Early Estimation of Wire Length for Dedicated Test Access Mechanisms in Networks-on-Chip based SoCs

Alexandre M. Amory,
Fernando G. Moraes
Faculdade de Informática,
PUCRS Catholic University,
Porto Alegre, Brazil
alexandre.amory@pucrs.br,
fernando.moraes@pucrs.br

Cristiano Lazzari
INESC-ID,
Lisbon, Portugal
lazzari@inesc-id.pt

Marcelo S. Lubaszewski
PGMICRO-UFRGS University;
CEITEC S.A.;
Porto Alegre, Brazil
luba@ece.ufrgs.br,
marcelo.lubaszewski@ceitec-sa.com

ABSTRACT

The use of existing Networks-on-Chip (NoCs) for test data transportation has been proposed to avoid conventional dedicated Test Access Mechanism (TAM), improving the modularity of the test architecture. This paper presents a wire length estimation method used to evaluate the cost of dedicated TAMs for NoC-based SoCs early in the design flow. This wire length information (together with test time, power dissipation, among other test metrics) can help the designer to decide the best test architecture (NoC TAM or dedicated TAM) for a given chip. The experimental results demonstrate that dedicated TAMs require, on average, 26% of the global wires, enforcing quantitatively the benefits of NoC TAMs. On the other hand, results can vary depending on the SoC, from 3% to 70%, demonstrating the need of a fast wire length estimation in early stages of design.

Categories and Subject Descriptors

B.7.3 [**Testability**]: Reliability and Testing.

General Terms

Reliability

Keywords

VLSI test, SOC test, networks-on-chip, wire length.

1. INTRODUCTION

With the scaling of microchip technology, computation is becoming cheaper than communication. The main reason is that *global wires* do not scale as transistors do. Global wires can be found in the chip-level communication infrastructures such as buses. These buses exist in both the test and the functional domains of the chip. In the functional domain, Networks-on-Chip (NoCs) [2] are replacing global buses due to scalability, modularity, and parallel communication features. NoCs alleviate the issues related to long global wires because NoCs consist of shared and segmented wires [1]; sharing wires reduces the number of global wires, while segmenting wires reduces their sizes.

In the test domain of the chip, the test architecture is typically based in a conceptual model [7] for modular testing that consists of: test wrappers, used to switch between functional and test modes; and Test Access Mechanisms (TAMs), used to transport test data from/to the test pins to/from the Core-Under-Test (CUT). The most common practice for TAM design is to include dedicated and global test buses used only for test data transportation. Since these TAMs consist of long global wires, dedicated test buses are also subject to the same scalability and modularity problems mentioned before. Considering these issues with global buses, Cota et al. [5] proposed the use of the NoC structure to transport test data, improving the modularity of the test architecture. Doing so, the same NoC would improve scalability and modularity of both functional and test domains.

Besides improving the modularity of the test architecture, NoC TAM can also avoid long global wires of dedicated TAMs, potentially reducing the silicon area. Previous approaches do not quantify the amount of wiring that could be actually saved by using NoCs as TAMs. The main *contribution of this paper* is to present a wire length estimation method used to early evaluate the amount of wiring required to implement dedicated TAMs in a NoC-based SoC. Thus, *the goal of this paper* is to get the test architecture generated by a given test scheduler for dedicated TAMs and to evaluate the resulting dedicated TAMs in terms of global wire length. This result may guide designers to select the most appropriate TAM architecture: dedicated or NoC TAM.

This paper is organized as follows. Section 2 presents the previous works. Section 3 introduces the proposed wire length estimation method and Section 4 evaluates the proposed method against layout information. Section 5 evaluates the wire length of dedicated TAMs for several SoCs. Section 6 presents the conclusion of the paper.

2. PREVIOUS WORK

As far as we know, this is the first paper to quantitatively evaluate the benefits of NoC TAMs in terms of amount of global test wiring. Prior works with closest motivation are related to test scheduling algorithm which optimizes both test length and TAM wire length [6, 8], but these papers

SBCCI'11, August 30-September 2, 2011, João Pessoa, Brazil.

do not actually give the amount of wires required to implement dedicated TAMs. Moreover, they require detailed and accurate layout information to perform the optimization. It means that the complete design layout is necessary to finally generate the test solution. On the other hand, the proposed wire length estimation model can also work with initial tile area and position estimates, thus, the test architecture optimization can be concluded earlier. Furthermore, NoCs are not considered in these previous papers.

