VLSI DESIGN 2001, Vol. 12, No. 2, pp. 113–124 Reprints available directly from the publisher Photocopying permitted by license only © 2001 OPA (Overseas Publishers Association) N.V. Published by license under the Gordon and Breach Science Publishers imprint.

## Symmetric and Programmable Multi-Chip Module for Low-Power Prototyping System

MAO-HSU YEN<sup>a,\*</sup>, SAO-JIE CHEN<sup>b</sup> and SANKO H. LAN<sup>a</sup>

<sup>a</sup>Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC; <sup>b</sup>Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC

(Received 20 June 2000; In final form 3 August 2000)

The advantages of a Multi-Chip Module (MCM) product are its low-power and smallsize. But the design of an MCM system usually requires weeks of engineering effort, thus we need a generic MCM substrate with programmable interconnections to accelerate system prototyping. In this paper, we propose a Symmetric and Programmable MCM (SPMCM) substrate for this purpose. This SPMCM substrate consists of a symmetrical array of slots for bare-chip attachment and Field Programmable Interconnect Chips (FPICs) for substrate routing. Experimental results demonstrate that our FPIC polygonal routing module uses 12% less switches than the conventional routing module for interconnecting bare-chip slots with 84 pads. Also, experiments are conducted to determinate proper parameters for the VLSI implementation of our FPIC.

Keywords: SPMCM; PMCM; MCM; FPIC; Low power; Prototype

## **1. INTRODUCTION**

Portable systems design and add-on cards have stringent limits on low-power and small-size constraints. A Multi-Chip Module (MCM) is a device in which several bare-chips are attached to a single substrate and then packaged as a smallsize and low-power system. Furthermore, MCM packaging technology used in electronic systems translate the semiconductor speed into system performance [1-3], but low-power and highdensity MCMs are expensive to fabricate and usually requires weeks of engineering effort for system prototyping and product verification. The engineering delay in designing and fabricating such MCMs become unacceptable in today's competitive market. The needs of quick turnaround time, high product yield, and low cost have led to the development of another approach, called Symmetric and Programmable Multi-Chip Module (SPMCM) [4–7]. This SPMCM technology provides the designers with a pre-characterized MCM substrate and some programmable interconnections such that they can generate a fast

<sup>\*</sup>Corresponding author.

prototyping or a final consumer product in a short time. The advantages of SPMCM are that the field programmable technology can reduce product development cycle and NRE (Non-Recurrence Engineering) cost, while MCM technology can achieve low power and small size.

Several systems have been proposed for the lowpower and small-size prototyping system design on MCM [8-13], most of them interconnect the Field Programmable Gate Arrays (FPGAs) with some Field Programmable Interconnect Chips (FPICs) [14-17] on an MCM substrate. For instance, the BORG [8,9] system is a reconfigurable prototyping board for FPGAs based on the Clos network. Galloway et al. [10] proposed a reconfigurable system, called Transmorgrifer-2, which is a hierarchical design based on the I-CUBE [15] routing chip. The Field Programmable Multi-Chip Module (FPMCM) [11] system is a reconfigurable system combining both the state-of-art FPGA and MCM technologies. Thomae and Bout [12] devised a multi-FPGA board for rapid prototyping, in which the ring architecture is used to interconnect FPGAs. A board for logic emulation has been developed by Babb et al., at MIT [13], which uses virtual-wires technique to overcome the pin count limitations. From the above existing reconfigurable systems [8-13], we observe that most of the efforts have been spent on designing a flexible interconnection architecture to mitigate the pin limitation.

In order to improve the foregoing problem, we propose an SPMCM structure, which consists mainly of a symmetrical array of bare-chip slots surrounded with some FPICs for slot interconnections. The bare-chip slots allow bare chips (BCs) from different manufacturing processes to attach on the MCM substrate; therefore, our architecture is more flexible and can be used to realize a lowpower and small-size prototyping system containing bare-chips of various technologies. Our proposed FPIC architecture uses polygonal routing modules and virtual-wires [13] techniques to reduce the requirements of programmable switches and pin count. Since this architecture spends less hardware cost and has a regular structure, it is suitable for VLSI implementation. Moreover, cascading the architecture can scale up the routing resources. This paper focuses on the design of efficient FPICs and the structure of a flexible barechip slot in an SPMCM, which can be used to support a low-power and small-size prototyping system.

