An AND-LUT Based Hybrid FPGA Architecture

Chen Liguang, Lai Jinmei†, and Tong Jiarong

(*Department of Microelectronics, Fudan University, Shanghai* 200433, *China*)

Abstract: A new hybrid FPGA architecture is proposed. The logic tile, which consists of a logic cluster and related connection boxes (CBs), can be configured as either programmable logic arrays (PLAs) or look-up tables (LUTs). This architecture can be classified as an AND-LUT array. PLAs are suitable for the implementation of high fan-in logic circuits, while LUTs are used to implement low fan-in logic circuits. As a result, the proposed hybrid FPGA architecture (HFA) is more flexible to improve logic density. Experiments based on MCNC benchmark circuits were performed in both the hybrid architecture and conventional LUT-based symmetrical FPGA architecture in term of area consumption. Preliminary results indicate that on average, the area is reduced by 46% using the new hybrid architecture.

Key words: hybrid FPGA; AND-LUT array; AND-OR array; PLA; LUT
EEACC: 1130; 1130B
CLC number: TN752 **Document code**: A **Article ID**: 0253-4177(2007)03-0398-06

1 Introduction

For the past twenty years, programmable devices have been widely used in digital systems. There are two main types of programmable devices, field programmable gate arrays (FPGA) and complex logic devices (CPLD). FPGAs are usually based on look-up tables (LUTs), the cost of which increases exponentially according to the inputs of circuits when used to implement random logic circuits. Thus, this architecture is suitable for low fan-in logic circuits. On the other hand, CPLDs are based on AND-OR arrays (or PLAs). A PLA usually has tens of inputs and is appropriate for high fan-in logic circuits.

Much research work has been addressed to improve the logic density of LUT-based FPGAs by mixing LUT and PLA. Kaviani *et al*. presented a typical hybrid FPGA architecture in Ref. [1]. A fixed ratio of 4 ∶ 1 is chosen for the number of LUT blocks (LUTB) and PLA blocks (PLAB, 16 inputs, 10 product terms and 3 outputs) in his architecture. But every circuit has a best ratio between LUTB and PLAB, so a fixed ratio decreases the flexibility of FPGA. As a result, this architecture is inappropriate for various circuits.

Implementing partial logic circuits with PLAs is also found in commercial FPGAs of the two biggest FPGA providers, Xilinx and Altera. An embedded system block (ESB) of Altera APEX20K can be configured as a PLA with 32 inputs, 32 product terms and 16 outputs[2]. Similarly, every Xilinx Virtex II slice has a dedicated OR gate named ORCY, with which a Virtex II configurable logic block (CLB) can implement 2 product terms with 16 inputs[3]. These architectures benefit the logic density of FPGA. However, they are not optimized specifically for implementing PLA, which results in low-efficiency in application. A detailed comparative analysis is provided in Section 2.3 of this paper.

In this paper, a new hybrid FPGA is proposed to solve these problems. The logic tile of the new architecture can be configured as either PLAs or LUTs without any ratio limit. Experiments based on MCNC benchmark circuits were performed between the hybrid architecture and conventional LUT-based symmetrical FPGA architecture in terms of area consumption. Preliminary results indicate that on average, the area is reduced by 46% using the new hybrid architecture.

2 A new hybrid FPGA architecture

2.1 Symmetrical FPGA architecture

A conceptual diagram of a symmetrical FP-

† Corresponding author. Email: jmlai@fudan.edu.cn
Received 3 August 2006, revised manuscript received 10 October 2006

GA is shown in Fig. 1. It consists of programmable logic clusters, connection boxes (CBs), and switch boxes (SBs). The connection box of the symmetrical FPGA, as shown in Fig. 2, connects signals between routing channels and logic clusters.