Cota et al. [5] propose the reuse of the NoC as a test access mechanism to the cores in SoCs. Despite a significant reduction on the test time is presented, no information about the wire length savings and its impact on the layout is shown.

An interconnect wire length model for a specific test access mechanism called NIMA (Novel Indirect and Modular Architecture) is proposed in [10]. However, the wire length model assumes that all cores have the same area and there is no space between them. These assumptions migth not fit well to a NoC based SoC where the NoC logic is usually spread among (the NoC being a soft-core) the tiles and the tiles migth have different area.

3. WIRE LENGTH ESTIMATION MODEL FOR DEDICATED TAMS

The wire length model for homogeneous and heterogeneous NoC based SoCs is proposed in this section. This paper uses the term *homogeneous NoCs* for systems of tiles with the same silicon area and *heterogeneous NoCs* for systems of tiles with different area. Sections 3.1 and 3.2 present the approach for homogeneous and heterogeneous NoCs, respectively. Section 3.3 presents the problem statement for both homogeneous and heterogeneous NoC based SoCs. Section 3.5 discuss the limitations of the proposed model.

3.1 Proposed Model for Homogeneous NoCs

This model assumes that the SoC is represented by tiles which are evenly distributed in the entire SoC area such that the distance between any two neighbor tiles is the same. Each tile can have zero or more cores. The NoC routers are not inside the tile, but they are supposed to be next to their corresponding tile. The rest of the system (clock and reset tree, test wires, and NoC) are described as soft-cores, thus, distributed among the tiles.

The proposed method for homogeneous NoCs counts the minimal number of hops required to reach all modules within a dedicated TAM. The wire length between cores within the same tile and the wire length between the tile and its router are supposed to be zero. The wire length between cores in different tiles is equivalent to the number of hops between these two tiles. The number of hops is used as the relative wire length unit for homogeneous NoCs.

Fig. 1 illustrates the proposed wire length estimation method representing a NoC-based SoC with three dedicated TAMs, depicted as fat lines connecting the tiles. Each box represents a tile (identified by rXY) which consists of one NI connected to zero or more cores (identified by the number within parentheses). Let us assume that a conventional test scheduling algorithm generated the following TAM assignment (Fig. 1) for this SoC:

- TAM1= $\{c1, c5, c6, c8, c9\}$,
- TAM2= $\{c4, r01, r11, r12, r02\}$,

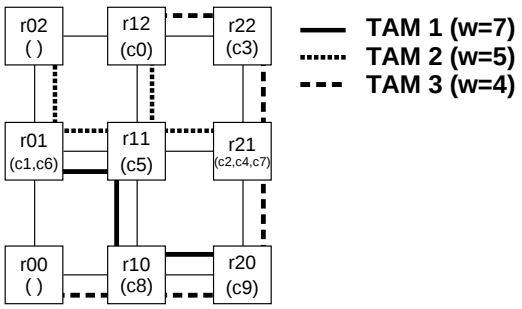


Figure 1: Example of a homogeneous NoC used to evaluate the wire length required to create the dedicated TAMs. *rXY* represents the router and *cZ* represents the cores connected to the router. *W* represents the TAM width in number of bits.

- TAM3= $\{c0, c2, c3, c7, r00, r10, r20, r21, r22\}$

assuming 16 test pins to connect the chip to the ATE. The width of these TAMs are 7, 5, 4 wires, respectively.

For instance, TAM1 (represented by the continuous fat line in Fig. 1) has five cores where two of them (c1 and c6) are located in the tile r01 and the remaining cores are located in tiles r11, r10, and r20. The proposed model estimated that the minimum distance between these four tiles is three hops. Since the width of TAM1 is seven test wires, then it results in $(3 + 1) \times 7 \times 2 = 56$ wires to implement the TAM1. The *plus one hop* represents the wires from the input test pins to the first core of the TAM while the $\times 2$ factor represents wires for both test stimuli and responses. The minimum number of hops for TAM2 and TAM3 are 4 and 5 hops, respectively, connecting the tiles r01, r11, r21, r02, and r12 (TAM2) and tiles r00, r10, r20, r21, r12, and r22 (TAM3). Finally, the total TAM wiring for this example is **154** wires ($(3 + 1) \times 7 \times 2 = 56$ for TAM1, $(4 + 1) \times 5 \times 2 = 50$ for TAM2, and $(5 + 1) \times 4 \times 2 = 48$ for TAM3).

