virtual link* but only a physical link.

The *Application Layer* encompasses the proposed IEEE P1500 standard core wrapper [27], test schedulers, and test programs that can be based on the *Core Test Language* (CTL) currently being developed by the IEEE P1500 standard working group. The *Network Layer* encodes/decodes the test data into/from the *packets* and transports these packets to their destinations. The *Physical Layer* is the electrical

connection and uses the voltages used for representing digital values of “1” and “0” in ICs.

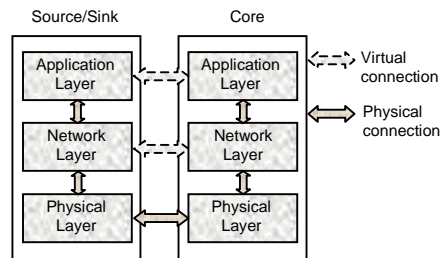


Figure 2: The Layer Model of NIMA.

NIMA enables the management of different *functional* and *structural* test schemes as well as different core types. Since the communication in the *Network Layer* is digital, if the core to be accessed is analogue, the test vectors and responses need to be converted into a digital format at the boundary of the *Application/Network* layer and converted back into an analogue format at the boundary of the *Network/Application* layer. If this is not acceptable for testing a core, another purely analogue TAM can be used for that core alongside NIMA TAM.

In the following, the *Network Layer* design is explained in detail. Work is underway to elaborate the *Application Layer* of the architecture and the results will be presented in the future publications.

2.3 Network Layer

The *Network Layer* is based on a *virtual indirect* connection between a source/sink and a core. Packets are sent in order and assumed to reach their destinations in order. This simplifies the *Application Layers* at the cores and thus, reduces the overhead hardware.

A packet definition is shown in Figure 3, where the *Sync Word* signals the beginning of a packet, the *Address Length* defines the length of the *Address* field to follow, and the *Address* field is a variable length of address bits. Finally, the *Data* field holds a fixed number of data bits. The sizes of these fields, in terms of bits, are: *S*, *L*, and *D* as shown in Figure 3. For the case of the *Address* field, *A* represents the number of *locations* in this field where each location contains *n* bits. The architecture is designed such that packets can succeed each other without any gaps.

<i>S</i>	<i>L</i>	<i>A</i>	<i>D</i>
Sync Word	Address Length	Address	Data

Figure 3: Packet Structure.

2.3.1 Addressing

Using the packet structure of Figure 3, the *Logical Address Space* (LAS) is equal to:

$$LAS = 2^{nA}$$

Since addressing is achieved through the two fields, *Address Length* and *Address*, where the former is used as a pointer to the length of the latter, then an n -dimensional LAS is provided where:

$$A \leq 2^L$$

and hence, the maximum LAS is:

$$LAS = 2^{n \times 2^L}$$

If values of n and L are chosen appropriately, the LAS can be very large and hence, enabling the system designer to use *partial addressing* for faster and more effective testing time.

Using variable addressing, the entire LAS is essentially divided into 2^L hierarchical levels each with 2^n pages, where n is the *dimension* of each location in the *Address* field pointed to by the *Address Length*. Since the length of the *Address* field, A , can be indicated by the *Address Length*, a core with large and lengthy test vectors can be assigned an address in the first level of the hierarchy instead of the last level, i.e., 2^L level if the entire LAS is used. This is effectively using a *partial addressing* strategy that reduces the access time for a core at the expense of not being able to use the *pages* in the lower levels. Variable addressing and the advantages it offers are discussed further in Section 2.3.3.

2.3.2 Routers

Using the variable addressing strategy and appropriately designing the routers results in minimal data storage requirements for routers. In fact, for n -dimensional addressing explained in Section 2.3.1, each router is only required to save $L_r = S + L + n$ bits, i.e., the *Sync Word*, the *Address Length*, and the first n bits of the *Address*.

As illustrated in Figure 4 for a 2-dimensional 3-layer addressing example, the routers can be arranged into an H-tree topology to reduce the wire-length. Moreover, to simplify the routers, the architecture is designed such that the routing decisions are hard-wired into the routers. Hence, it is assumed that the routing decision is taken by the system designer in advance and is static during packet transfer.

