

Power-Aware FPGA Logic Synthesis Using Binary Decision Diagrams

Kevin Oo Tinmaung, David Howland, and Russell Tessier
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003

ABSTRACT

Power consumption in field programmable gate arrays (FPGAs) has become an important issue as the FPGA market has grown to include mobile platforms. In this work we present a power-aware logic optimization tool that is specialized to facilitate subsequent power-aware technology mapping. Our synthesis framework uses binary decision diagram (BDD) based collapsing and decomposition techniques in conjunction with signal switching estimates to achieve power-efficient circuit networks. The results of synthesis and subsequent power-aware technology mapping are evaluated using two distinct physical design platforms: academic VPR and Altera Quartus II. Our approach achieves an average energy reduction of 13% for Altera Cyclone II devices versus synthesis with SIS-based algebraic optimization at the cost of 11% average circuit performance if performance-optimal technology mapping is performed after synthesis. If technology mapping is tuned to achieve the same average delay for both SIS and BDD-based flows, a 3% average energy reduction is achieved by our new synthesis approach.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids

General Terms

Algorithms

Keywords

FPGA, Binary decision diagram, Dynamic power

1. INTRODUCTION

The deployment of FPGAs in a variety of portable and embedded systems has demonstrated the need for device power efficiency. This need has led to a series of FPGA power reduction approaches at both the architectural and computer-aided design levels. Most power-aware FPGA CAD techniques have been focused on technology mapping [1], placement [5], and routing [5] and use signal switching information to reduce dynamic power. Although these techniques are effective at reducing area and signal switching activity, they rely on a circuit netlist which has first been processed by logic optimization.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FPGA'07, February 18–20, 2007, Monterey, CA, USA.
Copyright 2007 ACM 1-59593-292-5/07/0002...\$5.00.

Previously, it has been shown that effective power-aware FPGA technology mapping algorithms limit signal fanout and encapsulate high-transition nets within FPGA look-up tables (LUTs) [5][13][7][1]. These techniques reduce the need for area-consuming logic replication and limit signal switches on inter-LUT paths. Ideally, a circuit netlist generated by logic optimization has nodes which exhibit low switching and low fanout. These characteristics help power-aware technology mapping evaluate a broader range of low-power mapping choices.

Our new synthesis algorithms use binary decision diagram (BDD) clustering and decomposition to minimize circuit fanout and signal switching characteristics. Prior to BDD decomposition, circuit nodes are collapsed into clusters which minimize node fanout. Subsequently, clusters are decomposed so that signal switching is minimized. Efficient decomposition is facilitated with the use of a fast statistical switching estimator which allows for the evaluation of numerous intermediate decompositions.

Our new FPGA optimized synthesis system has been integrated into a compilation flow which includes EMap [5], a power-aware technology mapping tool. Placement and routing is performed with Quartus II, a commercial system from Altera, and alternatively, with VPR, an academic place and route tool. For a series of benchmark designs, our synthesis approach shows a 13% energy reduction versus synthesis with SIS [12] for Altera Cyclone II, and an 11% energy reduction for an academic FPGA model at a fixed clock frequency. These energy reductions come at the cost of a substantial loss in achievable peak design clock frequency for a subset of designs. If technology mapping performance constraints are relaxed in the SIS compilation flow so that compilation achieves the same average clock frequency obtained with our BDD-based compilation flow, a 3% energy savings is observed for BDD-based synthesis.

This paper is organized as follows. Section 2 describes necessary background regarding BDDs and power-aware technology mapping, and related logic synthesis work. Our power-aware logic synthesis and technology mapping framework is presented in Section 3 and the experimental approach is outlined in Section 4. Section 5 provides an analysis of the experimental results. Section 6 concludes the paper and offers directions for future work.

2. Background

2.1 BDD-Based Logic Synthesis

A BDD is a rooted directed acyclic graph (DAG) representing a switching function. BDD-based logic synthesis based on reduced ordered binary decision diagrams (ROBDDs) is an effective logic optimization approach for both standard cell [14] and FPGA

technology [15]. Typically, BDD synthesis relies on functional decomposition to explore a range of both algebraic and Boolean optimizations. This step is an important part of our BDD-processing flow which also includes clustering, variable reordering, and shared subcircuit extraction steps.