3.2 Proposed Model for Heterogeneous NoCs

Now let us assume that the NoC is heterogeneous and each tile might have a different size, thus, they are not evenly distributed into the chip. In this case, *the method for heterogeneous NoCs requires an estimated area and position of these tiles.*

The area and position information required for heterogeneous NoCs can be taken from physical syntheses, however, it is not mandatory. The estimated tile area can also be taken from the logic synthesis while the estimated tile position can be taken from a bin-packing optimizer where the item size corresponds to the tile area. Using these estimated values might incur in some error, but it allows early estimation of test wire length specially at the first steps of the system design where there is no accurate layout information.

The proposed method for heterogeneous NoCs counts the minimal distance between each neighbor tile within a dedicated TAM. The wire length between cores within the same tile is also supposed to be zero, while the wire length between cores in different tiles is equivalent to the distance of these two tiles. Fig. 2 illustrated a TAM assignment generated by a test scheduler. For instance, TAM1, 2 and 3 contains the tiles r01, r11, r10, and r20; r11, r10, r20, and r21; r00, r01, r02, r12, r22, and r21, respectively.

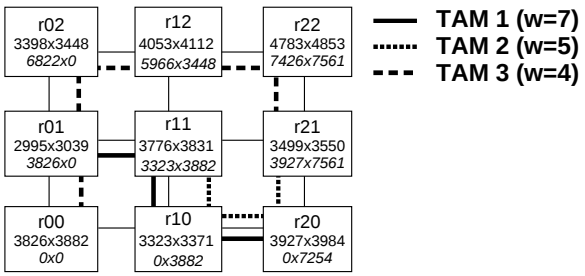


Figure 2: Example of a heterogeneous NoC used to evaluate the wire length required to create the dedicated TAMs. The second information in a tile is the tile area, represented by *Width*x*Height* (μm). The third information, in italics, is the tile position, represented by *X*x*Y* position. *W* represents the TAM width in number of bits.

The proposed method estimated that the sum of the distances of the tiles of TAM1 (i.e. the distance between r01 and r11; r11 and r10, and so on) is 11080 μm. Since the width of TAM1 is 7, then the total wire length for TAM1 is $11080 \times 7 = 77560 \mu m$. Similarly, the estimated length for TAM2 and TAM3 is, respectively, 54645 and $71528 \mu m$. As a result, the total SoC test wire length is the sum of the wire length of each TAM multiplied by two since there must be TAMs for test stimuli and responses. The total SoC test wire length is $2 \times (77560 + 54645 + 71528) =$**407466** μm.

3.3 Problem Statement

The problem of determining the minimal distance between the cores of a TAM (for both homogeneous and heterogeneous systems) is equivalent to the Minimum Rectilinear Steiner Tree problem which is $\mathcal{NP}$-complete [4, 11]. The problem can be modeled as follows: given N points in the plane, where a point represents a tile location, find a minimum length tree of rectilinear edges which connects the points. The description of this algorithm is out of the scope of this paper due to the paper size limit. However, note that it is a standard algorithm with several applications and implementations available. For more information about this algorithm, please refer to [4, 11].

While the problem seems to be simple to solve for a small number of points, as presented in the example in Fig.1 and Fig.2, it is a computationally intensive task. The fastest tools [4] can solve an instance of this problem with up to 80 points only. In the scope of this work it means that it supports systems with up to 80 tiles.