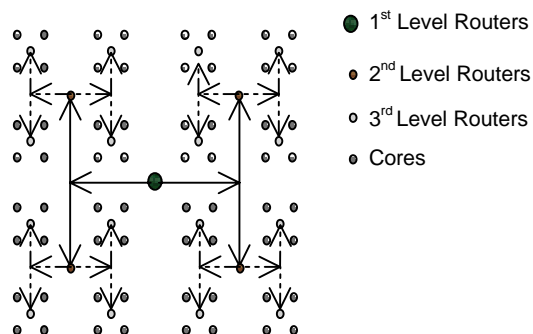


Figure 4: Router Topology.

In conjunction with the variable addressing scheme, the proposed H-tree router architecture provides a fully modular and scalable architecture for the TAM, where cores using new addresses can be added without changing the previous design. In addition, each first-level router can be connected to a primary I/O pin of the system chip. This effectively divides the entire chip space into subsections that can be individually addressed and accessed. Hence, the *Effective Logical Addressing Space*, LAS_{Eff} , for an n -dimensional architecture can be written as:

$$LAS_{Eff} = m \times 2^{n \times 2^L}$$

where n is the dimension of the *Address* field and m the number of chip primary I/Os. In this way, cores can be accessed even faster and in parallel if needed. In NIMA, it is also possible to achieve parallel accessing of the cores even if only one primary I/O pin is available for testing. This task can be achieved by using the *Application Layer* to schedule the tests concurrently.

2.3.3 Variable Addressing Length

Variable addressing provides many further advantages. A first advantage is the modularity and scalability of the architecture. A variable addressing-length is used in the architecture of the *Network Layer*. Hence, the length of the address, A , can be chosen according to the system requirements, i.e., total cores to be accessed, test time, and area overhead, and it is not constant. The *Physical Address Space* (PAS), defined as the part of the LAS being used, can vary in size, and can be chosen to be the minimum size required. As new cores are added to the SoC, if the present PAS has no free addresses, new levels of addresses can be introduced to increase the size of the PAS, and hence, accommodate new cores with no design modification. This results in having the advantage of a scalable TAM architecture between SoC

design versions, referred to here as *design-version scalability*. Moreover, when a first SoC is used as an embedded core in a second SoC, the PAS of the first SoC can be assigned to the lowest part of the second SoC's LAS. The PAS of the second SoC can then continue from the next available hierarchical level of the 2^n pages. This creates another level of scalability, *multi-level scalability*. As before and using the *design-version scalability* feature, new cores in the second SoC can be added to the new PAS.

Design-version scalability and *multi-level scalability* are illustrated conceptually in Figure 5.

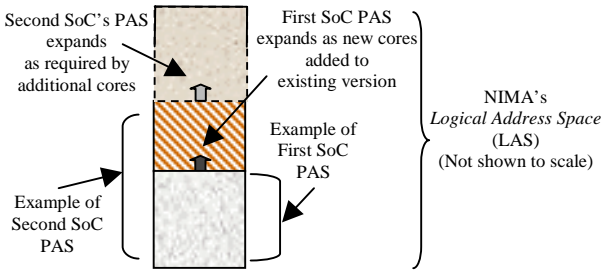


Figure 5: Address Spaces for the Proposed Network Layer.

A second advantage of variable addressing used in NIMA is to limit the latency introduced by each router to $L_r = S + L + n$ clock cycles compared to $L_f = A$ in the case of fixed-addressing. Choosing appropriate values for the parameters of the packet structure, i.e., S , L , and n , such that LAS is large enough to use partial addressing, and assuming a practical addressing range for a fixed-addressing example that can provide enough addressing space for more than thirty thousand cores, values greater than two for the ratio of L_f/L_r can be achieved [28].

A third advantage is the reduction in core access latency, L_a . In an example of a fixed-addressing scheme, $L_d = A$ hops are needed for each packet to get to its destination, where A needs to be greater than fifteen bits for an architecture that can provide enough addressing space for more than thirty thousand cores. NIMA enables the test source to access cores with:

$$L_a = O\left(\frac{\lceil \log_2 N \rceil}{n}\right)$$

hops, where N is the number of cores, n is the dimension used in the *Address* field, and $\lceil \cdot \rceil$ denotes the ceiling function. If the entire LAS is used, then it takes at most 2^L hops for packets to reach their destinations.

