Architecture Evaluation for FPGAs with Embedded Memory Blocks

Jason Cong and Songjie Xu
Computer Science Department
University of California, Los Angeles, CA 90095
cong@cs.ucla.edu sxu@cs.ucla.edu
http://cadlab.cs.ucla.edu/˜cong http://cadlab.cs.ucla.edu/˜sxu

Abstract

As the capacity of field programmable gate arrays (FPGAs) increases, on-chip memories have become an essential component to provide high degree of on-chip integration. Modern FPGAs, such as Actel MX [27] and DX FPGAs [26], Altera FLEX10K, FLEX10KE [29] and APEX20K device family [30], and Vantis VF1 series FPGAs [28], provide embedded memory blocks (EMBs) to be used as on-chip memories. Compared with the distributed on-chip memory implementation, as that in Xilinx XC4000 series FPGAs, the dedicated EMBs result in denser memory implementation, but require the FPGA vendors to partition the chip for seperate logic and memory implementation. If a circuit design does not consume all of the available EMBs as on-chip memories, the chip area devoted to the unused memories could be completely wasted. However, it was shown that uncommitted EMBs can be efficiently used to implement logic for area minimization [1] [2]. [1] also guarantees that the circuit delay does not increase when using EMBs to minimize the circuit area. In addition, it was shown in [6] that EMBs can be taken as large look-up tables (LUTs) and used to effectively minimize the circuit delay. With the algorithms developed in [1], [4] and [6], we present in this paper the studies on FPGA embedded memory block architecture evaluation. We shall explore how the sizes and the number of available embedded memory blocks affect the circuit delay and area when these on-chip memories are used to implement logic functions. Extensive experiments and analysis are presented according to different design requirements, physical implementations, and the combination of several key design parameters within reasonable ranges. A number of interesting results are obtained. For example, our experimentation shows that EMBs with 512 memory bits lead to the best performance and area improvement in a 4-LUT based FPGA.

1 Introduction

As FPGAs tend to implement large systems, on-chip memories have become an essential component in high-density FPGAs. There are two major types of FPGA on-chip memory implementations: *distributed* or *fine-grained* on-chip memory, as that in Xilinx XC4000 series FPGAs [33], and *dedicated* or *course-grained* on-chip memory, as that in Actel MX [27] and DX FPGAs [26], Altera FLEX10K, FLEX10KE [29] and APEX20K device family [30], and Vantis VF1 series FPGAs [28].

The dedicated on-chip memory results in denser memory implementations compared with the distributed memory, but requires the FPGA vendors to partition the chip for seperate logic and memory implementation. Figure 1 shows the embedded memory blocks (EMBs) in Altera FLEX10K and FLEX10KE devices, which are all located in the middle of the chip, with one EMB in each row of logic array blocks (LABs)[1]. APEX20K devices (see Figure 2) contain an EMB[2] in each MegaLab structure that consists of 16 LABs[3]. In Vantis VF1 series FPGAs (see Figure 3), the embedded memory is implemented as two columns of memory blocks that run the full length of the FPGA device. Two columns of Super Variable-Grained Blocks (Super VGBs) run between the memory columns, and additional columns of Super VGBs are outside the memory columns.

For dedicated FPGA on-chip memory, if a circuit design does not consume all the available EMBs as on-chip memories, the chip area devoted to the unused memories could be completely wasted. However, in [1] and [2], it was shown that uncommitted embedded memory blocks can be efficiently used to implement logic for area minimization. At the same time of using EMBs to minimize the circuit area, the EMB_Pack algorithm proposed in [1] also guarantees that the circuit delay will not increase. Furthermore, the two delay-oriented technology mapping algorithms, BinaryHM and CN-HM, presented in [6], can take EMBs as large LUTs to significantly reduce the circuit delay.

Early FPGA logic block architecture studies, such as those presented in [18] and [19], focus on FPGAs with homogeneous LUTs. Recently, architecture studies of the EMBs for memory implementations have been proposed in [14],

[1] Each LAB in FLEX10K and FLEX10KE devices consists of eight logic elements (LEs, which are 4-LUTs).

[2] In APEX20K devices, EMB is called ESB as each ESB can be configured to implement product-term logic, LUT-based logic or three types of memory: dual-port RAM, read-only memory (ROM), or content addressable memory (CAM).

[3] Each LAB in APEX20K devices consists of ten LEs or 4-LUTs

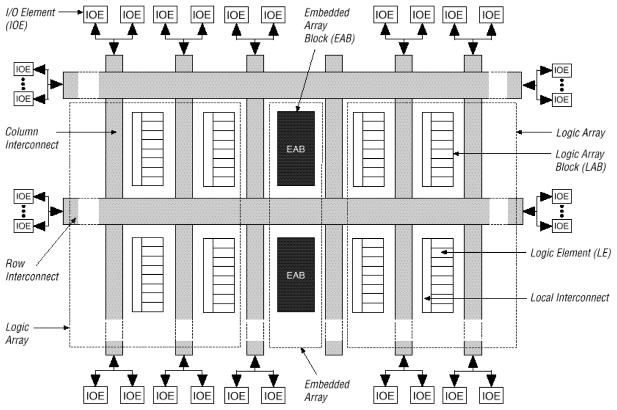


Figure 1: Altera FLEX10K and FLEX10K device block diagram. (Courtesy of Altera Corporation.)

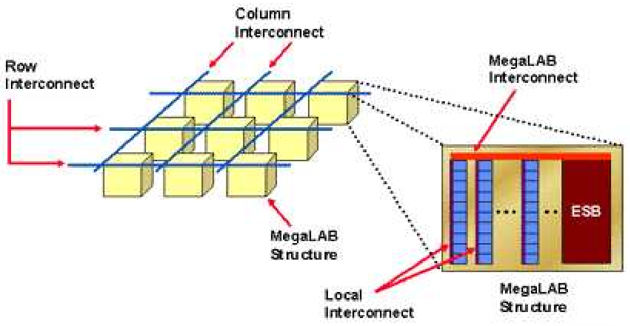


Figure 2: Altera APEX20K MegaLab and FastTrack interconnect routing structure. (Courtesy of Altera Corporation.)

[15] and [16]. The recent study in [17] also showed how the depth, width and configuration flexibility of the EMBs affect their ability to implement logic from the area minimization point of view. However, minimizing the circuit area using EMBs without considering the potential delay increase may not be practical in real circuit design.