3.4 Estimating the Wire Length

The first step is to execute the Minimum Rectilinear Steiner Tree solver for a single TAM_i, where it determines the minimum wiring to connect all the tiles assigned to the TAM_i. In case of homogeneous NoC the wire length can be expressed in terms of number of hops required to connect all tiles while in case of heterogeneous NoC the wire length is necessarily expressed in terms of μm since the tiles have different area. Let h_i represent the resulting number of hops to connect all tiles of TAM_i (in case of homogeneous) or represent the resulting wire length to connect all tiles of TAM_i (in case of heterogeneous). The final step is to determine the total number of wires to implement the TAM, formalized as:

$$w_{SoC} = \sum_{i=1}^{n} (h_i \times w_i \times 2) \quad (1)$$

where w_i is the width of the TAM_i. Since there must be wires for both test stimuli and responses, then, the number of wires is multiplied by two. The process is repeated n times where n is the number of SoC TAMs. The overall SoC test wire length (w_{SoC}) is given by the sum of wire length for each TAM.

3.5 Limitations of the Approach

The actual TAM wire length in a chip also depends on the layout congestion. Congested layouts might require longer TAM wiring from one point to another than non-congested layouts. Since the proposed model does not capture layout congestion, the model represents the shortest wiring length required to design dedicated TAMs. The actual TAM wiring is expected to be longer than the estimated wire length.

Both models for homogeneous and heterogeneous systems assume that the tiles were initially created as a hard IP core, then these blocks are integrated at the system level with the NoC described as soft IP core, thus, the NoC logic is spread among the tiles. If the entire system is based on soft hardware descriptions, the approach may lead to a greater error.

4. PROPOSED APPROACH EVALUATION

4.1 Evaluation Set Up for Homogeneous NoC

Let us take the SoC presented in Fig. 1 as an example to compare the *actual* and the *estimated* wiring for dedicated TAMs. As a reference value, the amount of test wire is compared with the amount of wiring required to implement the NoC channels of 32-bits each.

As calculated in Section 3.1, the *estimated* number of test wires required to implement dedicated TAMs is 154. In an i-by-j mesh, there are $2 \times (i \times (j-1) + j \times (i-1))$ channels. For example, the system d695 is a 3-by-3 mesh, thus, it has 24 channels of 32 bits or $24 \times 32 =$ **768** wires[1]. Thus, according to the proposed model, close to $\frac{154}{768} =$ **20%** *of the global wires of the chip are required to implement the dedicated TAMs.*

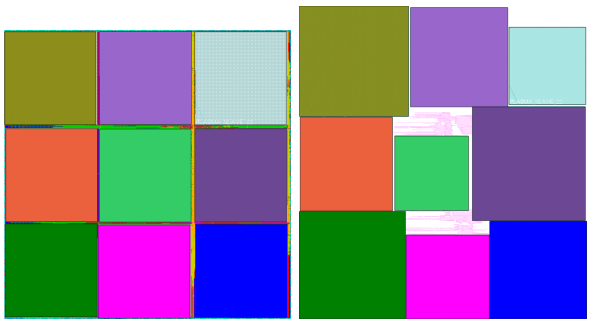
Layout is required to evaluate the *actual wiring* for dedicated TAMs. The system presented in Fig. 1(a) using dedicated TAMs is described in VHDL. First, a tile blackbox is created. It is based on the HeMPS [3] tile with a 32-bit MIPS processor, network interface, DMA, and 16KB dual-port memory resulting in a tile with $2560 \times 2550 \mu m$ of area. Second, the tile is connected to the HeMPS NoC (called Hermes), which is automatically generated by the HeMPS environment[2], with buffers of size 16 and 32-bit channel width. Third, the dedicated TAMs, depicted in Fig. 1, are included into the SoC. This entire system is finally synthesized.

4.2 Evaluation Set Up for Heterogeneous NoC

Fig. 1(b) shows the tile area and position used for the heterogeneous NoC. Similarly to the previous section, first we

[1]This wire count does not consider control wires used to implement the protocol.

[2]Available for download at `https://corfu.pucrs.br/redmine/projects/hemps`.



(a) Homogeneous NoC (b) Heterogeneous NoC

Figure 3: Layout of the d695 SoC with dedicated TAMs. The squares are the HeMPS tiles while the horizontal and vertical lines are the Hermes NoC with dedicated TAMs.

evaluate the amount of wires used to implement the NoC data channels by summing up their distances and multiplying it with the data width (32 bits), resulting in **1362880** μm. Recall that the proposed model estimated 407466 μm for test wire length, which correspond to $\frac{407466}{1362880}$ = **29.9%** compared to the total wire length required to implement the data channels of the entire NoC.