However, this is an unlikely address space and a smaller PAS is used limiting these number of *hops* to the order of five to ten, and hence, resulting in an improvement of $L_d/L_a \geq 1.5$. Moreover, NIMA can be configured such that at most one *hop* be necessary to access cores with the largest test vector sets [28]. Doing so greatly reduces the core access latency L_a .

A fourth advantage of variable addressing in NIMA is the gain in the time to *feed* in the packets that translates into test time required for each core. A fixed-addressing scheme requires $data_f = A + D$ bits for each packet whereas NIMA requires $data_p = S + L + A + D$ bits. However, A in this latter case is variable and can be as small as n for a core assigned an address in the first level of the hierarchy. Choosing practical values for the parameters of the packet structure as mentioned before, and assuming $A > 15$ in the case of a practical fixed-addressing example, values greater than three for the ratio of $data_f / data_p$ can be achieved, which amounts to significant saving in the total test time [28].

A final advantage is that the used address bits in each router can be taken out of the packet and discarded. This progressively shortens the length of the packets, and hence, the introduced latency, as packets move in the network.

3 MODELLING AND SIMULATIONS

3.1 Core Access Time

To study the core access time requirements of NIMA TAM, we developed a model for the access time to the first level router. This access time can represent the core access time provided the packets are pipelined. In order to compare the access time in our packet switching TAM to that of a bus-based TAM, the bus architecture as shown in Figure 6 is used and its corresponding core access time modeled. These models have then been simulated in Matlab.

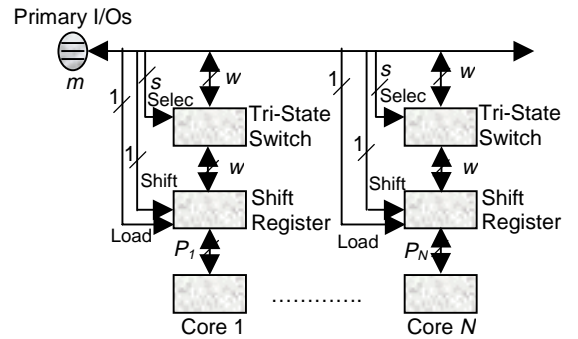


Figure 6: Bus Architecture Model.

In Figure 6, N denotes the total number of cores, P_i is the number of test pins for core i , w is the bus data width, s is the number of signals needed to address the tri-state switches, and m is the total number of primary I/O pins.

Detailed accounts of the models development are given in [28]. For NIMA TAM, it is shown in the latter that:

$$T_p = \left(\frac{P}{D \times m} + \frac{\sum_{i=1}^N R0_i}{m} + \sum_{i=1}^N R1_i \right) (S + L + A + D) \quad (1)$$

where the following conditions apply:

$$\begin{aligned} 0 &\leq R0_i < 1 \\ 0 &\leq R1_i < 1 \\ n &\leq A \leq \left\lceil \frac{\log N}{\log 2} \right\rceil \end{aligned}$$

Also, for the bus-based TAM the following is shown [28]:

$$T_b \geq \frac{P}{w} + \sum_{i=1}^N R_i \quad (2)$$

where the following conditions must satisfy:

$$\begin{aligned} 0 &\leq R_i < 1 \\ w &\leq m - 2 - s \end{aligned}$$

In Equations (1) and (2), T_p denotes the first-router access time for NIMA and T_b is the core access time for the bus architecture of Figure 6. In these models, $\lceil \cdot \rceil$ denote the ceiling function, N is the total number of cores, P denotes the total number of cores' test pins, w denotes the bus data width, s is the number of select signals in Figure 6, m represents the number of primary I/O pins, S , L , A , and D denote the field length of the packet in Figure 3, and n denotes the dimension used in the *Address* field. Finally, R_i , $R0_i$, and $R1_i$ are residual values to convert the addition of ceiling functions to normal addition over real numbers [28].

Figure 7 shows the first-router access time for the packet switching TAM where $N=1000$, $D=100$, $S=L=6$, and $R0_i=R1_i=0$. To find the worst values for the access time, the maximum value of A in the conditions of Equation (1) is used for this plot.