Considering the full functionality of EMBs on both area and delay minimization, we present in this paper the architecture study for FPGA embedded memory blocks for both area and delay minimization. We shall explore how the sizes and the number of EMBs available have effect on the overall circuit delay and area minimization. The EMB_Pack algorithm [1] is used to evaluate the area minimization capability of various types of EMBs under the constraint that the circuit delay will not increase (Section 4). To evaluate the EMB architecture for delay minimization, we shall apply HeteroMap, proposed in [4], under the assumption that there is no resource limitation for EMBs, to obtain the minimum circuit delay one could expect from using EMBs and the corresponding EMBs resources needed to achieve this minimum delay (Section 5). We shall also perform BinaryHM and CN-HM [6] for EMBs with resource constraint to

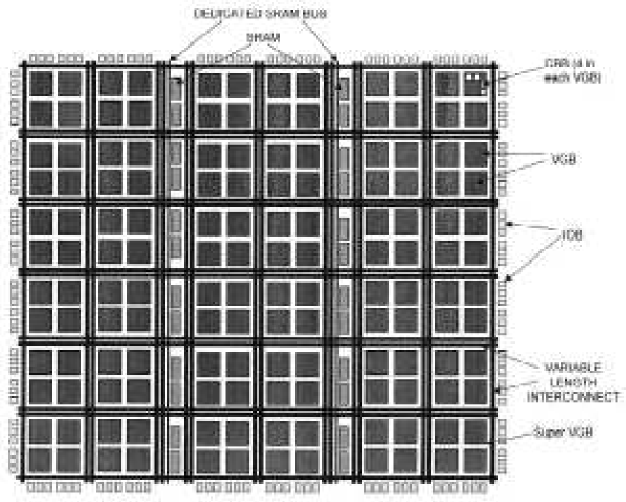


Figure 3: Vantis VF1 Family Architecture. (Courtesy of Advanced Micro Devices.)

see their delay minimization capability (Section 6).

The remainder of this paper is organized as follows. Section 2 gives the architecture assumption and problem formulation. Section 3 introduces the architecture evaluation environment, including algorithm selection, benchmark selection and design parameter set-up. Section 4 presents the EMB architecture evaluation for area minimization based on MFFC and MFFS statistics and a set of results achieved by EMB_Pack [1]. Section 5 presents the architecture study for delay minimization with EMBs achieved by HeteroMap [4] under the assumption of no resource limitation. Section 6 presents the EMB architecture evaluation for delay minimization based on a set of results achieved by BinaryHM and CN-HM [6]. Section 7 concludes the paper.

2 Architecture and Problem Formulation

As an assumption, an LUT-based FPGA with EMBs consists of an array of K-LUTs and r EMBs which can be configured to implement any function of α_1-input β_1-output, α_2-input β_2-output, …, or α_c-input β_c-output. The area ratio of a K-LUT *vs.* an EMB is $a_1 : a_2$. The delay ratio of a K-LUT *vs.* an EMB is $d_1 : d_2$[4].

In our architecture study, the following characters about EMBs are examined.

1. EMB input size I_{emb}: the address input size of one EMB when the EMB data output has width of one. If

[4] The delay of an EMB of specific size depends on the configuration (α_1-input β_1-output, α_2-input β_2-output, …, or α_c-input β_c-output) to be used. We shall consider the worst case delay among all the configurations.

the number of memory bits in one EMB is denoted by M_{emb}, then $M_{emb} = 2^{I_{emb}}$.

2. EMB number N_{emb}: the number of EMBs available to a circuit design. Circuit designs of different sizes should acquire different number of EMBs available, as larger circuits tend to use larger devices that provide more EMB resources than the smaller ones. In addition, to make a fair comparison among different EMB input sizes I_{emb}, for the same circuit, smaller number of EMBs of larger size should be provided than EMBs with smaller size. To obtain a uniform measurement on the number of EMBs of different input sizes I_{emb} available to circuits of different sizes, the character N_{emb} is determined by the ratio of memory bits dedicated to the normal K-LUTs versus those dedicated for EMBs, denoted as R_{lm}. For any K-LUT mapped circuit N, the number of EMBs available, N_{emb}, is determined by the following equation,

$$R_{lm} = \frac{n \times 2^K}{N_{emb} \times 2^{I_{emb}}} \tag{1}$$

where n is the number of K-LUTs in N. Therefore[5],

$$N_{emb} = \frac{n \times 2^K}{R_{lm} \times 2^{I_{emb}}}. \tag{2}$$

3. EMB configuration flexibility C_{emb}: a collection of all the feasible configurations of one EMB. Each configuration is characterized by two numbers, D and W, in the form of $D \times W$, where D is the input size of the configured EMB, W is its correspnding output width, and $2^D \times W = M_{emb}$. In this paper, we assume that all the EMB configurations satisfying $D \geq W$ are available. From the viewpoint of EMBs as logic implementation media, having $D < W$ may not be beneficial, and therefore is less interesting for architecture evaluation.

Table 1 shows the EMB characters, I_{emb}, M_{emb}, R_{lm} and C_{emb}, for Altera FLEX10K, FLEX10KE [29] and APEX20K device family [30], and Vantis VF1 series FPGAs [28]. For the different parts in the same device family of FLEX10K, APEX20K and VF1 FPGAs, the value of R_{emb} remains the same. When an FPGA with EMBs is to be designed, designers should first determine the value of R_{lm} based on the application of the FPGA to be developed and other design requirements. With the specific R_{lm}, designers could start evaluating different EMB characters as depicted above for delay and/or area optimization. In this paper, we shall evaluate the EMB architectures for a variety of R_{lm} values, which is detailed in Section 3.3. As what we assume in our archtecture study for the EMB configuration flexibility C_{emb}, FLEX10K, APEX20K and VF1 FPGAs provide all the EMB configurations satisfying $D \geq W$. As the variation of FLEX10K devices, FLEX10KE FPGAs adopt EMBs with 4096 memory bits, which enables the implementation of a 5×4 multiplier in one EMB. This family has a variety of R_{lm} values. FLEX10KE devices provide the 8×16 configuration, which enables the efficient implementation of the 256×16 RAM.

Commercial FPGA	I_{emb}	M_{emb}	R_{lm}	C_{emb}
FLEX10K	11	2048	1.5	$\{11 \times 1, 10 \times 2, 9 \times 4, 8 \times 8\}$
FLEX10KE	12	4096	1.125 1.625 2.375 3.25	$\{11 \times 2, 10 \times 4, 9 \times 8, 8 \times 16\}$
APEX20K	11	2048	1.25	$\{11 \times 1, 10 \times 2, 9 \times 4, 8 \times 8\}$
VF1	7	128	6	$\{7 \times 1, 6 \times 2, 5 \times 4\}$

Table 1: EMB characters of commercial FPGAs.

Based on these EMB characters, we shall explore how the sizes (I_{emb}) and the number of EMBs available (N_{emb}) have effect on the overall circuit delay and area minimization.

3 Architecture Evaluation Environment

In this section, we shall state the alogrithm selection for EMB architecture evaluation (Section 3.1) for both area and delay optimization, the benchmark set selection (Section 3.2), and the parameter and area/delay model set-up (Section 3.3).