The tile hard IP cores were generated according to their area and location as illustrated in Fig. 1(b) and they were integrated to the same NoC presented before while the dedicated TAMs were adapted also according to TAM assignment illustrated in Fig. 1(b).

4.3 Wire Length Evaluation Results

After the SoC setup, Cadence^{TM} tools were used for logic and physical synthesis. The system has been synthesized to the library UMC 130nm. Fig. 3 illustrates the resulting layout of the d695[3] SoCs based on dedicated TAMs. The resulting wires are classified into *four classes of wires*: local, global, clock, and TAM wires. "Local wires" are required to implement the internal router logic. "Global wires" connect the routers to each other and the router to the tile, excluding TAM and clock wires. "Clock wires" implement the clock and the reset trees. Finally, "TAM wires" represent the dedicated TAMs. Table 1 shows the distribution of wire length among these types of wires.

Table 1: Distribution of wires in a NoC-based SoC with dedicated TAMs.

	local wire length (%)	global wire length (%)	clock wire length (%)	TAM wire length (%)
Homo. NoC	67.32	19.42	8.11	5.15
Hetero. NoC	73.18	16.32	5.04	5.46

The overhead of dedicated TAMs is small (5.15% and 5.46%) compared to the total wire length of the NoC. However, it consists of $\frac{5.15}{19.42}$ = 26.5% and $\frac{5.46}{16.32}$ = 33.45% of the global wires, respectively, for the homogeneous and heterogeneous case studies. Recall that the proposed model estimated that 20% and 29.9% of the global wires would be used to implement dedicated TAMs. The difference between the actual and estimated TAM wiring is due to routing congestion which is not captured in the proposed model.

[3]d695 is a benchmark SoC from ITC'02 SoC Test Benchmarks [9].

5. EXPERIMENTAL RESULTS

This section presents the results, most of them related to homogeneous SoCs since heterogeneous SoCs present similar results, as demonstrated next.

5.1 Experimental Setup

The first step to evaluate the proposed approach is to build the NoC-based SoCs used as case studies. The following systems from ITC'02 SoC Test Benchmarks [9] have been modified to include a NoC (the NoC size, i.e. the number of routers, for each system is in parentheses): d281 (3,3), d695 (3,3), g1023 (4,3), p22810 (5,5), p34392 (4,4), p93791 (6,5). The size of the NoC has been selected based on the number of cores of the system. There is also the so called 'big(9,9)' SoC which has been created to test the scalability of the proposed model. This SoC is placed in a 9×9 mesh with 117 cores of the five biggest ITC'02 SoC Test Benchmarks.

The ITC'02 SoC Test Benchmarks were modified to include the NoC into the SoC. First, each core receives two OCP (Open Core Protocol) ports to be able to receive test stimuli and to send test responses at the same time. Second, the routers and the NoC are generated. For the sake of simplicity we assume that all routers in a system are identical, i.e. they have five bi-directional ports and the same number of test patterns. Third, the cores of each SoC are placed on the NoC. Ten random placements have been generated for each SoC because the placement has an impact on the TAM wire length. The cores have been placed in the NoC randomly such that if the number of cores is greater than the number of routers, it makes sure that all routers have at least one core and no router receives more than two cores. It also makes sure that all placements are different from each other. At this point the SoCs have been generated. Next, the TR-Architect [7] algorithm is used for SoC test scheduling based on dedicated TAMs.

5.2 Wire Length Savings

This section uses the model presented in Section 3 to evaluate the amount of wiring spent in systems based on dedicated TAMs. In other words, *it evaluates the amount of wiring that could be saved by using NoC TAM.*

First, we calculate the number of wires to implement a NoC with 32-bit width channels for each SoC. For instance, system d695 has nine routers, thus, it has 24 channels of 32 bits, then, the NoC requires $24 \times 32 = 768$ wires. The column "NoC Wires" of Table 2 presents the amount of wires to implement the NoC channels, which is a reference value to compare with the TAM wiring. The parameter w_{max} represents the number of test pins available for the test architecture.