An early step in BDD based synthesis involves collapsing portions of a circuit DAG into clusters. This **eliminate** procedure makes the resulting BDD network easier to evaluate and optimize. Each collapsed cluster is represented as a BDD which is subsequently decomposed. Partial collapsing helps remove logic redundancy in multi-level logic representations, such as the redundancy caused by reconvergence. In [15], it was determined that maximum fanout free cone (MFFC) based collapsing is especially effective for FPGA synthesis with BDDs. Collapsed multi-input cones often match the structure of FPGA LUTs, leading to area-efficient implementations.

Cluster generation with partial collapsing is followed by iterative BDD **decomposition** based on variable partitioning approaches. The theory of dominators [14] has been shown to be an effective basis for BDD decomposition. The dominator approach quickly identifies cuts which divide a target subcircuit into two parts. The subcircuits which result from these cuts can then be combined with either an AND, OR, or XOR function to perform original BDD functionality. Algebraic decompositions result in subcircuits which do not share input variables; Boolean decomposition allows sharing. Typically a depth-first BDD scan is performed to identify dominators. Unlike strictly algebraic synthesis tools, dominator based decomposition can identify AND/OR, AND/XOR, and multiplexer decompositions equally efficiently. Since arithmetic FPGA designs often contain numerous XOR and multiplexer operations, this flexibility is desirable for FPGA synthesis. To explore a variety of decompositions, the variables in the BDD may be reordered to expose more efficient cuts [14]. Previously, it was determined that frequent variable reorderings provide area-efficient FPGA implementations [15].

2.2 Related Work

Several power-aware technology independent synthesis algorithms have been proposed. In a paper by Iman and Pedram [4], a power-aware synthesis approach which uses algebraic decompositions is proposed. Given a Boolean function of a node, dynamic power is optimized by minimizing its load in terms of the number of product terms in the function and by minimizing the estimated switching activity of internal sum-of-product nodes. Lennard et al. [6] uses the probability distribution of don't care sets to optimize area with an eye towards power reduction. An approach presented by Roy et al. [11] transforms sum-of-product expressions into factor-tree forms and uses capacitance information of AND/OR gates to guide logic optimization. Holt and Tyagi [3] consider interconnect capacitance using a bounding box during synthesis. In a paper by Lindgren et al. [8], a power-aware BDD decomposition approach for pass transistor logic is defined. In general, these techniques support power reduction in fine-grained ASICs which are based on logic libraries. The presence of coarse-grained LUTs and logic clusters make these power-aware synthesis approaches inappropriate for FPGAs.

To effectively perform power-aware FPGA logic optimization, the goals of subsequent power-aware technology mapping must be understood. Most related mapping approaches examine the reduction of dynamic power in FPGA interconnect, where nearly

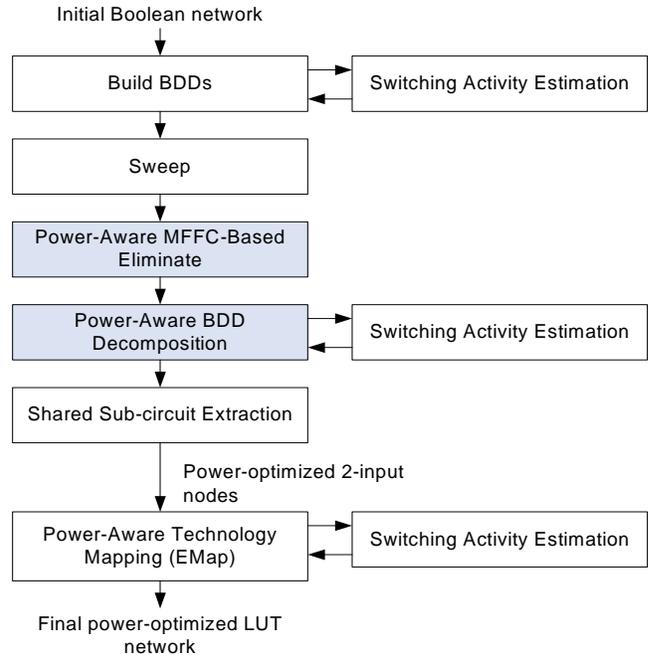


Figure 1: Power-aware logic synthesis and technology mapping flow

two-thirds of contemporary FPGA power is consumed. An early research study by Farrahi and Sarrafzadeh [2] showed that a power consumption cost metric based on net signal switching can be the basis for a LUT packing heuristic. Although this approach showed a modest improvement in circuit power (14% reduction), a net increase in LUT area was noted. An approach by Li et al [7] first maps circuits using a depth-optimal cut-based heuristic and then re-cuts logic node outputs on the critical path to reduce power. This technique resulted in a 19% power reduction when compared to the approach in [2].