3.1 Algorithm Selection

To evaluate the EMB architecture for area and delay optimization, we make use of the following algorithms.

1. EMB_Pack [1]: The first objective of our EMB architecture study is to evaluate the area minimization capability of various types of EMBs under the constraint that the circuit delay will not increase. Therefore, we choose the EMB_Pack algorithm as post-mapping processing, which minimizes the circuit area in terms of the number of K-LUTs after K-LUT technology mapping while maintaining the circuit delay.

2. HeteroMap [4]: HeteroMap is the first polynomial-time delay-optimal algorithm for FPGAs with hetero-

[5] All the available EMBs whose number is determined by Eq. 2 are considered to be available for logic implementation as our focus in this paper is to use EMBs for logic function implementation.

geneous LUTs. Taking EMBs as large LUTs, HeteroMap can be applied to evaluate various EMB architectures without EMB resource constraint for delay optimization. With this evaluation, we can understand how many EMBs in general are considered to be *enough* for HeteroMap to minimize the circuit delay and the corresponding performance measurements.

3. BinaryHM and CN-HM [6]: FPGAs with EMBs can be taken as the heterogeneous FPGAs with bounded resources as only a limited number of EMBs are available for a circuit design. In [6], it was shown that the technology mapping problem for delay minimization for *heterogeneous* FPGAs with *bounded* resources is NP-Hard for general networks, but can be solved optimally in pseudo-polynomial time for trees. Two heuristic algorithms, named BinaryHM and CN-HM, are then presented for delay minimization of general networks for *heterogeneous* FPGA designs with *bounded* resources, which have been successfully applied to FPGAs with EMBs to minimize the circuit delay during and/or after technology mapping. We shall apply BinaryHM and CN-HM to evaluate the EMB architecture for delay optimization.

3.2 Benchmark Selection

In order to effectively carry out the EMB architecture study, a set of optimized large benchmarks are needed. We first apply the standard SIS optimization script script.rugged [13] on all 145 MCNC benchmarks to obtain the logic optimized circuits. The decomposition algorithms, tech_decomp [13] and dmig [7], are then used to generate all the 2-input simple gate circuits. We finally apply FlowMap [8] followed by MPack [7] and FlowPack [8] to obtain all the 4-LUT mapped circuits. Among them fourteen large benchmarks, each with more than 1000 4-LUTs, are selected for architecture evaluation. These benchmarks and their features are briefly illustrated in Table 2. Notice that for sequential circuits, the number of primary inputs ("#PI") and the number of primary outputs ("#PO") include also those qseudo-PIs and qseudo-POs generated from Flip-Flops. As some of the algorithms used to evaluate the EMB architecture, such as EMB_Pack and CN-HM, are post-mapping procedures, while others, such as HeteroMap and BinaryHM, are performed during technology mapping, we need both 2-input circuits and 4-LUT mapped circuits.

3.3 Area/Delay Models

In this section, we shall set up the ranges of the EMB characters to be evaluated, and area/delay models to guide our architecture study. The parameters, illustrated below, include the range of the EMB characters to be evaluated, the delay and the area of the K-LUT and EMBs with different sizes. In general, the ranges of the parameters are chosen based on the architecture design and application interest, which also takes into consideration the corresponding values of the commercial FPGAs (Table 1).

Circuits	2-Input Circuit		4-LUT Circuit		#PI	#PO
	Depth	#Gate	Depth	#LUT		
alu4	16	4215	7	1319	14	8
apex2	17	5151	8	2303	39	3
apex4	13	2714	6	1240	9	19
bigkey	8	8592	3	1599	487	422
clma	76	24252	24	7874	416	116
des	13	6728	6	1349	256	245
dsip	12	2537	3	1148	453	422
ex5p	15	2732	7	1057	8	63
i10	38	2673	15	1346	257	224
misex3	15	6207	7	2490	14	14
s38417	29	9350	10	3657	1492	1570
s38584.1	28	12570	9	4313	1314	1580
seq	14	4661	7	2111	41	35
too_large	14	14490	7	5524	38	3

Table 2: Benchmark feature description.

1. I_{emb}: The EMB input size I_{emb} = 7, 8, 9, 10, 11, 12, 13, 14, which means that $M_{emb} = 2^7, 2^8, 2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}, 2^{14}$.

2. R_{lm}: The value of R_{lm} = 0.2, 0.7, 1.2, 1.7, 2.2. Thus, the number of EMBs available to a circuit design, N_{emb}, can be computed according to Eq. 2.

3. C_{emb}: As stated in Section 2, we assume that all the EMB configurations satisfying $D \geq W$ are available. Therefore, corresponding to the value of I_{emb}, the available EMB configurations are listed in Table 3.

4. d_{emb}: The value of d_{emb} is the delay of one EMB. We assume that the delay of a 4-LUT logic block is 1.0, and d_{emb} is increased by a constant value d_m with each increase of the input size. Therefore,

$$d_{emb} = 1.0 + (I_{emb} - 4) \times d_m \tag{3}$$

   where d_m could be 0.2, 0.3 and 0.4.

5. a_{emb}: The value of a_{emb} is the area of one EMB. We assume that the area of a 4-LUT logic block is 1.0, which includes LUT, FFs, buffers, MUXes, various internal and external interconnect resources, *etc.*. According to the area modeling and analysis in [22] and [21], about one fifth of the total area of an LUT-based logic block will double with the LUT input size increased by 1, and the other four fifths will grow approximatedly linearly with the LUT input size increase. Therefore, a_{emb} is obtained by

$$a_{emb} = 2^{I_{emb}-4} \times \frac{1}{5} + \frac{I_{emb}}{4} \times \frac{4}{5} \tag{4}$$

I_{emb}	C_{emb}
7	$\{7\times\ 1, 6\times 2, 5\times 4\}$
8	$\{8\times\ 1, 7\times 2, 6\times 4\}$
9	$\{9\times\ 1, 8\times 2, 7\times 4\}$
10	$\{10\times\ 1, 9\times 2, 8\times 4\}$
11	$\{11\times\ 1, 10\times 2, 9\times 4, 8\times 8\}$
12	$\{12\times\ 1, 11\times 2, 10\times 4, 9\times 8\}$
13	$\{13\times\ 1, 12\times 2, 11\times 4, 10\times 8\}$
14	$\{14\times\ 1, 13\times 2, 12\times 4, 11\times 8\}$

Table 3: EMB configuration.