The remainder of this paper is organized as follows. First, we show the SPMCM and the bare-chip slot structure in Section 2. Section 3 describes a brief review of the conventional routing module; then our proposed polygonal routing module architecture and some experimental results are shown. Section 4 depicts our FPIC VLSI implementation, its polygonal routing modules, and virtual-wires technique. Conclusions are reported in Section 5.

## 2. SPMCM ARCHITECTURE

Our SPMCM is a programmable MCM substrate [5-7] that consists of an array of bare-chip slots and interconnection FPICs [14-17] on an MCM substrate, as shown in Figure 1. The MCM substrate and FPICs are pre-fabricated in large volume and well tested. On the MCM substrate, parts of the pads are designed for the FPICs; others are for the commercial or customized bare chips attached to the bare-chip slots. The FPICs are attached to the MCM substrate using flipchip bounding technology, while bare-chips using wire bounding technology. Thus, these bare chips attached on the SPMCM can be manufactured with different processes. On the substrate, each bare-chip pad is connected via a substrate metal wire to one of the FPIC pads, and net routing is accomplished by programming the FPICs.

The purpose of these flexible bare-chip slots is aimed at attaching bare chips on an SPMCM in different combinations. Figures 2(a) and (b) show two different usages of our four-slot structure. In each of the bare-chip slots, each pair of horizontal (or vertical) pads is connected together through



FIGURE 1 SPMCM architecture.

substrate metal wiring. Each bare-chip pad (illustrated by black pad) is connected *via* a substrate metal wire to one of the FPIC pads. We can have four small bare chips attached to the four small bare-chip slots, as shown in Figure 2(a), or a large bare chip occupying all the four slots, as shown in Figure 2(b). When applying this SPMCM design methodology for prototyping, designers do not need to consider the MCM substrate design, because SPMCM owns a reprogrammable interconnection MCM substrate which permits us to quickly reconfigure the prototyping system. With this SPMCM, we can flexibly attach the I/O pads on the bare chips to the bare-chip slots on the MCM substrate. For low- to medium-volume MCM designs, this SPMCM design methodology can lower design costs and shorten design cycles. Moreover, the design engineers do not need to have high MCM design skills in order to design an MCM system.

While SPMCM technology reduces the engineering delays and the cost of MCM development by a significant margin, it also degrades system performance in comparison with fully customized MCMs, due to the programmable switches usually have high resistance and capacitance, and occupy a large area. The number of programmable switches of an FPIC affects its speed performance, die size, and routability. Intuitively, increasing the number of programmable switches in an FPIC deliver good routability. However, an FPIC with fewer programmable switches can reduce the impedance of interconnect paths, and the overall speed of the SPMCM can thus be improved.



FIGURE 2 Bare-chip slots structure; (a) Attached with several small bare chips, and (b) Attached with a large bare chip.



FIGURE 3 (a) A typical symmetric FPGA model and routing module RM(4, m, n); (b) Switch module SM(4, m); and (c) Connection module CM(m, n).  $F_C = 6$ ,  $F_S = 3$ , m = 6, n = 4 and  $r_{F_C} = 1$ .

Our proposed SPMCM is similar to the symmetric FPGA (Xilinx XC4000-type) [18-19]. A brief review of the symmetric FPGA architecture is given as follows. A typical symmetric FPGA consists of an array of logic modules that can be interconnected using routing resources, as shown in Figure 3(a). The routing resources comprise metal wires and routing modules. Thus, an arbitrary digital circuit can be implemented by appropriately configuring these routing modules and logic modules. A routing module (RM) consists of two connection-modules (CMs) connected to a switch-module (SM), and each of these modules contains itself many programmable switches, as shown in Figure 3(a). The routing module is the section of the routing resources to be replicated across the entire symmetric FPGA. The logic-module (L) contains configurable digital circuits to implement logic functions. The input and output pins of a logic module are connected to its surrounding connection modules, which in turn are connected to the switch modules.