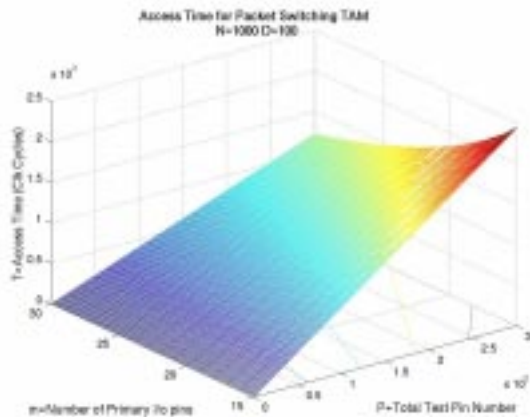


Figure 7: First-router Access Time for NIMA.

Figure 8 shows the minimum core access time for the bus-based TAM where $N=1000$ and $R_i=0$. In addition, the minimum value of s and the maximum value of w have been used for this plot.

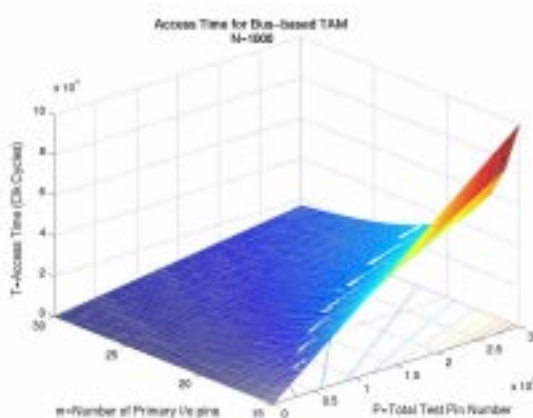


Figure 8: Core Access Time for Bus-based TAM.

From the plots in Figures 7 and 8, relative to the bus-based TAM, the packet switching TAM has comparable access times when more than about thirty primary I/O pins are available for testing. However, packet switching outperforms the bus-based TAM for fewer I/O pins. Moreover, the bus-based TAM shows a much sharper increase in access time as the number of cores or their test pin counts increase. In fact, as the number of primary I/O pins falls below a certain value, about twenty for the case shown in Figure 8, the rate of increase in access time in the bus-based architecture increases rapidly. This can be a major source of problems in bus-based TAM in future SoCs where the number of the cores and their test pins are expected to be larger than what we used in the example in Figure 8. To compensate, many more primary I/Os would be

required to keep the access time at a reasonable level. From conditions of Equation (2), the bus-based TAM requires a minimum number of primary I/O pins to provide for the bus data and control bits. NIMA TAM, on the other hand, can function without any restriction on the number of primary I/Os. This is a very attractive feature. It enables the system designer to have full control over the configurability of the TAM according to the system requirements, and still use the packet switching TAM for a variety of conditions on the number of available primary I/O pins.

3.2 Interconnect Length

The total wire-length of any TAM architecture is an important metric. Wire-lengths greater than 10mm in 100 nm technology and beyond have dramatic and those with length of more than 2 mm have significant effects on signals delay and integrity [25]. To overcome these effects, many buffers are required for long wires, increasing total area. In this section, we develop a model to estimate the total wire-length of NIMA. In addition, to enable comparison, we also develop a model for the wire-length of a bus-based TAM.

3.2.1 Wire-Length Model for NIMA TAM

Based on the proposed topology of routers shown in Figure (4), the total interconnect length for the packet switching architecture, assuming the first router is placed in the center of the chip, can be bounded as follows [28]:

$$WL_{ps} \leq \frac{19L}{4} \quad (3)$$

To model the total length of NIMA, we assume that all cores have the same area and that there is no space between them. It is also assumed that the length and width of the chip is equal to L and that the total number of cores is N .

Given that gate pitch is 40λ , where λ is the half-length of a technology, and assuming the number of gates in a core to be N_g , then Equation (1) can be re-written as:

$$WL_{ps} \leq 190\lambda\sqrt{N}\sqrt{N_g} \quad (4)$$

The important trend emerging from these bounds is that the TAM length is fully independent of the number of primary I/O pins used for parallel accessing. This holds when neglecting the wire length to the first router of each group of routers connected to the primary I/O pins.