With the area/delay modeling, Table 4 shows the EMB area and delay for each I_{emb}. The resulted area and delay for one EMB with I_{emb} of 7 and 11 are quite close to the reality [20]. When I_{emb} is larger than 10, the EMB area almost doubles with each increase of the I_{emb}, which is also true in the real implementation [20]. We would like to point out that our evaluation methodology can be applied to other models as well. In fact, we also tried other area/delay models. For instance, if only memory bit area is counted for both LUTs and EMBs, assuming that the area of a 4-LUT is 1.0, the area of an EMB is then obtained by

$$a_{emb} = 1.0 \times 2^{I_{emb}-4} \tag{5}$$

With the alternative area model, however, the architecture conclusions remain unchanged, which shows their insensitivity to different models. This will be demonstrated in Section 5.

I_{emb}	a_{emb}	d_{emb}		
		$(d_m = 0.2)$	$(d_m = 0.3)$	$(d_m = 0.4)$
7	3.0	1.6	1.9	2.2
8	4.8	1.8	2.2	2.6
9	8.2	2.0	2.5	3.0
10	14.8	2.2	2.8	3.4
11	27.8	2.4	3.1	3.8
12	53.6	2.6	3.4	4.2
13	105.0	2.8	3.7	4.6
14	207.6	3.0	4.0	5.0

Table 4: EMB areas and delays under our area/delay model.

4 EMB Architecture Evaluation for Area Optimization

In this section, we shall evaluate the area minimization capability of various types of EMBs under the constraint that the circuit delay will not increase, compared to the case where no EMBs are used. Section 4.1 gives a brief review of the EMB_Pack algorithm [1]. As EMB_Pack is an MFFC [9] and MFFS [12] based algorithm, and MFFC and MFFS in general are important for area minimization due to their fanout-free feature, Section 4.2 presents the detailed MFFC and MFFS statistics on the selected benchmarks, which may also be beneficial to other applications, such as area-oriented technology mapping [9] and MFFC and MFFS based clustering and partitioning [12]. A set of evaluation results obtained from EMB_Pack are presented in Section 4.3.

4.1 Review of the EMB_Pack Algorithm

EMB_Pack, as post-mapping processing, works on the mapped K-LUT network and is applicable to any LUT-based FPGA with EMBs. Given a mapped network, which in fact is a K-bounded netlist with each node as a K-LUT, EMB_Pack tries to appropriately *group* as many K-LUTs as possible into each EMB by calculating MFFCs and MFFSs in the network. MFFCs and MFFSs are chosen as the candidates of EMB implementations because the nodes covered by one MFFC or MFFS do not have fanouts outside this MFFC or MFFS. Therefore, they can be implemented in one EMB without duplication, which is beneficial for area minimization. The algorithm runs in three phases: (1) select a set of MFFCs and MFFSs in the network for possible EMB implementations; (2) those MFFCs and MFFSs with infeasible input size are cut to cater to the EMB configurations; (3) EMBs are selected from the EMB candidates obtained from phase (1) and (2). The third phase also guarantees that the circuit delay will not increase, compared with the mapped input circuit.

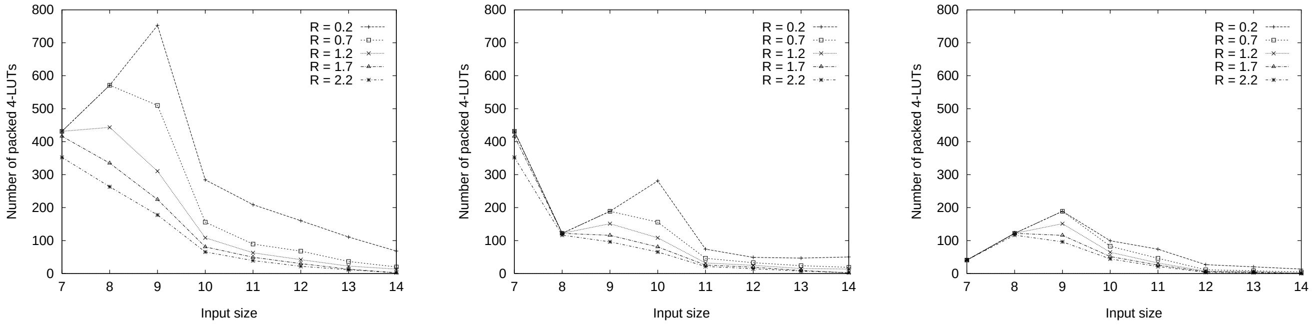
4.2 MFFC and MFFS Statistics

A set of statistical results are gathered for MFFCs and MFFSs in the circuits. Figure 11(a) shows the distribution of MFFCs on different MFFC sizes. Figure 11(b) and (c) depict the average and maximum MFFC sizes of the MFFCs with the input size from 7 to 14. Figure 12 shows the average and maximum MFFS sizes for each input size. Please refer to Appendix for the detailed information.

4.3 Area Reduction by EMB_Pack

To evaluate the area minimization capability of EMBs with different sizes under the constraint that the K-LUT mapped circuit delay will not increase, we apply EMB_Pack on 4-LUT circuits obtained from Flowmap [8] and the area minimization postprocessing (MPack [7] and FlowPack [8]). With R_{lm} = 0.2, 0.7, 1.2, 1.7 and 2.2, and d_m = 0.2, 0.3 and 0.4, Figure 4 shows the number of 4-LUTs packed into EMBs by EMB_Pack. In the rest of the paper, the symbol R in the figures represents R_{lm}.

It is clearly shown that the results are consistent for different values of R_{lm} and d_m. In general, EMBs with I_{emb} equal to 8 or 9 can pack more logic into EMBs while guaranteeing that the mapped circuit delay will not increase. Note



(a) d_m= 0.2 (b) d_m= 0.3 (c) d_m= 0.4

Figure 4: The number of 4-LUTs packed by EMBs.

that this result is somewhat different from the conclusion drawn in [17], which makes the preference of using EMBs with I_{emb} equal to 11. However, the algorithm used in [17] does not consider the potential delay increase incurred from using EMBs for area minimization.

Larger EMBs are associated with longer delays. However, if the larger EMBs can generate much better logic implementation capability than the smaller ones, they could overcome their delay increase such that more 4-LUTs of the original circuits can be packed into EMBs. On the other hand, EMB_Pack will select EMB implementations which will not increase the overall circuit delay, which means that EMBs could be used on both non-critical and critical paths under the delay constraint.

As EMBs with I_{emb} equal to 8 or 9 are able to pack more logic, it would be interesting to know the appropriate number of EMBs to be provided for maximum density of logic packing by EMBs. Figure 5 shows the number of 4-LUTs packed by one EMB ($I_{emb} = 9$) on average for different R_{lm} under the three d_m values. The EMB packing densities for $d_m = 0.3$ and $d_m = 0.4$ are exactly the same for this set of experimentation. It is shown that selecting R_{lm} larger than 0.7 would be beneficial for higher EMB packing density.