Second, the TAM wire length of a system based on dedicated TAM depends on how the cores are placed into the NoC. For this reason we evaluate ten placements for each system. The column "TAM Wires" of Table 2 presents the average/best/worst TAM wire length for each system considering different number of test pins. The column "diff" of Table 2 represents the relative number of wires to implement dedicated TAMs compared to the number of wires of the 32-bit NoC. For instance, the 202 wires (see Table 2, SoC d695, $w_{max} = 16$) correspond to 26% (202/768), of wires of a 3x3

Table 2: TAM wire length estimation for dedicated TAMs.

SoC	w_{max}	NoC wires	avg TAM wires	diff (%)	best TAM wires	diff (%)	worst TAM wires	diff (%)
d281	16	768	255	33.23	232	30.21	272	35.42
	24	768	205	26.64	164	21.35	234	30.47
	32	768	147	19.19	130	16.93	156	20.31
	40	768	208	27.06	196	25.52	230	29.95
	48	768	216	28.07	184	23.96	244	31.77
	56	768	242	31.56	202	26.30	264	34.38
	64	768	277	36.02	226	29.43	302	39.32
d695	16	768	202	26.35	168	21.88	212	27.60
	24	768	257	33.46	228	29.69	288	37.50
	32	768	397	51.74	370	48.18	452	58.85
	40	768	222	28.91	170	22.14	284	36.98
	48	768	398	51.88	344	44.79	440	57.29
	56	768	330	42.97	268	34.90	410	53.39
	64	768	445	57.94	356	46.35	534	**69.53**
g1023	16	1088	243	22.33	216	19.85	262	24.08
	24	1088	277	25.50	266	24.45	302	27.76
	32	1088	283	26.03	240	22.06	324	29.78
	40	1088	337	31.01	274	25.18	374	34.38
	48	1088	428	39.38	390	35.85	466	42.83
	56	1088	361	33.16	304	27.94	422	38.79
	64	1088	156	14.36	148	13.60	164	15.07
p22810	16	2560	415	16.23	374	14.61	458	17.89
	24	2560	419	16.35	382	14.92	438	17.11
	32	2560	518	20.22	464	18.13	530	20.70
	40	2560	582	22.74	536	20.94	628	24.53
	48	2560	419	16.38	366	14.30	490	19.14
	56	2560	628	24.52	570	22.27	672	26.25
	64	2560	910	35.55	848	33.13	968	37.81
p34392	16	1536	274	17.85	250	16.28	304	19.79
	24	1536	227	14.79	214	13.93	236	15.36
	32	1536	192	12.49	160	10.42	216	14.06
	40	1536	283	18.44	238	15.49	312	20.31
	48	1536	431	28.05	372	24.22	498	32.42
	56	1536	431	28.05	372	24.22	498	32.42
	64	1536	431	28.05	372	24.22	498	32.42
p93791	16	2880	499	17.34	454	15.76	532	18.47
	24	2880	451	15.66	402	13.96	522	18.13
	32	2880	903	31.37	834	28.96	1010	35.07
	40	2880	1005	34.90	904	31.39	1088	37.78
	48	2880	897	31.14	812	28.19	994	34.51
	56	2880	1154	40.06	1080	37.50	1290	44.79
	64	2880	1324	45.97	1256	43.61	1380	47.92
big	16	9216	1159	12.58	1104	11.98	1224	13.28
	24	9216	1668	18.10	1578	17.12	1758	19.08
	32	9216	479	5.20	278	**3.02**	690	7.49
	40	9216	776	8.42	664	7.20	898	9.74
	48	9216	382	4.14	338	3.67	436	4.73
	56	9216	804	8.72	418	4.54	1182	12.83
	64	9216	1139	12.36	902	9.79	1566	16.99
average	—	—	—	**25.97**	—	**22.74**	—	**29.11**