3.2.2 Wire-Length Model for Bus-based TAM

An H-tree connection topology, as illustrated in Figure 9, provides the shortest wire-length in a bus scheme.

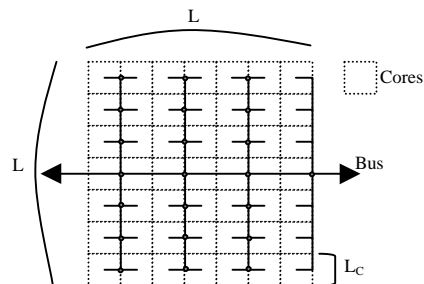


Figure 9: H-Tree Topology for Bus Connection.

To model the total length of a bus-based TAM architecture, we assume that all cores have the same area, their pitch all equal to L_c , and that there is no space between them. It is also assumed that the length and width of the chip is equal to L , that the total number of cores is N , and that there are the same numbers of rows as columns of cores in the chip. Using w for the bus data width, the bus wire-length is:

$$WL_{bus} = w \left(L\sqrt{N} + \frac{L}{2} \right) \quad (5)$$

Knowing

$$L_c = 40\lambda\sqrt{N_g} \quad (6)$$

and assuming $n \gg 1$, the total bus wire-length, in terms of number of cores, can be derived from Equation (5) to yield:

$$WL_{bus} \approx 40Nw\lambda\sqrt{N_g} \quad (7)$$

3.2.3 Discussion

Comparing Equations (3) and (5) reveals that the packet switching architecture offers an improvement in the wire-length by a factor

$$\eta = \frac{WL_{bus}}{WL_{ps}} = \frac{4w}{19} (\sqrt{N} + 0.5)$$

This is clearly a very large improvement directly proportional to the bus data width and to the square root of the number of cores. For a typical value of $w=20$ and for the number of cores from 1 to 10,000, η ranges from

about 6 to 423. Even for a conservative value of $w=5$ and the same range of core counts, η ranges from 1.5 to 105.

Davis et al. in [26] presented a model for the total length of wires, used for functional interconnections, in integrated circuits:

$$L_{\text{Wires}} = \frac{1}{p} - \frac{\sqrt{N_G}}{p-0.5} - \frac{1}{6\sqrt{N_G}(p+0.5)} + N_G^p \left(\frac{-p-1+4^{p-0.5}}{2(p+0.5)(p-0.5)p(p-1)} \right)$$

where p is the Rent's Rule empirical coefficient, N_G is the total gates number, and the length is in units of gate pitch. Figure 10 shows plots of the total functional wire-length, converted to meters, for a range of values of λ from 90 to 15 nm and $10^6 \leq N_G \leq 10^9$.

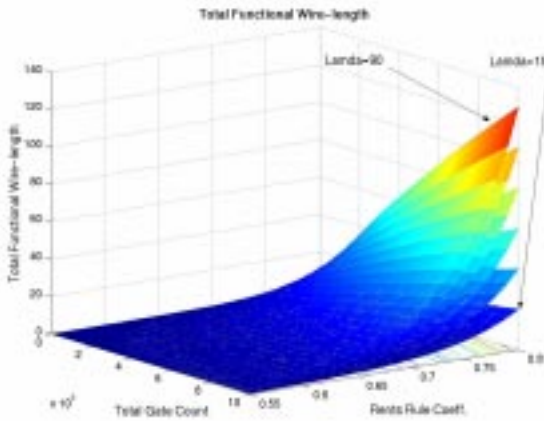


Figure 10: Total Functional Wire-Length [26].

From the plots in Figure 10, $L_{\text{Wires}} \cong 40m$ in $0.18 \mu m$ technology with fifty million gates on the chip. This length decreases to about twenty meters for future chips using the 30 nm technology and containing one-thousand million gates.

From Equation (3), the total wire-length of NIMA TAM is 0.25 meters for a chip containing 10,000 cores and of size 30 mm on a side. From Equation (5), the wire-length for a bus-based TAM architecture can be estimated to be about thirty meters for a 10-bit wide bus. This calculation is based on the same core number and chip length.