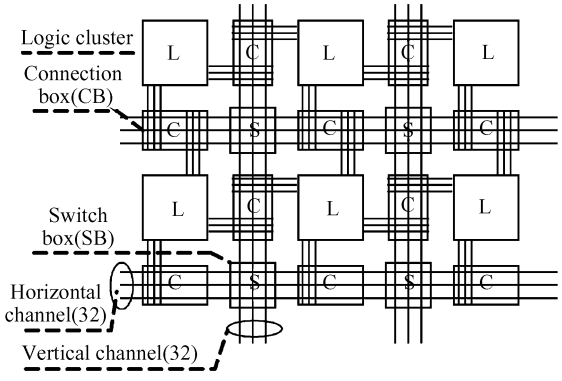

Fig. 1　Symmetrical FPGA architecture

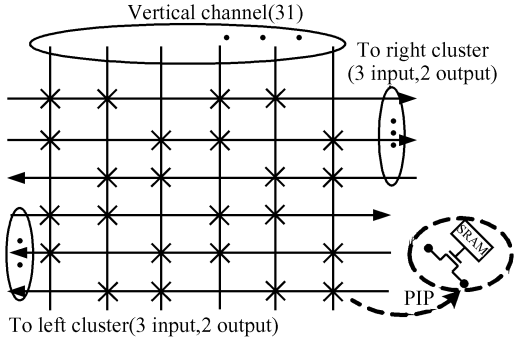

Fig. 2　Connection box

The symmetrical FPGA architecture, which benefits from the regular array, is friendly to CAD algorithms, but it has a disadvantage of area inefficiency.

A typical symmetrical FPGA architecture similar to Fig. 1 was designed by us in 2004[4]. Its routing channel width is 31. A connection box in the vertical channel, shown in Fig. 2, includes vertical channels, input paths, output paths, and programmable interconnect points (PIPs). PIP consists of an nMOS pass transistor and an SRAM, which controls the pass transistor. There are 5 paths from CB to cluster on both sides of the CB. The paths on each side include 3 input paths and 2 output paths. A frequency of connectivity of 2/3 is chosen in the design (FC = 2/3). Each input path thus has 21 PIPs, but just one PIP works in the input path when in use, so the utilization of PIPs is less than 5%. As a result, the inefficient CBs occupy about 40% of the total chip area in the prior FPGA. To make an improvement, a new hybrid FPGA architecture is proposed in this paper, which utilizes existing CBs for implementing logic functions.

2.2　A new hybrid FPGA architecture

2.2.1　Logic tile

The HFA tile consists of CBs, a cluster, and a local interconnection. Figure 3 shows a top view of the HFA cluster and local interconnection between the cluster and the left CB. An input multiplexer array routes the 12 input signals (4 CBs) and 8 feedback signals to the slices. Detailed architecture is shown in Fig. 6.

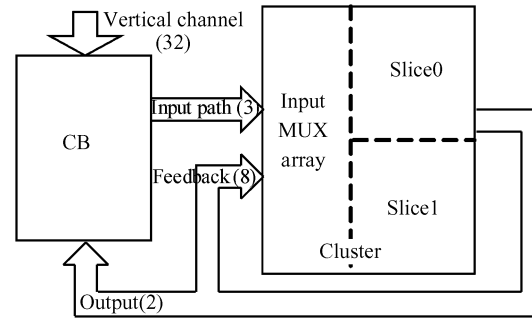

Fig. 3　Interconnections between CB and cluster

2.2.2　Wire-and CB architecture

In the new hybrid FPGA architecture, the logic tiles can be configured as either PLAs or LUTs. In the PLA mode, CBs are employed to implement the AND plane, and LUTs are used to realize the OR plane. While in the LUT mode, it can be configured as LUT, which is the same as general LUT-based FPGA.

Figure 4 shows the CB structure used in the hybrid FPGA. Its channel width is 32, close to the prior FPGA. A fully-populated CB is chosen for PLA implementation (FC = 1). A 2∶1 MUX with an inverse input is used because both the input and its inverse are needed simultaneously in the AND plane of PLA. Thus in the PLA mode the PLA inputs are 16, half of the channel width. Signals from the routing channel can be programmably connected to the cluster inputs. All the connected signals from channels to an input path are wire-and connected and regarded as a product term.

Detailed architecture of the wire-and CB is

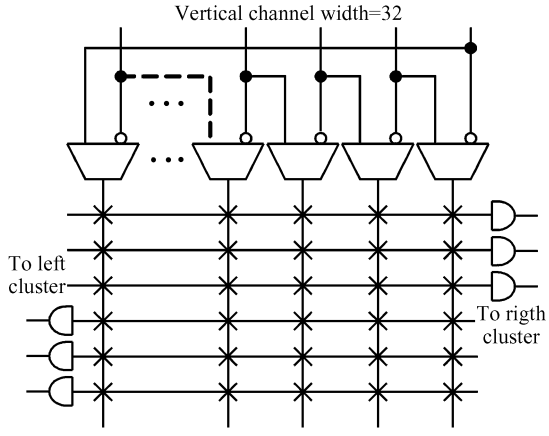

Fig. 4 Wire-and CB architecture

shown in Fig. 5. A pull-up resistor is used in each wire-and line. Two serial connected nMOS pass transistors draw the wire-and line to the ground when the SRAM and channel input are both at logic "1". The SRAM should be kept at logic "0" when the related channel input is not used. If the CB works in the LUT mode, the channel input signal is routed through the inverse input of the MUX and the corresponding SRAM is set to logic "1", while other SRAMs in the same wire-and line are kept at logic "0". In contrast to the LUT mode, several SRAMs of the same wire-and line may be set to logic "1" when the CB is in the PLA mode.