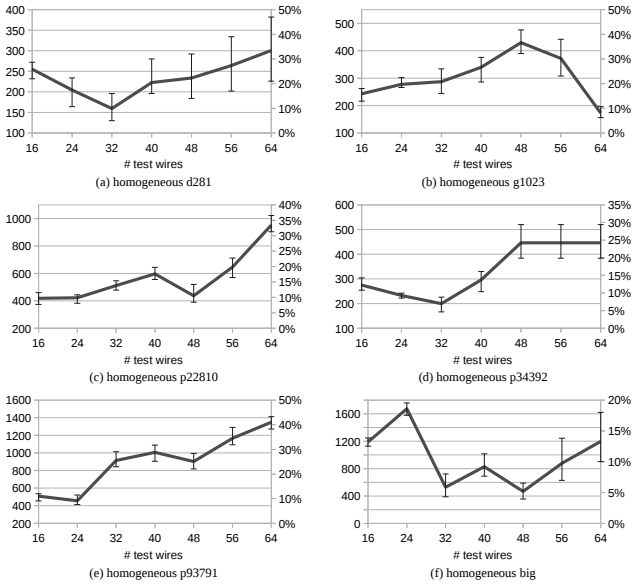


Figure 4: Wire length for dedicated TAMs. The line represents the average wire length for the ten placements while the error bars represent the wire length for the worst and best placements. The number of hops in the *left axis* is used as an absolute wire length unit. The *right axis* shows the percentage of test wiring compared to wiring for NoC data channels.

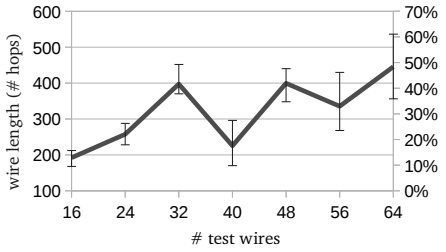
NoC considering channels of 32-bit width. The illustrations of Fig. 4 represent the same results of Table 2, but, it is more intuitive and less detailed. Fig. 5(a) and (b) also compare the results for the homogeneous and the heterogeneous d695, respectively, to demonstrate that both types present similar results. The rest of this section discusses these results individually.

The g1023 SoC (Fig. 4(b)) has reduced TAM wiring when the number of test pins is 54 and 64. It happens because, in this SoC, the test scheduler assigns almost one core per TAM. These cases require a smaller amount of wiring since there are wires only between the chip test pins and the CUT test ports. Most other SoCs have TAMs with more than one core.

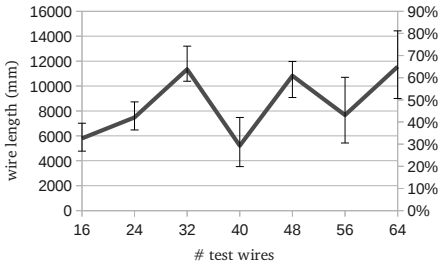
Other cases, which have not been mentioned, fall in one of the following situations: the test scheduler assigns most of the test wires to a TAM with only one core or it assigns few test wires to a TAM with most cores of the SoC. In these cases the TAM has wires only from/to the test pin to the/from the CUT, requiring wide and short wires. The remaining cores of these systems are tested by narrow but long TAMs. The combination of TAMs with wide and short wires and TAMs with narrow but long wires leads to shorter TAM wire lengths, globally. For this reason these systems require fewer wires.

The average results for the average/best/worst placements are, respectively, 25.97%, 22.74%, and 29.11% (bottom of Table 2, in bold). It means that, on average, *about 25.97% of the SoC global wires are used to implement dedicated TAMs.* For some systems, like d695 with 64 test pins, the TAM wiring can be about 70% of the SoC global wires (see d695 row of Table 2, in bold). On the other hand, the *big* SoC with 32 test pins requires about 3% of test wires (see *big* row of Table 2, in bold). Note that the model is optimistic, as explained in Section 3.5. It means that the actual wiring for dedicated TAMs is larger. This variability from 3% to 70% means that it is hard to antecipate the test wiring for a given SoC. The results depends on several information such as the SoC test inforamtion (number of test pattern, number and depth of scan chains, etc), number of test pins, and core placement in the NoC. This variability in wire length justifies the need for tools that can quickly evaluate test wiring upfront.

Finally, Fig. 6 has been generated by grouping the results in Table 2 (columns 5, 7, and 9) in terms of number of test pins and taking the average results. It presents the average usage of TAM wires (for the average, best, and worst placements) per number of test pin considering all SoCs. It shows that the TAM wire length increases as the number of test pins increases, demonstrating that dedicated TAM might not be viable for large number of test pins. It also shows that the error bars are increasing as the number of test pins increases. It means that the impact of placement on the wire length tends to increase as the number of test pins increases.