The total wire-length of NIMA is hence, about one percent of the total expected functional wire-length in future chips, and this ratio is independent of the number of primary I/O pins used. This is a very attractive feature of our proposed architecture in that it can keep the total wire-length to acceptable levels as well as

providing a parallel access to cores according to system requirements.

On the other hand, the wire-length found here for a bus-based TAM suggests that bus-based architectures fall well short of being appropriate for future chips, as they add more than one hundred percent to the wire overhead in future chips. The other problem is the dependency of their wire-length on the number of the primary I/Os used. As more of these I/Os are used to shorten the access time, the total wire-length increases to unacceptable levels.

4 CIRCUIT IMPLEMENTATION

To date, the proposed router has been designed and simulated at the RTL level, where the number of gates amounts to 5400. In addition, a test bench, comprised of five simple synchronous digital blocks acting as *cores* and arranged into four levels, has been designed to test NIMA. Using NIMA TAM in this test-bench, test vectors have been successfully transferred and applied to the *cores* holding the TAM's functionality.

From simulation, the time, in clock cycles, from the moment the first bit of the data is *fed* into the first router to the time this bit is available at the output of the last router, is:

$$L_t = (S + L)(d - 1) + 1$$

where d is the total number of the routing stages. For most systems, as explained in Section 2.3.3, the TAM can be designed to have five to ten levels of routers. This yields a small value of L_t in the range forty-nine to one hundred and nine for an example of the architecture with $S=L=6$.

5 CONCLUSIONS

As the number of cores used on a system chip increases, the classical TAM architectures tend to present more problems in terms of limited configurability and scalability, and also in terms of access time and interconnect length. Here, we proposed an architecture based on a packet switching communication network, called NIMA, and described its *Network Layer* architecture. Core access time and interconnect length models for our proposed TAM as well as for a bus-based TAM architecture have been developed and simulation results presented.

It is shown that NIMA, due to its modular structure, provides two levels of scalability, i.e., *design-version scalability* and *multi-level scalability*. In addition,

NIMA, using a clever variable addressing scheme, provides a configurable TAM scheme easily adaptable to the requirements of the system design.

Moreover, the simulation results show that NIMA, while providing better access time, adds less than one percent of the functional wire-length to the wire area of future SoCs, and that this added area is fully independent of the number of primary I/O used. This is in contrast to the bus-based scheme that increases this area by more than one hundred percent for a bus data width of ten bits, and even more as the number of the bus data width increases.

The results given support the feasibility of our proposed TAM architecture. In addition, an RTL level of the *Network Layer* has been developed and simulated. Work is underway to elaborate the *Application Layer* of the architecture and constructing a system DFT based on the proposed architecture. In addition, NIMA is being implemented on a realistic test bench to assess its performance and potential shortcomings. Concurrently, to provide a systematic tool for designing the TAM in future SoCs, investigations are underway into the development of a formal cost model for TAM architectures.