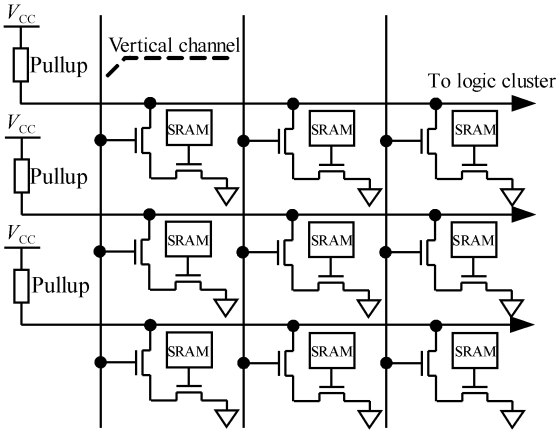

Fig. 5 Detailed CB architecture

2.2.3 HFA cluster architecture

The basic building block of the HFA cluster is the logic slice. A slice includes two LUT4s, one LUT2, carry logic, F5 MUX, F6 MUX, and two multi-purpose registers. The output from the LUT in each slice drives both the cluster output and the D input of the flip-flop. Each HFA cluster contains two slices, as shown in Fig. 6. Figure 7 shows a more detailed view of a single slice.

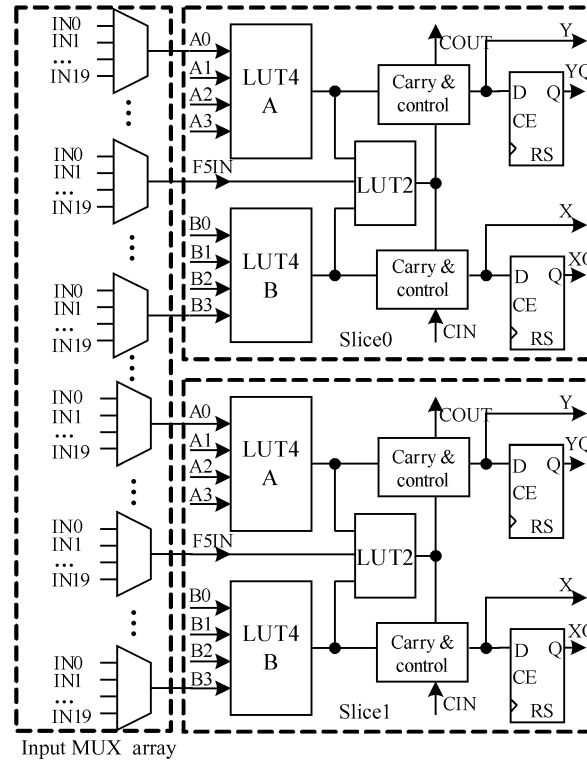

Fig. 6 Hybrid FPGA logic cluster

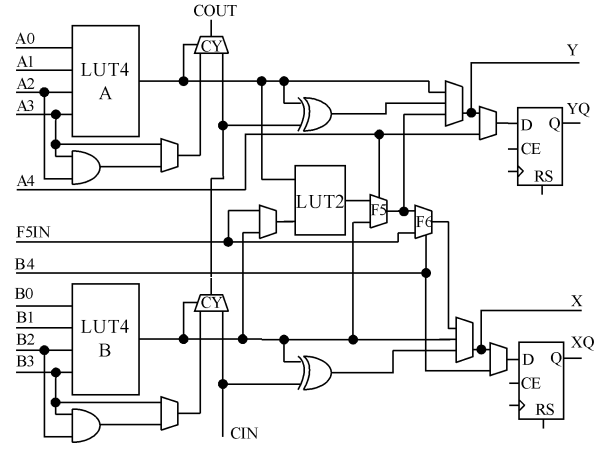

Fig. 7 Logic slice detailed architecture

In the LUT mode, the F5 MUX and F6 MUX in the cluster combine all four LUT4 outputs. This combination provides either a LUT6 that can implement any 6-input function, an 8 : 1 MUX, or selected functions of up to 12 inputs.