(a) homogeneous d695



(b) heterogeneous d695

Figure 5: Wire length for the homogeneous and the heterogeneous d695 SoC. The number of hops and *mm* are the wire length units for homogeneous and heterogeneous systems, respectively. The *right axis* shows the percentage of test wiring compared to wiring for NoC data channels.

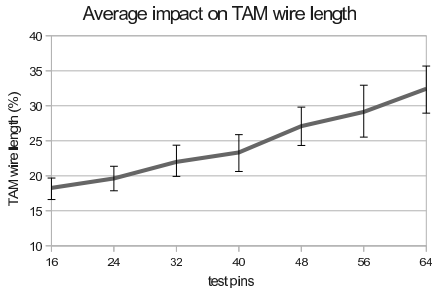


Figure 6: Average amount of test wiring compared to the global wires. The line represents the average percentage of wires used to implement dedicated TAMs. The error bars represent the average percentage for the worst and best placements.

6. CONCLUSION

This paper proposed a wire length estimation model used to quickly evaluate the amount of wiring required to implement dedicated TAMs. To the best of our knowledge, this is the first paper to quantify the amount of wiring saved by using NoC as TAM.

Furthermore, the information about the amount of wire savings at early stages of design is very important because more accurate decisions regarding the type of test architecture can be done earlier, accelerating the whole design flow.

The proposed model has been used to evaluate several ITC'02 SoC Test Benchmarks SoCs, including a large SoC with 117 cores. It has been concluded that the TAMs increase the total number of global wires of the chip in 26%, on average, but the variation can be large, depending on features of the SoC; in some cases it can be 70% while for others cases it can be about 3%. This variation corroborates with the proposed model since the designer can quickly decide, without requiring fully accurate layout information, whether the extra wires required by the dedicated TAMs must or must not be replaced by a NoC TAM.

Acknowledgements

Part of this work was accomplished while Alexandre was holding a doctoral scholarship from the CNPq-PNM (grant number 141993/2002-2). Currently, Alexandre is supported by postdoctoral scholarships from Capes/PNPD (grant number 02388/09-0) and FAPERGS/ARD (grant number 10/0701-2). Cristiano is partially supported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program funds. Marcelo is partially supported by CNPq (Projeto Universal), grant number 478200/2008-0. Fernando is partially supported by CNPq and FAPERGS scholarships, grant number 301599/2009-2 and 10/0814-9, respectively.

7. REFERENCES

[1] F. Angiolini et al. A layout-aware analysis of networks-on-chip and traditional interconnects for MPSoCs. *IEEE Trans. on CAD*, 26(3):421–434, 2007.

[2] T. Bjerregaard and S. Mahadevan. A survey of research and practices on network-on-chip. *ACM Computing Surveys*, 38(1), 2006.

[3] E. A. Carara, R. P. Oliveira, N. L. V. Calazans, and F. G. Moraes. HeMPS - a framework for NoC-based MPSoC generation. In *Proc. ISCAS*, pages 1345–1348, 2009.

[4] C. Chu. FLUTE: fast lookup table based wirelength estimation technique. In *Proc. ICCAD*, pages 696–701, 2004.

[5] E. Cota, L. Carro, and M. Lubaszewski. Reusing an On-Chip Network for the Test of Core-based Systems. *ACM TODAES*, 9(4):471–499, 2004.

[6] S. K. Goel and E. J. Marinissen. Layout-driven SOC test architecture design for test time and wire length minimization. In *Proc. DATE*, pages 738–743, 2003.

[7] S. K. Goel and E. J. Marinissen. SOC test architecture design for efficient utilization of test bandwidth. *ACM TODAES*, 8(4):399–429, Oct. 2003.

[8] E. Larsson and H. Fujiwara. Test resource partitioning and optimization for SOC designs. In *Proc. VTS*, page 319, 2003.

[9] E. J. Marinissen, V. Iyengar, and K. Chakrabarty. A set of benchmarks for modular testing of SOCs. In *Proc. ITC*, pages 519–528, 2002.

[10] M. Nahvi and A. Ivanov. A Packet Switching Communication-Based Test Access Mechanism for System Chips. In *Proc. ETW*, pages 81–86, 2001.

[11] B. T. Preas and M. J. Lorenzetti. *Physical Design Automation of VLSI Systems*. Benjamin Cummings Publishing Company, 1988.