Recent efforts at achieving power reduction in FPGA technology mapping focus on two specific goals: the reduction of LUT output signal switching and the minimization of logic replication. The first goal minimizes the switching activity of the nets between LUTs by encapsulating high activity nets inside LUTs. The second goal minimizes the number of LUT outputs by minimizing node duplication for performance enhancement. Anderson and Najm [1] use a depth-first, cut-based mapping to determine an initial mapping. This work focuses on eliminating node replications along non-critical paths to reduce power. A cost function based on LUT area, depth, and power is derived to allow for mapping tradeoffs. The authors note an additional 10% power reduction if optimal technology mapping depth is relaxed. A similar mapping approach is applied in EMap [5]. The EMap approach uses both signal switching and signal fanout as cost metrics used to encapsulate logic in LUTs. Our power-aware FPGA synthesis approach follows up on these recent efforts in the logic optimization step by optimizing node fanout, output switching, and area during partial collapsing and decompositions.

3. Power-aware Logic Optimization Flow

Our BDD-based logic synthesis system uses a series of steps to convert a logic design into a technology mapped circuit. Figure 1 shows our power-aware FPGA synthesis flow. The initial circuit input is an un-optimized Boolean network. BDDs are built for each Boolean node in the network. After pre-processing to remove logic redundancy (sweep), power-aware iterative eliminate and decomposition procedures are performed to generate a collection of 2-input one-output nodes. Power-aware shared subcircuit extraction is performed to minimize power and area in the final LUT network. As a final step, the 2-input nodes are mapped to LUTs using EMap.

Accurate switching activity estimation forms an important part of our logic optimization flow. To allow for rapid re-evaluation of switching activity during circuit processing, a probabilistic, rather than simulation based approach is used. An effective probabilistic approach for estimating switching activity in logic circuits is the transition density model [9]. The model has two parameters: transition density D – the average number of transitions per unit time, and static probability P – the probability of the signal being high for a certain time period. For a signal y , the transition density of the signal $D(y)$ is:

$$D(y) = \sum_{i=1}^n P\left(\frac{dy}{dx_i}\right) \cdot D(x_i) \quad (1)$$

where n is the number of signals input to y , $P(dy/dx_i)$ is the probability a change in x_i will cause a change in y , and $D(x_i)$ is the transition density of x_i . A modeling tool based on this calculation approach [10] provides switching activity estimation for our flow for each node from primary inputs to primary outputs.

The following sections detail our approaches for power-aware BDD-based eliminate and decomposition.

3.1 Power-Aware Eliminate

Global BDD eliminate procedures create a single large global BDD for each design primary output (design output/flip flip input) using design primary inputs (design inputs/flip flop outputs) as the BDD inputs. The resulting size and complexity of these BDDs has been shown to lead to poor subsequent decompositions [14]. As a result, we develop a collapsing approach which results in multiple smaller BDDs for each primary output. To reduce logic minimization complexity, the combinational nodes needed for each primary output are reduced to a collection of smaller subcircuits via partial collapsing of the circuit. Each collapsed logic cluster is represented as a *local* BDD that is later processed by BDD decomposition.

To achieve clusters that are suitable for LUTs and that are also power efficient, we use a two-phase eliminate procedure. Each eliminate phase requires one or more DAG traversals with a specific cost and fanout objective. In the first phase, MFFCs are constructed in an effort to identify cones which match the properties of LUTs. In the second phase, nodes remaining from the first phase are collapsed in an effort to minimize cluster size and fanout. For each phase, collapsing is performed iteratively until the process converges based on the specific cost objective.