Similar to the symmetric FPGA model shown in Figure 3(a), if we substitute the logic modules with the bare-chip slots and each of the routing modules with an FPIC, we obtain an SPMCM system, as shown in Figure 1. In this manner, the routing algorithm and architecture of the SPMCM are similar to the symmetric FPGA [19]. Therefore, we use the terms "routing module" and "FPIC" interchangeably in this paper. In the following sections, we will first indicate that the conventional routing module presents an obstacle to the implementation of an FPIC in terms of the number of programmable switches. Thus, we propose a polygonal routing module architecture to minimize the number of programmable switches.

## 3. CONVENTIONAL ROUTING MODULE AND POLYGONAL ROUTING MODULE

#### 3.1. Conventional Routing Module

In a conventional symmetric FPGA, the switch module is a 4-sided block, denoted as SM(4, m), where m is the number of terminals on each side of the switch module. For example, a Xilinx XC4000type SM(4, m) can be partitioned into m independent submodules SM(4,1), as shown in Figure 3(b). Let the flexibility of a switch module be  $F_S$ [18], which is used to represent the number of programmable switches connecting one terminal to  $F_S$  terminals on the other three sides of a switch module. For a conventional switch module with  $F_S = 3$ , its switch module SM(4, m) would contain 6m programmable switches.

A connection module, denoted as CM(m, n), is an  $m \times n$  rectangular block, where m is the number of tracks connected to the switch modules, and nis the number of tracks connected to the bare-chip slots (logic modules), as shown in Figure 3(c). Therefore, each bare-chip slot can have at most 2npads. A conventional routing module consisting of two connection modules CM(m, n) and a switch module SM(4, m) is denoted as RM(4, m, n), as shown in Figure 3(a). The flexibility of a connection module [18],  $F_C$ , is defined as the number of tracks to which each pad in a bare-chip slot (logic module) can be connected; for the example in Figure 3(c),  $F_C = 6$ . Thereafter, a connection module can contain  $F_C \times n$  programmable switches. In a connection module CM(m, n), the ratio of  $F_C$ to *m* is called the flexibility ratio, *i.e.*,  $r_{F_c} = F_c/m$ . This  $r_{F_c}$  is the probability that a wire arriving at a particular track in the connection module is able to connect to the required pin of a bare-chip slot (logic module) [18], thus  $0 \le r_{F_c} \le 1$ . Rose and Brown [18] suggested that  $F_S = 3$  and a high value of  $F_C$ , *i.e.*,  $r_{F_C}$  close to 1, are sufficient to provide high routability in a symmetric FPGA. For example, the Xilinx XC4000 family FPGAs

# 3.2. Number of Switches in a Conventional Routing Module

use  $F_S = 3$  and  $r_{F_C} = 1$ .

Let the number of programmable switches in an RM(4, m, n) be denoted as PS(4, m, n), which is equal to the number of programmable switches between two CM(m, n) and an SM(4, m). This is given by:

$$PS(4, m, n) = 2F_C n + 6m = 2r_{F_C} mn + 6m$$
  
= 2m(r\_{F\_C} n + 3) (1)

For the Xilinx XC4000 family FPGAs with  $F_S = 3$ and  $r_{F_c} = 1$ , we have:

$$PS(4, m, n) = 2m(n+3)$$
 (2)

In terms of the number of switches, we will show in our experiments that the conventional routing module RM(4, m, n) is unsuited for interconnecting bare-chip slots with high pincount, because the PS(4, m, n) values obtained are very large. Furthermore, bare-chip slots with a large 2n number of pins are very usual in an SPMCM system. This presents an obstacle to the VLSI implementation of an RM(4, m, n) FPIC. Therefore, in order to improve the switchefficiency, we propose a polygonal routing module that consists of many small connection modules connected to a polygonal switch module for interconnecting bare-chip slots with high pincount.