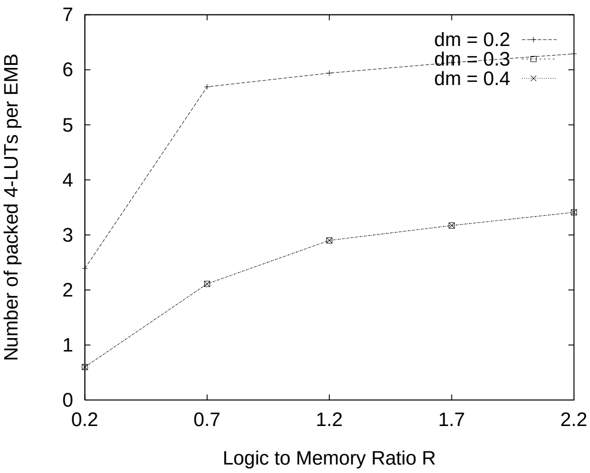


Figure 5: The number of 4-LUTs packed by one EMB.

5 EMB Architecture Study with HeteroMap

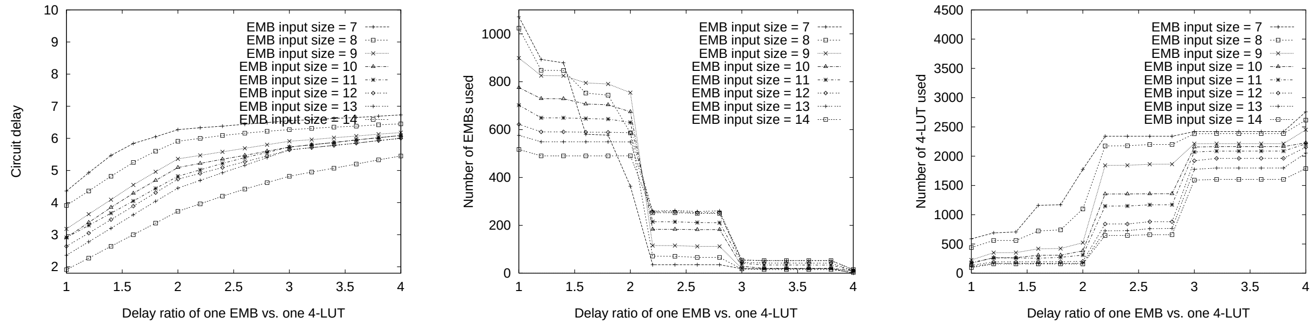
In this section, we shall evaluate the delay optimization capability of various types of EMBs under the assumption that there is no limitation on the EMB resources. Section 5.1 gives a brief review of the HeteroMap algorithm [4]. Section 5.2 presents the architecture study results obtained from HeteroMap [4].

5.1 Review of the HeteroMap Algorithm

HeteroMap [4] is a technology mapping algorithm that computes delay-optimal mapping solutions in polynomial time for heterogeneous FPGAs without resource constraints. Taking different delays of heterogeneous LUTs into consideration, HeteroMap computes the minimum mapping delay of a circuit based on a series of minimum height K-feasible cut computations at each node in the circuit. For a heterogeneous FPGA consisting of K_1-LUTs, K_2-LUTs, ..., and K_c-LUTs, HeteroMap computes the minimum delay mapping solution in $O(\sum_{i=1}^{c} K_i \cdot n \cdot m \cdot \log n)$ time for a circuit netlist with n gates and m edges.

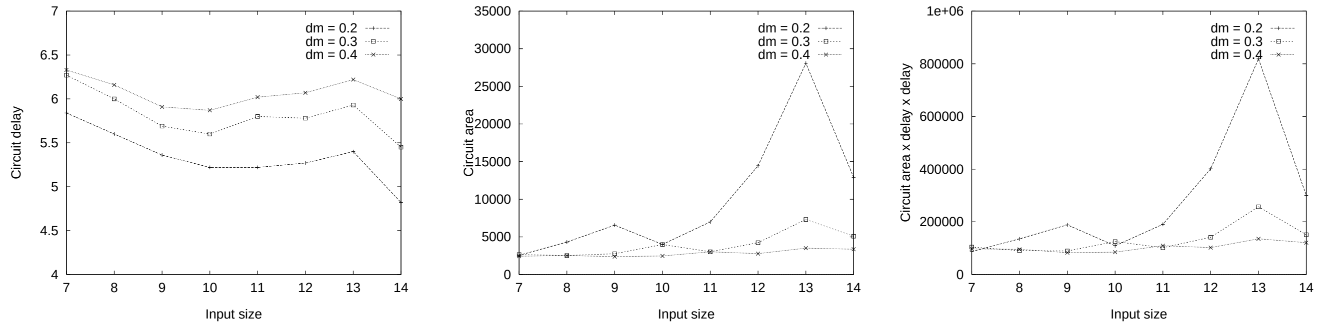
5.2 Evaluation Results with HeteroMap

We would like to see in this section that if we do not have resource limitation on the EMBs, how many EMBs are finally used in the delay optimal mapping solution for a given set of delay ratios between one EMB and a K-LUT.



(a) Circuit delay **(b) Number of EMBs used** **(c) Number of 4-LUTs used**

Figure 6: Circuit delay, number of EMBs and number of 4-LUTs used in HeteroMap mapping solution.



(a) Circuit delay **(b) Circuit area** **(c) AD^{2}measurement**

Figure 7: Circuit delay, area and AD^2 measurement of HeteroMap mapping solutions for EMBs of different sizes.

Taking each EMB as a large LUT (I_{emb}-LUT), the HeteroMap algorithm is performed to map the circuits into 4-LUTs and EMBs based on their delay ratio, which is set to be from 1 all the way to 4. Figure 6(a) to (c) illustrate the circuit delay, number of EMBs and 4-LUTs used respectively. It is interesting to see that the results are consistent with one another for different values of I_{emb}. Figure 6(b) shows that the number of EMBs used drops very fast for I_{emb} equal to 7 and 8 with respect to the delay ratio increase.

If the delay for each size of EMB is known, Figure 6(a) can tell the minimum circuit delay one could expect from using EMBs, and Figure 6(b) shows exactly the number of EMBs used by HeteroMap to achieve this minimum circuit delay. Such a comparison could be made among all the EMBs with I_{emb} from 7 to 14.

With d_m equal to 0.2, 0.3 and 0.4, Figure 7(a) shows the circuit delay obtained by HeteroMap for different I_{emb} values, which indicates the minimum circuit delay one could expect from using EMBs. I_{emb} of value 10 generates the best circuit delay with HeteroMap.