The first phase of eliminate creates MFFCs by iteratively collapsing nodes with a single fanout into their children.

Collapsing takes place during a straightforward DAG traversal from primary inputs to primary outputs. This approach has been shown [15] to lead to effective decompositions which can be packed into K-input LUTs during subsequent technology mapping. As a result of collapsing, all adjacent single fanout nodes can be placed into the same LUT reducing the required circuit area. BDD node count is used as a collapsing constraint during this iterative phase.

The objective for the power-aware second phase of eliminate is to minimize node fanout during collapsing while allowing for sufficient cluster growth to enable subsequent decomposition. This phase considers both BDD node count and node fanout. During each iteration possible collapsing is determined with a power-aware cost function which takes the fanout of nodes into account. Nodes with large fanout are undesirable since they can create a large number of wires between LUTs after technology mapping, increasing interconnect capacitance.

Collapsing for the second phase is evaluated via two cost metrics for each potential eliminate operation:

1. A cost metric which determines a possible increase in average node fanout as a result of collapsing.
2. A cost metric which measures the possible increase in BDD nodes.

The following definitions are used to better clarify the cost metrics. A **candidate** node is a node which may be collapsed into its fanout (or **child** nodes). **Parent** nodes have outputs which fan into the candidate node.

The fanout cost metric for a collapse of a candidate node into its child nodes is the resulting average fanout of the parent nodes divided by a scaling constant. This value can be defined as follows:

$$Fanout\ Cost = \frac{\sum_{i \in parent_nodes} |fanouts(i)|}{\beta * |parent_nodes|} \quad (2)$$

where $fanouts$ is the number of fanouts of the parent nodes after the collapse, $|parent_nodes|$ is the number of parent nodes, and β is a scaling constant. A β value of 4 was found to give best results through experimentation. A potential node collapse is only accepted if the result of Eq. (2) is less than 1, indicating that fanouts greater than 4 are undesirable. A β value of greater than 4 often leads to at least one high capacitance inter-cluster connection. Values less than 4 penalize connections which are generally routed using low-capacitance intra-cluster wiring.

The second cost metric which evaluates BDD node increase as a result of collapsing is defined as follows:

$$Node\ Cost = \left(\sum_{i \in new_nodes} |nodes(i)| \right) - \left(|nodes(candidate)| + \sum_{i \in child_nodes} |nodes(i)| \right) \quad (3)$$

where $|nodes(candidate)|$ is the BDD node count of the candidate node before collapsing, $\sum_{i \in child_nodes} |nodes(i)|$ is the BDD node count of all the child nodes before collapsing, and $\sum_{i \in new_nodes} |nodes(i)|$ is

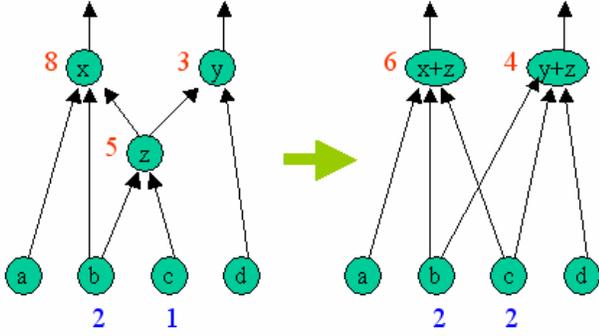


Figure 2: Example of power-aware partial collapsing

the BDD node count of combined nodes created by collapsing the candidate node into its children. The cost gives the difference between BDD node count before and after collapsing the candidate node. If the *Node Cost* is greater than the predefined threshold value, collapsing is not performed. A node count growth of greater than 10 generally requires an additional LUT, adversely affecting area.

Figure 2 illustrates a power-aware partial collapsing of candidate node *z* into child nodes *x* and *y*. First, *Fanout Cost* is calculated. There are two parent nodes of node *z* (*b* and *c*). After collapsing, nodes *b* and *c* have 2 fanouts each. The *Fanout Cost* is $(fanouts(b) + fanouts(c)) / (4 * \text{fanin_nodes}) = (2 + 2) / (4 * 2) = 0.5$. Since *Fanout Cost* is less than 1, *Node Cost* is calculated to determine if the collapse should be performed. As shown in Figure 2, the number of nodes before collapsing is $(8 + 3 + 5) = 16$ and the number of nodes after collapsing is $(6 + 4) = 10$. Therefore, the *Node Cost* is $10 - 16 = -6$, indicating collapsing can proceed.