#### 3.3. Polygonal Routing Module

Based on the conventional routing module RM (4, m, n), as shown in Figure 3(a), we can divide each connection module CM(m, n) into s smaller connection modules CM(m', n') such that each CM(m', n') is connected to one of the 4s sides of the polygonal switch module, as shown in Figure 4(a), where  $m = s \times m'$ ,  $n = s \times n'$ ,  $F_C =$  $s \times F_C'$ ,  $F_C$  and  $F_C'$  are the flexibilities of CM(m, n)and CM(m', n'), respectively. The polygonal switch module is a 4s-side block, denoted as SM(4s, m'), where m' is the number of terminals on each side of the polygonal switch module, as shown in Figure 4(b). Furthermore, a terminal in one side can be connected to a terminal in one of the other (4s-1) sides of the SM(4s, m') through programmable switches, thus the flexibility  $F_{S}'$  of an SM (4s, m') is equal to (4s-1). A polygonal switching module SM(4s, m') can be partitioned into m' independent submodules SM(4s, 1). Compared with the conventional routing module, a polygonal routing module RM(4s, m', n') comprises 2s smaller connection modules CM(m', n') interconnected by a 4s-side switch module SM(4s, m'), as shown in Figure 4(a). That is to say, the conventional routing module RM(4, m, n) is a special case of our polygonal routing module RM(4s, m', n')with s=1. For example, Figure 4(a) represents a polygonal routing module RM(8, 3, 2) with s = 2and Figure 3(a) represents a conventional routing module RM(4, 6, 4) with s = 1.



FIGURE 4 (a) Polygonal routing module RM(4s, m', n'); (b) Polygonal switch module SM(4s, m'); and (c) Connection module CM(m', n').  $F_C' = 3$ ,  $F_S' = 7$ , m' = 3, n' = 2,  $r_{F_C}' = 1 = r_{F_C}$  and s = 2.

## 3.4. Number of Switches in a Polygonal Routing Module

Since the number of switches needed by 2s connection modules CM(m', n') is equal to  $2sF_C'n'$ , and the number of switches in a polygonal switch module SM(4s, m') is equal to  $m'((4s-1)+(4s-2)+\cdots+1)$ . Denote the number of switches in an RM(4s, m', n') as PS(4s, m', n'). By summing the number of switches in all the above modules,

we have:

$$PS(4s, m', n') = 2sF_C'n' + 2m's(4s - 1) = 2F_C'n + 2m's(4s - 1)$$
(3)

Let  $r_{F_c}' = 1$ , we have  $F_c' = m'$  substituting it into Eq. (3) results in

$$PS(4s, m', n') = 2nm' + 2m's(4s - 1)$$
  
= 2m'(4s<sup>2</sup> - s + n) (4)

TABLE I Minimum number of tracks and switches needed for detailed-routing completion in original net order

Grouped 12 4-LUT			Polygonal Routing Module RM(4s, m', n')							
Large			<i>RM</i> (4, *)		<i>RM</i> (8, *)		<i>RM</i> (12, *)		<i>RM</i> (16, *)	
Circuit	logic modules	# Con.	m = m'	PS	m = 2m'	PS	m = 3m'	PS	m = 4m'	PS
BUSC	10 × 10	392	31	3162	40	2480	51	2754	48	2592
DMA	$12 \times 12$	71	38	3876	56	3472	66	3564	72	3888
DFSM	$16 \times 16$	1422	39	3978	54	3348	66	3564	76	4104
BNRE	$14 \times 16$	1257	47	4794	68	4216	81	4374	84	4536
ZO3	$18 \times 18$	2135	50	5100	70	4340	84	4536	100	5400
9symm1	$8 \times 8$	259	32	3264	38	2356	51	2754	44	2376
alu2	$10 \times 10$	511	33	3366	46	2852	57	3078	64	3456
alu4	$14 \times 12$	851	45	4590	60	3720	75	4050	80	4320
apex7	$8 \times 8$	300	31	3162	44	2728	57	3078	60	3240
example 2	$10 \times 10$	444	41	4182	62	3844	78	4212	80	4320
k2	$8 \times 8$	1256	61	6222	80	4960	96	5184	100	5400
term1	$10 \times 10$	202	32	3264	44	2728	54	2916	60	3240
too large	$10 \times 10$	519	38	3876	52	3224	60	3240	64	3456
vda	$12 \times 12$	722	48	4896	60	3720	72	3888	68	3672
TOTAL	_		566	57732	774	47988	948	51192	1000	54000
Comparison		-	1	1	1.37	0.83	1.67	0.89	1.77	0.93

From Eq. (4), PS(4s, m', n') is determined by the s and m' values because n is constant. In the following subsection, we will find the proper s and m' values to minimize the number of switches needed in a polygonal routing module through experiment.