Dedicated carry logic provides fast arithmetic carry capability for high-speed arithmetic func-

tions. The arithmetic logic includes an XOR gate that allows a 2bit adder to be implemented in a slice. In addition, a dedicated AND gate improves the efficiency of multiplier implementation.

The storage elements in the HFA slice can be configured either as edge-triggered D-type flip-flops or level-sensitive latches. In addition to clock and clock enable signals, each slice has synchronous and asynchronous set and reset.

The LUT2 in the slice was specially designed to implement the OR plane of the PLA. In the PLA mode, LUT4A and LUT4B were both configured as 4-input OR gates, and LUT2 was then used as 2-input OR gate. As shown in Fig. 8, an 8-input OR gate can be implemented in an HFA slice (Fig. 8(a)). Combining two slices in a cluster with F5IN allows a 12-input OR gate to be implemented in a cluster (Fig. 8(b), only 12 input signals of a cluster).

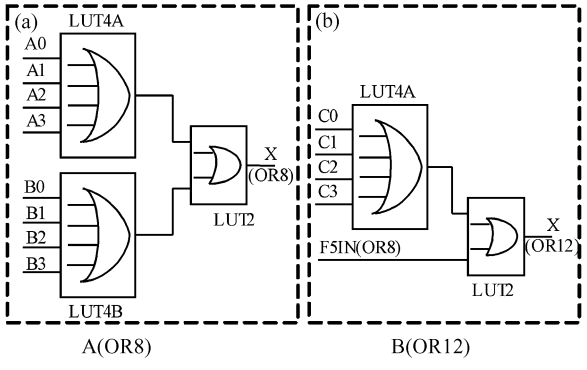

Fig. 8　Logic slice in PLA mode

A cluster and related CBs can be configured as a PLA with 16 inputs, 12 product terms and up to 4 outputs. Additionally, the HFA cluster can also be configured as a multi-input NAND gate. Then single-input product terms can be merged into one multi-variable product term[1].

The new hybrid FPGA architecture in this paper can be classified as an AND-LUT array. It can implement not only AND-OR arrays, but also AND-XOR, NAND-NAND and many other logic functions when needed. For this matter, the architecture provides high flexibility and benefits to improve logic density.

2.3　Analysis of the new architecture

The most significant improvement of the new architecture is the CB structure compared to symmetrical FPGA architecture[4]. As shown in Fig. 5, one more nMOS pass transistor was added in every crossbar connector. But the additional pass transistor hardly affects the CB layout area because the two serial nMOS pass transistors share an active area. Although the hybrid CB increases about 1/3 PIPs, the CB area is mostly determined by channel width. Thus the HFA just has a slight area increase compared to prior symmetrical FPGA for the same array size.

The PLA implementation was compared in three architectures, as shown in Table 1: new hybrid architecture (HFA), Kaviani's hybrid FPGA architecture (PLAB[1]), and Xilinx's Virtex II architecture (Virtex II CLB[3]). "Pterms" represents the product terms of the PLA. There are three kinds of PLA modes of the HFA. The first has 4 outputs with 4 Pterms per output; the second has 2 outputs with 8 Pterms per output; and the last has 2 outputs, one output with 12 Pterms, and the other with 4 Pterms. The three PLA modes can be mixed arbitrarily in application. Obviously, the new HFA is more flexible than the others.

Table 1　Comparison of PLA realizations

Architecture	Input	Pterm	Pterm per output	Output
HFA	16	12	4	4
			8	2
			12 + 4	2
PLAB[1]	16	10	10 + 6 + 3	3
Virtex II CLB[3]	16	2	2	1

3　Experimental results

The MCNC benchmarks in the format of BLIF were optimized by SIS. PLAMap[5] was then used to map the benchmarks to PLAs with 16 inputs, 12 product terms and one output. The inputs and product terms of every PLA are calculated. Those inefficient PLAs with fewer inputs are converted to LUTs. For convenience, it is defined that a LUT4 costs one area unit, and a PLA with 8 product terms needs two LUT4s, costing 2 area units.

Table 2 shows the method of area estimation. The first column illustrates the inputs of the PLA. "PLA, 2" in the table denotes PLA implementation and costs 2 area units. If there are no more than four inputs, the PLA, which is mapped by

PLAMap, will be replaced by one LUT4 no matter how many Pterms it has, and the cost is one area unit. When there are five inputs, PLA implementation has lower area cost if there are no more than four Pterms and LUT implementation has lower area cost if there are more than four Pterms. When there are more than five inputs, PLA implementation has the best area efficiency.