Figure 3 lists the per-phase steps in the identification and subsequent collapse of nodes during the power aware second phase. The node-based design is topologically ordered and traversed from primary inputs (PIs) to primary outputs (POs). An ROBDD is then implemented for each node in the design. Clusters are determined and collapsible nodes are identified and collapsed. During each iteration these steps are performed until no additional collapsible nodes are found in the network. Node collapsing is performed via variable substitution.

3.2 Power-Aware BDD Decomposition

Unlike previous BDD decomposition approaches which have focused on area reduction [14][15], our approach simultaneously optimizes both area and signal switching characteristics of decomposed functions. As a result, switching activity, in the form of transition density, is directly used in evaluating the costs of decompositions. The result of BDD decomposition is an optimized 2-input node network that is suitable for subsequent shared subcircuit extraction and technology mapping. As mentioned earlier in Section 3, node transition densities can be quickly evaluated for each proposed decomposition based on probabilistic techniques.

Our decomposition engine, which is based on BDS [14], performs a heuristic search for efficient BDD decompositions, including both algebraic and Boolean decompositions. For each BDD node under decomposition, the engine first searches for algebraic

Begin

Topologically order network nodes from PIs to POs

Build BDD for each node

Identify collapsible nodes

While number of collapsible nodes $\neq 0$

Evaluate Eq (2) and (3) for node

If *Fanout Cost* < 1 and *Node Cost* < 10

Collapse node into immediate fanout

Update network and re-identify all collapsible nodes

End while

End

Figure 3: Iterative eliminate algorithm – second phase

decompositions, which are based on 0, 1, and *x*-dominators, to perform simple AND, OR, and XOR decompositions, respectively. Subsequently, a series of Boolean decompositions based on generalized AND, OR, and XOR decompositions are attempted. If none of these decompositions are successful, the BDD is co-factored with respect to its top variable. For each BDD, several different decompositions may exist depending on where the BDD is cut. Following decomposition, the source BDD is broken into two parts, the dominator *dm* and decomposed *dp* functions. These parts are then combined via a simplified function (e.g. AND, OR, XOR). For example, in Figure 4 both a disjunctive (top, right) and conjunctive (bottom, right) decomposition of the BDD on the left are possible.

Each potential BDD decomposition is evaluated with a cost function which considers both the resulting area and switching characteristics of the resulting circuit. Our cost function is defined as follows:

$$Cost = \alpha \cdot \frac{D(dm * dp)}{D(orig)} + (1 - \alpha) \cdot \left(\frac{nodes(dm) + nodes(dp)}{2 \cdot nodes(orig)} + \frac{variables(dm \cap dp)}{2 \cdot variables(orig)} \right) \quad (4)$$

where α is a scaling constant, $D(dm * dp)$ is the transition density of the BDD function after decomposition, $D(orig)$ is the transition density of the original BDD before decomposition, $nodes()$ indicates the number of BDD nodes in a BDD, $variables(dm \cap dp)$ is the number of shared variables in *dp* and *dm* and $variables(orig)$ is the number of variables in the original BDD.

The first part of the cost function optimizes the switching activity (power cost) and the second part optimizes area for BDD decompositions. Through experimentation, an α value of 0.25 was determined to generate best results since this value puts more of a bias on area than switching activity. Since minimizing area also minimizes dynamic power for most designs due to power reductions in logic and associated clocking and routing, this factor is given more weight than the switching activity metric. Values of α less than 0.25 tend to cause our algorithm to ignore decompositions that have slightly greater area but significantly reduced switching.