## 3.5. Experimental Results

In Figures 1 and 3(a), each of our bare-chip slot has 84 (2n) pads to connect to the FPICs in the SPMCM system. To explore the effects of s, m' and n' values of a polygonal routing module on the switch-efficiency of an SPMCM, we implemented a maze router in C language on a SUN Ultra-1 workstation. We examine three parameters s, m'and n' related to the number of the switches needed in the CGE [18] and SEGA [20] benchmark circuits. Note that no industrial benchmarks for SPMCM are available. For modeling the bare-chips in the SPMCM system, the N 4input look-up tables (4-LUTs) are grouped to form thirteen large modules in these circuits, where  $N = 4, 5, \dots, 16$ . A larger logic-module (bare-chip) has 8N pins, and each pin in a large logic-module can be connected to any of the m' tracks  $(r_{F_c}' = 1)$ in a connection module. Because net ordering often affects the performance of a maze router, we router the benchmark circuits by using the following three net-ordering schemes to avoid possible biases: (1) original net order in the benchmark

 TABLE II Minimum number of switches needed for 14

 benchmark circuits in original net order

Pins	Polygona				
of BC	<i>PS</i> (4, *)	<i>PS</i> (8, *)	<i>PS</i> (12, *)	<i>PS</i> (16, *)	Ratio
32	14288	15600	22540	25688	1
40	18538	19992	25864	29600	1
48	23166	22952	27816	34272	0.99
56	29450	28812	34770	38720	0.98
64	34580	32016	37180	39192	0.93
72	38844	36000	40158	44544	0.93
80	45666	40284	47304	48400	0.88
88	50854	43732	47740	49504	0.86
96	57732	47988	51192	54000	0.83
104	66220	56100	59840	59584	0.85
112	73396	60340	65148	64264	0.82
120	79254	63936	65844	65760	0.81
128	85894	68484	70616	69192	0.80

circuits, (2) longest net first, and (3) shortest net first.

By detailed routing these large logic modules each having different pin size, the switches performance of our polygonal routing module was evaluated. For the original net order in the benchmark circuits, Tables I and II show the results. From the routing results of a 96-pin (N=12) logic module as listed in Table I, we first determined the minimum number of tracks m' required for 100% routing completion for each circuit, in each of the four cases of polygonal routing modules RM(4s, m', n') with s = 1, 2, 3 and 4, respectively. Then we get the PS(4s, m', n') value

TABLE III Minimum number of switches needed for 14 benchmark circuits in longest net first

Pins	Polygonal Routing Module $RM(4s, m', n')$				
of BC	<i>PS</i> (4, *)	<i>PS</i> (8, *)	<i>PS</i> (12, *)	<i>PS</i> (16, *)	Ratio
32	13072	14580	20188	22648	1
40	17066	17884	23426	26560	1
48	20952	20672	26220	30576	0.99
56	27590	26628	32208	35552	0.97
64	32690	29992	35750	34776	0.92
72	36504	33400	36708	40512	0.91
80	43602	37908	43216	43800	0.87
88	48786	40020	45430	44512	0.82
96	55488	45384	48276	49032	0.82
104	63580	52272	57290	54880	0.82
112	71508	57260	62478	59624	0.80
120	76608	60680	62310	59520	0.78
128	84420	64740	66348	63736	0.75

TABLE IV Minimum number of switches needed for 14 benchmark circuits in shortest net first