Figure 7(b) shows the circuit area of the mapping solution according to the area model specified in Eq. 4. Figure 7(c) shows the overall circuit performance comparison for EMBs of different sizes based on the AD^2 metric [23], which represents $area \times delay^2$. The reason why we do not use the metric of AD, which was considered to be a more straightforward metric, is because doubling the area usage does not result in two times improvement on the circuit performance. As a result, AD metric tends to exaggerate the effect of area on the overall circuit performance. The metric of AD^2 puts more weight on the delay factor and we intend the area factor to be used only for breaking the delay ties, but not a real trade-off factor with delay. In general, for delay minimization without EMB resource constraint, I_{emb} should take value of 9 to 11 to achieve better overall circuit performance.

In order to see the effect of different area models on the evaluation conclusions, an alternative area model, specified in Eq. 5, is applied, where only memory bit area is counted for both LUTs and EMBs. Figure 8 shows the corresponding AD^2 measurement of the mapping solutions obtained from HeteroMap based on this area model. Comparing Figure 7(c) and Figure 8, the Y-axis scale in Figure 7(c) is smaller than that in Figure 8. However, the performance comparison based on the AD^2 metric does not change among EMBs of different input sizes I_{emb}.

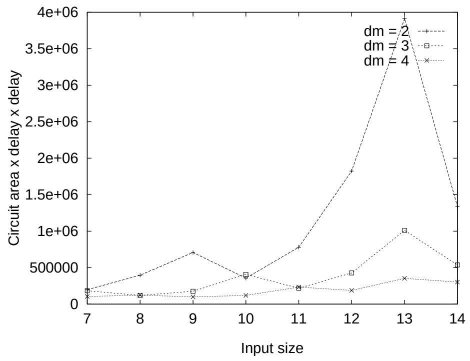


Figure 8: Circuit AD^2 performance obtained from HeteroMap for EMBs of different sizes under the area model specified in Eq. 5.

6 EMB Architecture Evaluation for Delay Optimization

In this section, we shall evaluate the delay optimization capability of various types of EMBs with resource constraint. Section 6.1 gives a brief review of BinaryHM and CN-HM, proposed in [6]. Assuming that there are only a limited number of EMBs, a set of evaluation results obtained from BinaryHM and CN-HM are presented in Section 6.2.

6.1 Review of BinaryHM and CN-HM

BinaryHM and CN-HM [6] are two heuristic mapping algorithms for heterogeneous FPGAs with bounded resources, and can be directly applied to FPGAs with EMBs. BinaryHM is an extension of the HeteroMap algorithm. Assuming that there are two types of LUTs in one FPGA chip, K_1-LUT without resource constraint and K_2-LUT with resource constraint, BinaryHM operates on the *original unmapped* circuit and performs binary search on the delay ratio of K_1-LUT *vs.* K_2-LUT such that HeteroMap can eventually obtain the mapping solution with feasible number of K_2-LUTs used and the circuit mapping delay minimized. Instead, CN-HM takes the *mapped* circuit with each node as a K_1-LUT and performs binary search on the minimum delay of the circuit to get one delay target at each time. For each delay target, CN-HM identifies *critical* nodes and again performs binary search on the delay ratio of K_1-LUT *vs.* K_2-LUT such that a HeteroMap-like approach can decide whether the delay target is achievable or not.

6.2 Evaluation Results with BinaryHM and CN-HM

For any R_{lm} ($R_{lm} = 0.2$, 0.7, 1.2, 1.7 and 2.2) and d_m ($d_m =$ 0.2, 0.3 and 0.4), we perform technology mapping using the available EMBs with BinaryHM on the original unmapped circuits for delay minimization. Figure 9 shows the circuit delays obtained from BianryHM. With different d_m, the delay minimization capability for each type of EMBs under the specific value of R_{lm} is about the same. However, different R_{lm} values generate fairly deviated results. In general, EMBs with I_{emb} equal to 8 or 9 perform well for all the R_{emb}.

As an alternative approach, we use CN-HM on the 4-LUT mapping circuits for delay optimization. Figure 10 shows the circuit delays obtained from CN-HM. As CN-HM is a post-mapping procedure, compared with the results generated through BinaryHM, CN-HM does not show much difference on the circuit delay with different EMB characters.

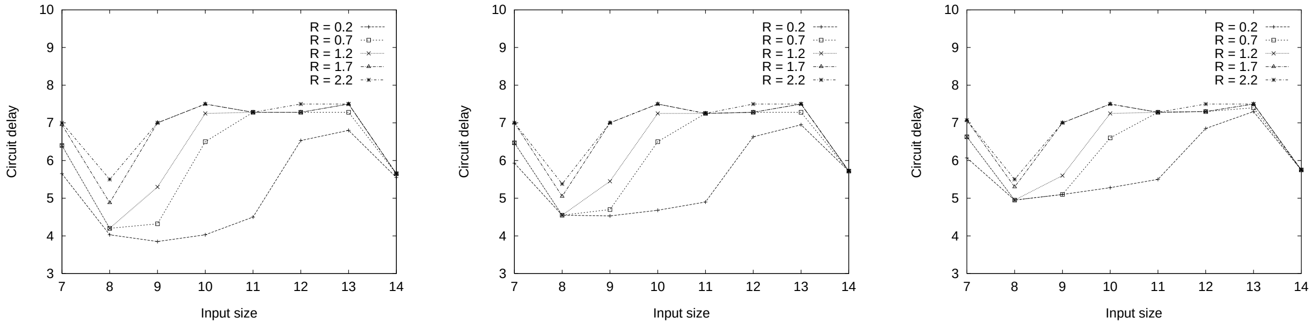
In general, EMBs with I_{emb} equal to 8 show the most promising capability to minimize the circuit delay.

7 Conclusions and Architecture Recommendation

In this paper, we applied the recently developed algorithms on FPGA EMB architecture evaluation for both area and delay optimization when the uncommitted EMBs are used to implement logic functions. A set of EMB characters have been examined, including the size, configuration flexibility and the memory bit area ratio between the normal K-LUT logic and the EMBs. Extensive experiments and analysis are presented according to different design requirements, physical implementations, and the combination of several key design parameters within reasonable ranges. Under the objective of minimizing the circuit area while preserving the delay, our experimental results show that I_{emb} of value 8 or 9 is advantageous over the others. For delay optimization without EMB resource constraint, HeteroMap results imply that the promising values for I_{emb} are 9, 10 and 11. The results obtained from BinaryHM and CN-HM suggest using I_{emb} of 8 or 9 for better circuit delay under the EMB resource constraint. In general, EMBs of memory bit size 2^9 are good for both area and delay optimization.