6 REFERENCES

- [1] The International Technology Roadmap for Semiconductors, 1999 Edition, <http://public.itrs.net/>.
- [2] E.J. Marinissen, Y. Zorian, "Challenges in Testing Core-Based System ICs", *IEEE Communication Magazine*, June 1999, pp. 104–109.
- [3] R.K. Gupta, Y. Zorian, "Introducing Core-Based System Design", *IEEE Design & Test of Computers*, December 1997 14(4), pp. 15-25.
- [4] Y. Zorian, E.J. Marinissen, S. Dey, "Testing Embedded-Core Based System Chips", *Proc. IEEE ITC*, 1998, pp. 130-143.
- [5] R. Chandramouli, S. Pateras, "Testing Systems on a Chip", *IEEE Spectrum*, Nov. 1996, pp. 42-47.
- [6] P. Varma, S. Bhatia, "A Structured Test Re-Use Methodology for Core-Based System Chips", *Proc. IEEE ITC*, 1998, pp. 294-302.
- [7] E.J. Marinissen, R. Arendsen, G. Bos et al., "A structured & Scalable Mechanism for Test Access to Embedded Reusable Cores", *Proc. IEEE ITC*, 1998, pp. 284-293.
- [8] E.J. Marinissen, Y. Zorian, R. Kapur, T. Taylor, L. Whetsel, "Towards a Standard for Embedded Core Test: An Example", *Proc. IEEE ITC*, 1999, pp. 616-627.
- [9] I. Ghosh, S. Dey, N.K. Jha, "A Fast & Low Cost Testing Technique for Core-Based System-on-Chip", *Proc. IEEE DAC*, 1998, pp. 542-547.
- [10] A. Benso et al., "HD2BIST: Architectural Framework for BIST Scheduling, Data Patterns delivering & Diagnosis in SoCs", *Proc. IEEE ITC*, 2000, pp. 892–901.
- [11] L. Whetsel, "Core Test Connectivity Communication & Control", *Proc. IEEE ITC*, 1998, pp. 303-312.
- [12] V. Immaneni, S. Raman, "Direct Access Test Scheme-Design of Block and Core Cells for Embedded ASICS", *Proc. IEEE ITC*, 1990, pp. 488-492.
- [13] I. Ghosh, N.K. Jha, S. Dey, "A Low Overhead Design for Testability & Test Generation Technique for Core-Based Systems", *Proc. IEEE ITC*, 1999, pp. 50-59.
- [14] B. Mathewson, "Core Provider's Test Experience", *Presentation at IEEE P1500 Working Group Meeting*, June 1998, CA. U.S.A., <http://grouper.ieee.org/groups/1500/pastmeetings.html#dac98>.
- [15] M. Nourani, C. Papachristou, "An ILP Formulation to Optimize Test Access Mechanism in System-on-Chip Testing", *Proc. IEEE ITC*, 2000, pp. 902-1000.
- [16] N.A. Touba, B. Pouya, "Testing Embedded Cores using Partial Isolation Rings", *Proc. IEEE VLSI Test Symposium*, 1997, pp. 10-16.
- [17] B. Dervisoglu, J. Swamy, "A Novel Approach for Designing a Hierarchical Test Access Controller for Embedded Core Designs in an SoC Environment", *IEEE Workshop in Testing Embedded Core-Based System-Chips*, 2000, pp. 1.4.1-1.4.7.
- [18] B. Pouya, N.A. Touba, "Modifying User-Defined Logic for Test Access to Embedded Cores", *Proc. IEEE ITC*, 1997, pp. 60-68.
- [19] L. Whetsel, "An IEEE 1149.1 Based Test Access Architecture for ICs with Embedded Cores", *Proc. IEEE ITC*, 1997, pp. 69-78.

[20] D. Bhattacharya, "Hierarchical Test Access Architecture for Embedded Cores in an Integrated Circuit", *Proc. VLSI Test Symposium*, 1998, pp. 8-14.

[21] T.J. Chakraborty, S. Bhawmik, C-H. Chiang, "Test Access Methodology for System-on-Chip Testing", *IEEE Workshop in Testing Embedded Core-Based System-Chips*, 2000, pp. 1.1.1-1.1.7.

[22] M. Benabdenbi, W. Maroufi, "CAS-Bus: A Scalable and Reconfigurable Test Access Mechanism for Systems on a Chip", *Proc. IEEE DATE*, 2000, pp. 141-145.

[23] L. Whetsel, "Addressable Test Ports, an Approach to Testing Embedded Cores", *Proc. IEEE ITC*, 1999, pp. 1055-1064.

[24] P. Guerrier, A. Greiner, "A Generic Architecture for On-Chip Packet Switched Interconnections", *Proc. IEEE DATE*, 2000, pp. 250-256.

[25] P. Nordholz et al., "Signal Integrity Problems in Deep Submicron Arising From Interconnects between Cores", *Proc. IEEE VLSI Test Symposium*, 1998, pp. 28-33.

[26] J.A. Davis, V.K. De, J.D. Meindl, "A Priori Wiring Estimations and Optimal Multilevel Wiring Networks for Portable ULSI Systems", *Proc. IEEE Electronic Components and Technology Conference*, 1996, pp. 1002-1008.

[27] IEEE P1500 Standard for Embedded Core Test (SECT), <http://grouper.ieee.org/groups/1500>.

[28] M. Nahvi, "Test Access Mechanism in System Chips using a Packet Switching Communication Network Methodology", *Ph.D. Thesis in development, Department of Electrical and Computer Engineering, The University of British Columbia*, 2001.