Pins	Polygonal Routing Module $RM(4s, m', n')$					
of BC	<i>PS</i> (4, *)	<i>PS</i> (8, *)	<i>PS</i> (12, *)	<i>PS</i> (16, *)	Ratio	
32	18430	21300	30478	34656	1	
40	24104	27336	35298	41600	1	
48	28836	30248	38304	46032	1	
56	36890	37128	45872	52096	1	
64	42630	42136	49530	52992	0.99	
72	47970	46000	52578	58560	0.96	
80	56330	52704	61028	64400	0.94	
88	63356	57420	62986	68224	0.91	
96	69768	61752	67392	71064	0.89	
104	79750	72336	77690	80864	0.91	
112	87556	76860	82236	85376	0.88	
120	93492	81696	85746	89040	0.87	
128	102376	87828	90598	91760	0.86	

by substituting the obtained s, m' and n into Eq. (4). Table II shows the total of programmable switches PS(4s, m', n') of 14 benchmarks varies with s for larger logic-modules (modeling barechips) with 8N pins, where s = 1, 2, 3 and 4, and N = 4, 5, ..., 16. For the longest and shortest net first, the Tables III and IV show the results, respectively. Experimental results demonstrate that the conventional routing module PS(4, m, n) works quite well only for a bare-chip with less than 40-pin. Each of our bare-chip slot has 84 (2n) pads, the RM(8, m', n') FPIC is well-chosen to interconnect bare-chips in the SPMCM. From Tables II, III and IV, the RM(8, m', 21) FPIC compare with the conventional routing module (4-side), an average



(a)



FIGURE 5 (a) An RM(8, 2, 21) formed by two scalable polygonal routing modules RM(8, 1, 21)'s; (b) Switch module SM(8, 1); and (c) Connection module CM(1, 21).  $F_{C}' = 1$ ,  $F_{S}' = 7$ , m' = 1,  $r'_{F_{C}} = 1$  and s = 2.



FIGURE 6 A polygonal routing module RM(8, 6, 21) architecture with  $F_{c'}=6$ ,  $F_{s'}=7$ , m'=6, n'=21,  $r_{F_{c'}}=1$  and s=2.

12% improvement in the switches performance was achieved.

Thus, our polygonal routing modules can be used to improve switch-efficiency of an SPMCM system. Although the polygonal routing module needs more number of tracks, but the number of switches it needs is much reduced. State-of-art VLSI technologies providing multi-metal layers can be used to solve the larger metal tracks requirement. Experimental results and VLSI technologies demonstrate that implementing our proposed polygonal routing module RM(8, m', 21) in an FPIC could enlarge the practicability of an SPMCM system.

## 4. VLSI CHIP IMPLEMENTATION OF THE POLYGONAL ROUTING MODULE

## 4.1. Scalable Polygonal Routing Module

The feature of scalability is very important to the architecture design of an FPIC and to its VLSI implementation. We will show that our polygonal routing modules possess the characteristics of scalability. An FPIC using RM(8, 1, 21)'s as its

routing modules is shown in Figure 5(a). In each RM(8, 1, 21), we use one switch module SM(8, 1) as shown in Figure 5(b), and four CM(1, 21) connection modules as shown in Figure 5(c).

Cascading m' RM(8, 1, 21)'s can be used to interconnect bare-chips, as shown Figure 5(a). In this manner, the m' cascaded RM(8, 1, 21)'s are equal to an RM(8, m', 21) routing module. Thus, the number of routing tracks is increased by m' times. That is to say, the routing resources were increased by m' times in a cascaded SPMCM. Our FPIC uses an RM(8, 6, 21) with 132 pins, where m' = 6, and is packaged as a 160-pin CQFP. An RM(8, 6, 21)polygonal routing module is shown in Figure 6. Similarity, the k cascaded RM(8, 6, 21)'s are equal to an RM(8, 6k, 21) routing module.



FIGURE 7 FPIC with hard wires interconnect.

## 4.2. Virtual Wires

Virtual-wires [13] technology is used in our FPIC to improve the routing resources and to overcome the pin count limitation by multiplexing each physical wire among multiple logical wires. Figure 7 shows an example of four logical wires allocated to four physical wires in an RM(8, 6, 21). Figure 8 shows the same example with the four



FIGURE 8 FPIC with virtual wires interconnect.



FIGURE 9 Chip layout of the FPIC.