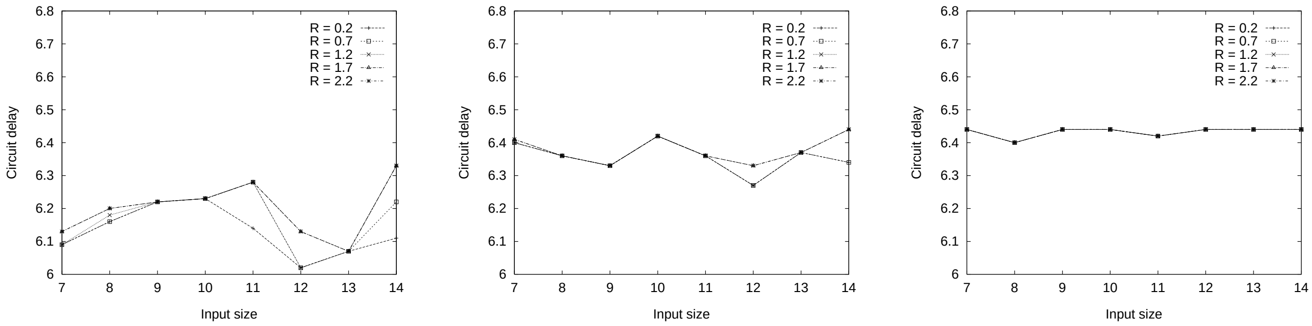
In addition, we have shown the MFFC statistics on K-LUT circuits with K from 3 to 6 in Appendix, as EMB_Pack is an MFFC and MFFS based algorithm, and MFFC and MFFS in general are important for area minimization due to their fanout-free feature.

As the future research, we shall consider the architecture evaluation on the combination of the LUT size and the EMB size for better circuit performance. We shall continue this architecture study on a larger set of large benchmarks, including industry designs, as soon as they are available. We also plan to use the EMBs on logic optimization and resynthesis for both area and delay minimization, which could be used to provide a more comprehensive architecture evaluation for FPGAs with EMBs. In addition, we are considering



(a) d_m= 0.2 **(b) d_m= 0.3** **(c) d_m= 0.4**

Figure 9: Circuit delay obtained from BinaryHM.



(a) d_m= 0.2 **(b) d_m= 0.3** **(c) d_m= 0.4**

Figure 10: Circuit delay obtained from CN-HM.

taking advantage of the generic placement and routing tool, such as VPR [24], to evaluate the location and the routing resources for EMBs.

Any architecture evaluation is algorithm dependent, which is one of the motivations for people to develop optimal or near-optimal algorithms. We believe that the algorithms used in the paper are the best choices at this moment for our purposes. However, using different algorithms could obtain different architecture conclusion. With this in mind, we are also working on developing better algorithms for FPGAs with EMBs. For instance, we may generalize the BinaryHM and CN-HM algorithm to handle EMB implementations with multiple-output configurations, which is expected to be more effective.

8 Acknowledgments

We would like to give our sincere thanks to Bryan Archell, H.T. Chang, Mike Hutton, David Mendel, Jay Schleicher and Tim Southgate for their helpful discussions. This work is partially supported by Altera Corp. and Advanced Micro Devices under the California MICRO Program and National Science Foundation Young Investigator Award MIP9357582.

References

[1] J. Cong and S. Xu, *"Technology Mapping for FPGAs with Embedded Memory Blocks"*, Proc. ACM Int'l Symposium on FPGA, Monterey, CA, Feb. 1998, pp. 179-188.

[2] S. J. E. Wilton, *"SMAP: Heterogeneous Technology Mapping for Area Reduction in FPGAs with Embedded Memory Arrays"*, Proc. ACM Int'l Symposium on FPGA, Monterey, CA, Feb. 1998, pp. 171-178.

[3] J. He and J. Rose, *"Technology Mapping for Heterogeneous FPGAs"*, Proc. ACM Int'l Symposium on FPGA, Monterey, CA, Feb. 1994.

[4] J. Cong and S. Xu, *"Delay-Optimal Technology Mapping for FPGAs with Heterogeneous LUTs"*, Proc. 35th ACM/IEEE Design Automation Conf., San Francisco, CA, June, 1998, pp. 704-707.

[5] M. R. Korupolu, K. K. Lee, and D. F. Wong, *"Exact Tree-based FPGA Technology Mapping for Logic Blocks with Independent LUTs"*, Proc. 35th ACM/IEEE Design Automation Conf., San Francisco, CA, June 1998, pp. 708-711.

[6] J. Cong and S. Xu, *"Delay-Oriented Technology Mapping for Heterogeneous FPGAs with Bounded Resources"*, Proc. IEEE Int'l Conf. on Computer-Aided Design, San Jose, CA, Nov. 1998, pp. 40-45.

[7] K. C. Chen, J. Cong, Y. Ding, A. B. Kahng, and P. Trajmar, *"DAG-Map: Graph-based FPGA Technology Mapping for Delay Optimization"*, IEEE Design and Test of Computers, Sep. 1992, pp. 7-20.

[8] J. Cong and Y. Ding, *"FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs"*, IEEE Trans. on Computer-Aided Design, Feb. 1994, Vol. 13, No. 1, pp. 1-12.

[9] J. Cong and Y. Ding, *"On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping"*, IEEE Trans. on VLSI Systems, June 1994, Vol. 2, No. 2, pp. 137-148.

[10] J. Cong and Y. Hwang, *"Simultaneous Depth and Area Minimization in LUT-Based FPGA Mapping"*, UCLA Computer Science Department Technical Report CSD-950001, January 1995.

[11] J. Cong and Y. Ding, *"Tutorial and Survey Paper — Combinational Logic Synthesis for LUT Based Field Programmable Gate Arrays"*, ACM Trans. on Design Automation of Electronic Systems, Vol. 1, No. 2, April 1996, pp. 145-204.

[12] J. Cong, P. Li, S.-K. Lim, T. Shibuya, and D. Xu, *"Large Scale Circuit Partitioning With Loose/Stable Net Removal and Signal Flow Based Hierachical Clustering"*, Proc. IEEE Int'l Conf. on Computer-Aided Design, San Jose, CA, Nov. 1997, pp. 441-446.

[13] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephen, R. Brayton, and A. Sangiovanni-Vincentelli, *"SIS: A System for Sequential Circuit Synthesis"* U.C.Berkeley Technical Report UCB/ERL M92/41, May. 1992.

[14] S. J. E. Wilton, J. Rose, and Z. G. Vranesic, *"Architecture of Centralized Field-Configurable Memory"*, Proc. ACM Int'l Symposium on FPGA, 1995, pp. 97-103.

[15] S. J. E. Wilton, J. Rose, and Z. G. Vranesic, *"Memory/Logic Interconnect Flexibility in FPGAs with Embedded Memory Arrays"*, Proc. IEEE Custom Integrated Circuits Conf., May 1996, pp. 144-147.

[16] S. J. E. Wilton, J. Rose, and Z. G. Vranesic, *"Memory-to-Memory Connection structures in FPGAs with Embedded Memory Arrays"*, Proc. ACM Int'l Symposium on FPGA, 1997, pp. 97-103.