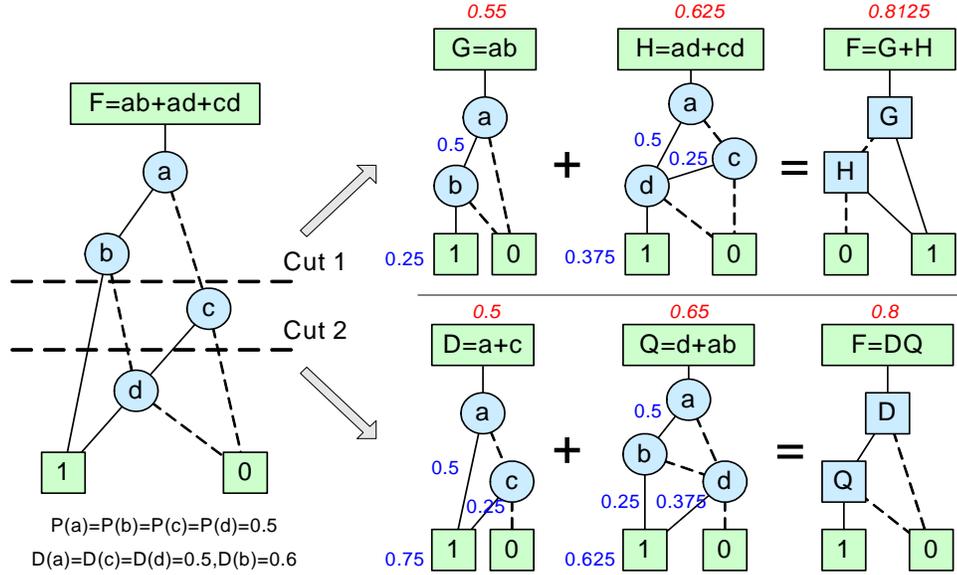


Figure 4: Different cuts for the BDD of $F = ab + ad + cd$

Figure 4 illustrates two possible cuts which generate two different decompositions of the BDD for the function F . Both cuts have the same area costs based on the number of nodes after decomposition. But using the transition density based switching activity estimations (shown in italics at the top of the BDDs), the signal switching for the two cuts differ. From the figure it can be seen that bottom cut is more desirable for F since it has a lowest transition density (0.8). Static probabilities are labeled on each BDD edge and transition density and static probability values for input variables are shown at the bottom left. The decomposed nodes are stored in factoring trees after BDD decomposition. To reduce the area in the final 2-input logic gate network, extraction of shared subtrees is performed to find logic sharing among different parts of the network. This technique reduces area (number of LUTs) since node redundancies are eliminated. Sharing extraction also reduces the number of duplicated fanins of a node in the subtree, thus reducing the number of wires at the input and within the extracted subtree.

4. EXPERIMENTAL APPROACH

Our new logic optimization system has been integrated into two FPGA mapping flows to evaluate its power and energy reduction potential. Following logic optimization, we use EMap, a power-aware technology mapper, for LUT packing. Following mapping, two place and route options are possible. In the first flow, we use VPR, an academic place and route tool for physical synthesis. Post-route core dynamic power for this flow is determined with the transition density estimator described in [10]. The target

architecture for this flow is an FPGA model with four LUTs per logic block and wire segments of length 4. A 0.18 μm technology is assumed for the FPGA model. A second flow uses Altera Quartus II 5.0 to perform placement and routing to Cyclone II devices. Altera's PowerPlay power analyzer was used with random input waveforms to determine core dynamic power results. The timing requirements are set to achieve maximum performance for all designs during place and route by setting the clock frequency constraint to an unattainable 1 GHz.

To evaluate the synthesis benefits of our new approach, we compare the power and area results of our power-aware BDD-based FPGA synthesis tool with two other logic synthesis tools using the two flows mentioned above. A synthesis flow which uses SIS and DMIG [12] is used to allow for comparison to previous work with EMap [5]. In the SIS flow, technology independent logic optimization is performed on the initial circuit designs using SIS (script.rugged). The optimized circuits are then transformed into networks of 2-input simple gates using SIS's *tech_decomp* and DMIG. In addition to SIS comparisons, we also compare the synthesis results of our power-aware BDD tool to our original BDD decomposition tool (BDS-pga [15]) that optimizes FPGA area. All other steps in the evaluation flows remain the same for these two alternate synthesis approaches. For each experiment, 16 large MCNC benchmarks were used, as listed in Table 1. The experiments were conducted on a Pentium-4/1.8GHz machine with 512MB of RAM. All designs were targeted to the smallest FPGA which would hold them.