122

TABLE V Performance data

Pin Count	160 pins CQFP
Technology	0.6µm SPTM (CMOS)
Transistor Count	168 K
Die Size	$4990 \times 4990 \ (\mu m)^2$
System clock rate	16 MHz
Point-to-point delay	10.01 ns

logical wires connected to four RM(8, 6, 21)'s by a single physical wire. The physical wire has to multiplex and demultiplex respectively between the bare chip and the FPIC. The FPIC VLSI implementation combining four RM(8, 6, 21)'s and multiplexing is shown in Figure 8.

## 4.3. Implementation

The chip layout of our FPIC is shown in Figure 9 and it will be fabricated in a  $0.6 \mu m$  Single-Poly-Triple-Metal (SPTM) CMOS technology through the Chip Implementation Center (CIC), National Science Council, R.O.C. its performance data are summarized in Table V. As mentioned above, this chip architecture is highly scalable, uses less programmable switches, and has lower pin count. Thus it is very suitable for VLSI implementation.

## 5. CONCLUSIONS

For low power and small size prototyping system design, an area-efficient and flexible Symmetric and Programmable MCM (SPMCM) has been described, and a symmetric-array FPIC VLSI architecture was proposed for the substrate routing between the bare-chip slots. This FPIC architecture consists of four polygonal routing modules and multiplexing structures, which can significantly reduce the requirements of programmable switches number and pin count compared with a conventional routing module. We have implemented this VLSI architecture in a  $0.6 \,\mu m$ CMOS technique to verify the function of our proposed SPMCM. In addition, this VLSI routing chip architecture can be easily scaled up with the routing resources. With its field programmable MCM substrate, our SPMCM can be used for implementing various prototyping designs based on user's requirements without going through the foundry facility. The advantages are that the field programmable technology can reduce product development cycle and NRE (Non-Recurrence Engineering) cost, while MCM technology can achieve low power and small size.

#### Acknowledgement

This work was partly supported by the National Science Council, R.O.C. under Grant NSC-88-2215-E002-037.

#### References

- Johnson, R., "Multichip Modules: Next-Generation Packages," *IEEE Spectrum*, pp. 34–48, March, 1990.
- [2] Iqbal, A., Swaminathan, M., Nealon, M. and Omer, A., "Design Tradeoffs among MCM-C, MCM-D and MCM-D/C Technologies," *IEEE Transactions on Components, Packaging, and Manufacturing-Part B: Advanced Packaging*, 17(1), 22-29, February, 1994.
- [3] Frye, R. C., "Physical Scaling and Interconnection Delays in Multichip Modules," *IEEE Transactions on Components, Packaging, and Manufacturing-Part B: Advanced Packaging*, 17(1), 30-37, February, 1994.
- [4] Yen, M. H., Chen, S. J. and Lan, S. H. (1999). "Symmetric and Programmable Multi-Chip Module for Rapid Prototyping System," *Proc. 1999 IEEE Workshop on SiGNAL PROCESSING SYSTEMS Design and Implementation*, pp. 301-310.
- [5] Dobbelare, A., El Gamal, How, D. and Kleveland, B., "Field Programmable MCM System-Design of an Interconnection Frame," *Proc. of the IEEE 1992 Custom Integrated Circuits Conference*, pp. 15.1/1-6.
- [6] Burman, S. and Naveed, Sherwani, A., "Programmable Multichip Module," *IEEE Micro*, pp. 28-35, April, 1993.
- [7] Lan, S. H., Architecture and CAD Tools for a Field Programmable Multi-Chip Module, *Ph.D. Thesis*, Stanford University, August 1995.
- [8] Chan, P. K., Schlag, M. and Martin, M. (1992). "BORG: A Reconfigurable Prototyping Board Using Field-Programmable Gate Arrays," *Proc. of the 1st International* ACM/SIGDA Workshop on Field-Programmable Gate Arrays, pp. 47-51.
- [9] Chan, P. K. and Schlag, M. (1993). "Architecture Tradeoffs in Field Programmable Device-Based Computing Systems," Proc. IEEE Workshop on FPGAs for Custom Computing Machines, pp. 152-161.
- [10] Galloway, D., Karchmer, D., Chow, P., Lewis, D. and Rose, J., "The Transmogrifier: The University of Toronto Field-Programmable System," Proc. Second Canadian Workshop on Field-Programmable Devices, 1994.