Table 2 Area estimation

Input	Pterm≤4	4<Pterm≤8	Pterm>8
≤4	LUT1		
5	PLA1	LUT2	
≥6	PLA1	PLA2	PLA3

For fair comparison, the optimized benchmarks were mapped to LUT4s for symmetrical FPGA architecture by Flowmap[6]. The area cost of LUT4 is the same as that in HFA. As shown in Table 3, the HFA architecture achieves 46% area gain on average. Most of the benchmark circuits benefit from HFA except for C499. C499 is a 32bit single error correcting circuit. It contains many multi-input XOR gates. Pterms exponentially increase according to inputs when XOR gates are mapped to an AND-OR array. As a result, PLA is less efficient than LUT for some multi-input XOR gates.

Table 3 Area comparison of mapping results

Benchmark	Symmetrical FPGA	Hybrid			Gain/%
	LUT4	LUT	PLA	Area	
alu4	333	21	110	253	24
apex6	396	7	103	173	56
C499	92	24	32	128	−39
cse	99	0	17	39	60
Des	1578	394	269	857	46
frg2	457	125	66	254	44
i2	72	0	27	58	19
i7	207	3	64	103	50
s820	144	2	29	51	64
s1488	322	0	38	88	72
term1	81	4	21	57	30
x3	353	10	100	164	54
Average					46

Additionally, the mapping results of HFA were pessimistic for two reasons. First, PLAMap cannot support the architecture in which the number of Pterms is not equal to the number of Pterms per output. The three kinds of PLA implementations of the HFA in Table 1 cannot be fully utilized. Thus in this work a PLA with 16 inputs, 12 product terms and one output was chosen to evaluate the new architecture. Second, the results were obtained by using only an AND-OR array, and the additional benefits brought by the flexibilities of AND-LUT architecture have not been evaluated yet. Thus a good technology mapper that targets the HFA could increase the gain.

4 Conclusions

In this paper a new hybrid FPGA architecture has been developed, which can be classified as an AND-LUT array. It achieves flexibility of architecture and good logic density, i. e., its logic tiles can be configured as either PLAs or LUTs with only slight area penalty. Experimental results show that the chip area was reduced by 46% using this hybrid architecture.

References

[1] Kaviani A, Brown S. The hybrid field programmable architecture. IEEE Design and Test, 1999, (12): 74

[2] Altera Corporation. APEX20K programmable logic device family data sheet, 2004

[3] Xilinx Corporation. Virtex-II platform FPGAs complete data sheet, 2005

[4] Wang Yuan. 0.18μm FPGA system design and implementation. Master Thesis, Fudan University, 2005 (in Chinese) [王元. 0.18μm FPGA 系统结构设计与实现. 复旦大学硕士学位论文, 2005]

[5] Chen D, Cong J, Ercegovac M, et al. Performance-driven mapping for CPLD architecture. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 2000: 39

[6] Cong J, Ding Y. FlowMap: an optimal technology mapping algorithm for delay optimization in LookUp-table based FPGA designs. IEEE Trans Comput Aided Design, 1994, (13): 12

一种基于 AND-LUT 的混合 FPGA 结构

陈利光　来金梅†　童家榕

（复旦大学微电子系，上海　200433）

摘要：提出了一种混合 FPGA 新结构——新颖的 AND-LUT 阵列结构.其创新之处在于由可编程逻辑簇(Cluster)和相关的连接盒(CB)组成的可编程逻辑单元片(Tile)可以根据应用需要灵活地配置成 PLA 或 LUT,前者较适合于高扇入逻辑,后者较适合于低扇入逻辑.因此,结合两者优点的新颖 AND-LUT 阵列结构在实现各种输入的用户逻辑时都能保持很好的逻辑利用率.MCNC 电路测试结果进一步表明,同一逻辑电路在文中提出的混合 FPGA 新结构中实现与在基于 LUT 的对称 FPGA 结构中实现相比,面积平均可节省 46%,因而大大提高了 FPGA 器件的逻辑利用率.

关键词：混合 FPGA 结构；AND-LUT 阵列；与或阵列；可编程逻辑阵列；查询表
EEACC：1130；1130B
中图分类号：TN752　　**文献标识码**：A　　**文章编号**：0253-4177(2007)03-0398-06

† 通信作者.Email:jmlai@fudan.edu.cn
2006-08-03 收到,2006-10-10 定稿