[17] S. J. E. Wilton, *"Implementing Logic in FPGA Embedded Memory Arrays: Architecture Implications"*, Proc. IEEE Custom Integrated Circuits Conf., May 1998.

[18] J.S. Rose, R.J. Francis, D. Lewis, and P. Chow, *"Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency"*, IEEE JSSC, Vol. 25, No. 5, Oct. 1990, pp. 1217-1225.

[19] S. Singh, J. Rose, P. Chow, and D. Lewis, *"The Effect of Logic Block Architecture on FPGA Performance"*, IEEE JSSC, Vol. 27, No. 3, Mar. 1992, pp. 281-287.

[20] *"Private Communication"*, 1997, 1998.

[21] V. Betz, *"Architectures and Algorithms for Field-Programmable Gate Arrays with Embedded Memory"*, PH.D. Thesis, University of Toronto, 1997.

[22] V. Betz, *"Architecture and CAD for Speed and Area Optimization of FPGAs"*, PH.D. Thesis, University of Toronto, 1998.

[23] J. D. Ullman, *"Computational Aspects of VLSI"*, Computer Science Press, Inc., Rockville, MD, 1983.

[24] V. Betz, and J. Rose, *"VPR: A New Packing, Placement and Routing Tool for FPGA Research"*, 7th Int'l Workshop on Field-Programmable Logic and Applications, London, UK, 1997, pp. 213-222.

[25] T. Cormen, C. Leiserson, and R. Rivest, *"Algorithm"*, MIT Press, Cambridge, MA, 1990

[26] Actel Corp., *"Data Sheet: 3200DX FPGAs"*, Actel Corp., Sunnyvale, CA, 1995.

[27] Actel Corp., *"MX FPGA Data Sheet"*, Actel Corp., Sunnyvale, CA, 1996.

[28] Advanced Micro Devices, *"VANTIS VF1 FPGA Data Sheet"*, Advanced Micro Devices, Inc., Sunnyvale, CA, 1998.

[29] Altera, *"Programmable Logic Devices Data Book"*, Altera Corp., San Jose, CA, 1998.

[30] Altera, *"APEX20K Programmable Logic Device Family Advance Information Brief"*, Altera Corp., San Jose, CA, 1998.

[31] Lattice Semiconductor Corp. *"Data Sheet: ispLSL and pLSI 6192 High Density Programmable Logic with Dedicated Memory and Register/Counter Modules"*, Lattice Semiconductor Corp., Hillsboro, OR, 1996.

[32] Lucent Technologies, *"ORCA OR2C-A/OR2T-A Series FPGAs Data Sheet"*, Lucent Technologies, Inc., Allentown, PA, 1996.

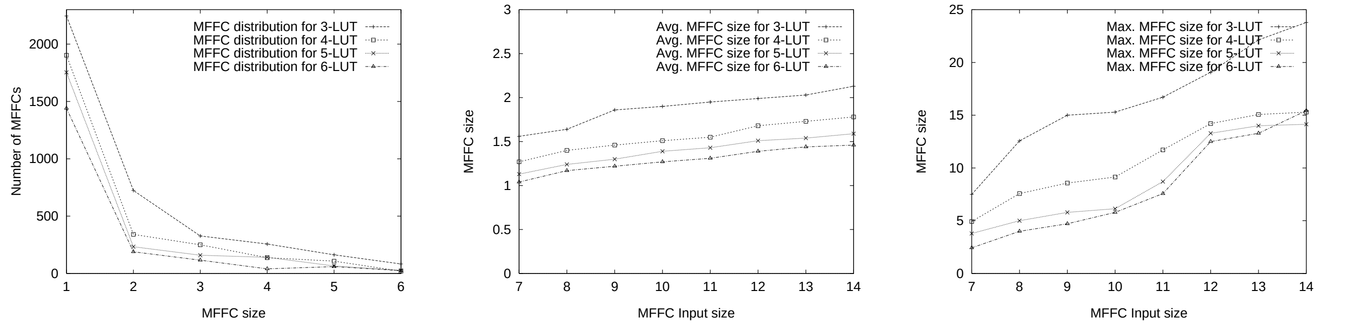
[33] Xilinx Inc., *"The Programmable Logic Data Book"*, Xilinx Inc., San Jose, CA, 1997.

Appendix: MFFC and MFFS Statistics

For a netlist N with n nodes, there are n different MFFCs since each node has its unique MFFC. Based on MFFC enumeration on the K-LUT mapped circuits with K equal to 3, 4, 5, and 6, Figure 11(a) shows the distribution of MFFCs on different MFFC sizes. We could observe that most of the MFFCs are small in size, and with the increase of the LUT input size, the MFFC size will decrease.

In addition, Figure 11(b) and (c) depict the average and maximum MFFC sizes of the MFFCs with the input size from 7 to 14. Notice that the average and maximum MFFC sizes depicted for the X-axis point of 7 represents the values for MFFCs with the input sizes smaller or equal to 7.

As stated before, for a netlist N with n nodes, there are n different MFFCs since each node has its unique MFFC. However, there are $\binom{n}{M}$ different MFFSs with root set size equal to M (called an M-root set). As we can not afford to enumerate all the MFFSs in the network, [1] presented an algorithm to compute the good MFFSs from the netlist based on the root set selection. With this MFFS selection algorithm, we generate all the feasible MFFSs according to the input size (I_{emb}) and its configurations stated in Table 3.



(a) MFFC distribution on difference MFFC sizes **(b) Average MFFC size** **(c) Maximum MFFC size**

Figure 11: MFFC statistics.

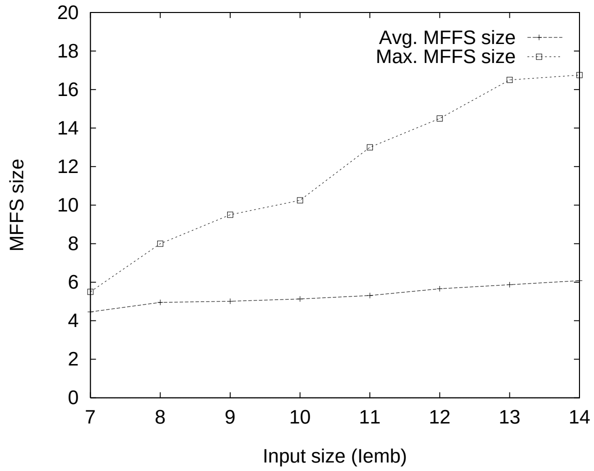


Figure 12: MFFS selection results on avg. and max. MFFS size.

Figure 12 shows the average and maximum MFFS sizes in terms of the number of 4-LUTs for each input size.