- [11] Darnauer, J., Garay, P., Isshiki, T., Ramirez, J. and Dai, W. W.-M. (1994). "A Field Programmable Multi-Chip Module (FPMCM)," Proc. IEEE Workshop on FPGAs for Custom Computing Machines, pp. 1–10.
- [12] Thomae, D. A. and Van Den Bout (1992). "Automatic Circuit Partitioning in the Anyboard Rapid Prototyping System," *Microprocessors and Microsystems*, pp. 283-90.
- [13] Babb, J., Tessier, R. and Agarwal, A. (1993). "Virtual Wires: Overcoming Pin Limitations in FPGA-Based Logic Emulators," Proc. IEEE Workshop on FPGAs for Custom Computing Machines, pp. 142–151.
- [14] Yen, M. H., Shie, M. C. and Lan, S. H. (1999). "Polygonal Routing Network for FPGA/FPIC," Proc. 1999 International Symposium on VLSI Technology, System, and Applications, pp. 104-107.
- [15] I-Cube, Inc., *I-Cube FPID Family Data Book*, November 1993.
- [16] Aptix, Inc., Aptix Field Programmable Interconnect Data Book, 1993.
- [17] Guo, R., Nguyen, H., Srinivasan, A., Verheyen, H., Cai, H., Law, S. and Mohsen, A., "A 1024 Pin Universal Interconnection Array with Routing Architecture," *Proc. IEEE 1992 Custom Integrated Circuits Conference*, pp. 4.5.1–4.5.4, May, 1992.
- [18] Rose, J. and Brown, S., "Flexibility of Interconnection Structures for Field-Programmable Gate Arrays," *IEEE J. Solid State Circuits*, 26(3), 277–282, March, 1991.
- [19] Sun, Y., Wang, T. C., Wong, C. K. and Liu, C. L., "Routing for Symmetric FPGAs and FPICs," *Proc. IEEE Trans. Computer Aided Design*, 16(1), 20-31, January, 1997.
- [20] Lemienx, G. G. and Brown, S. D. (1993). "A Detailed Routing Algorithm for Allocating Wire Segments in Field-Programmable Gate Arrays," *Proc. ACM/SIGDA Physical Design Workshop*, pp. 215–216, Lake Arrowhead, CA.

### **Authors' Biographies**

**Mao-Hsu Yen** received the B.S. and M.S. degrees in Electronic Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1991 and 1993, respectively. He is currently working towards the Ph.D. degree in Department of Electronic Engineering at National Taiwan University of Science and Technology. His research interests include Configurable VLSI physical design, architecture and VLSI architecture.

Sao-Jie Chen has been a member of the faculty in the Department of Electrical Engineering, National Taiwan University, where he is currently a full professor. During the fall of 1999, he was a visiting scholar at the Department of Electrical and Computer Science and Engineering, University of California, San Diego. His current research interests include: VLSI circuits design, VLSI physical design automation, and object-oriented software engineering. Dr. Chen is a member of the Association for Computing Machinery, the IEEE, and the IEEE Computer Society.

Sanko H. Lan received the M.S. degree in electrical and computer engineering from North Carolina State University in 1988 and the Ph.D. degree in electrical engineering from Stanford University in 1995. He was a software engineer at Actel Corporation, California, USA in 1988-1990. Since 1995, he has been a faculty member in the Department of Electrical Engineering, National Taiwan University of Science and Technology. His current research interests include Configurable VLSI physical design, architecture, and automation algorithms. Dr. Lan is a member of the Chinese Institute of Engineers and the Institute of Information and Computing Machinery, R.O.C.

124





Rotating Machinery

Hindawi



Journal of Sensors



International Journal of Distributed Sensor Networks





Journal of Electrical and Computer Engineering



Advances in OptoElectronics

Advances in Civil Engineering

> Submit your manuscripts at http://www.hindawi.com









International Journal of Chemical Engineering



**VLSI** Design

International Journal of Antennas and Propagation



Active and Passive Electronic Components



Shock and Vibration



Advances in Acoustics and Vibration