Table 1: Detailed results of SIS flow, BDS-pga flow, and power-aware BDD flow for Altera 90 nm Cyclone II FPGAs

	SIS Flow (A)			Original BDS-pga Flow (B)			Power-Aware BDD Flow (C)				
	LUTs	Max Delay (ns)	Energy @ 50 MHz (nJ)	LUTs	Max Delay (ns)	Energy @ 50 MHz (nJ)	LUTs	Max delay (ns)	Energy @ 50 MHz (nJ)	Energy ratio (C)/(A)	Energy ratio (C)/(B)
alu4	1400	7.65	1.60	552	6.94	0.79	528	7.11	0.81	0.51	1.02
apex2	1779	9.00	1.61	1846	10.84	1.69	1740	9.77	1.61	0.99	0.95
apex4	1294	7.06	0.85	1302	9.39	0.90	1203	9.09	0.70	0.83	0.78
bigkey	1810	3.92	2.59	1365	4.62	2.46	1478	4.49	2.42	0.93	0.98
des	1391	6.18	3.59	1069	4.89	2.71	1077	5.30	2.82	0.78	1.04
diffreq	1069	10.01	0.53	1090	8.54	0.55	1086	10.30	0.53	1.00	0.96
dsip	1366	3.76	2.43	1433	4.36	2.39	1460	4.18	2.33	0.96	0.97
elliptic	2319	11.72	1.87	2494	12.10	1.91	2444	12.56	1.77	0.95	0.93
ex1010	4405	9.51	1.26	4655	13.55	1.54	4610	11.98	0.81	0.64	0.52
ex5p	1057	8.37	0.86	1112	9.25	0.90	1095	9.78	0.75	0.88	0.84
frisc	2563	15.67	1.07	2825	16.76	1.12	2819	17.70	1.00	0.93	0.89
misex3	1324	7.83	1.34	2032	10.25	2.33	1443	9.00	1.68	1.25	0.72
pdc	4498	10.71	1.33	3600	11.90	1.45	3472	13.10	1.24	0.93	0.86
s298	1738	15.54	0.83	1880	19.68	1.16	1113	17.07	0.72	0.86	0.61
seq	1605	7.30	1.61	1631	8.50	1.66	1619	8.51	1.60	1.00	0.96
spla	3725	9.64	1.44	2842	11.67	1.20	2818	11.49	1.01	0.70	0.84
Geometric Ave.	1858	8.38	1.39	1739	9.38	1.41	1629	9.33	1.20	0.87	0.85

Table 2: Detailed results of SIS flow, BDS-pga flow, and power-aware BDD flow for VPR 0.18 μm model

	SIS Flow (A)			Original BDS-pga Flow (B)			Power-Aware BDD Flow (C)				
	LUTs	Max Delay (ns)	Energy @ 5 MHz (nJ)	LUTs	Max Delay (ns)	Energy @ 5 MHz (nJ)	LUTs	Max delay (ns)	Energy @ 5 MHz (nJ)	Energy ratio (C)/(A)	Energy ratio (C)/(B)
alu4	1400	42.23	27.61	552	33.84	16.88	528	33.61	16.79	0.61	0.99
apex2	1779	47.54	26.53	1846	59.64	25.29	1740	53.00	25.22	0.95	1.00
apex4	1294	36.82	16.78	1302	52.96	14.18	1203	50.23	14.16	0.84	1.00
bigkey	1810	45.86	71.23	1365	22.76	73.03	1478	38.69	57.25	0.80	0.78
des	1391	31.31	75.03	1069	29.36	65.73	1077	29.98	73.46	0.98	1.12
diffreq	1069	47.31	12.56	1090	50.24	12.88	1086	52.62	11.76	0.94	0.91
dsip	1366	24.58	55.23	1433	37.00	50.28	1460	36.42	50.45	0.91	1.00
elliptic	2319	69.19	28.05	2494	66.91	30.79	2444	75.38	27.87	0.99	0.91
ex1010	4405	49.76	37.13	4655	83.14	55.61	4610	62.03	28.95	0.78	0.53
ex5p	1057	43.52	15.77	1112	48.08	14.62	1095	53.80	14.63	0.93	1.00
frisc	2563	87.96	17.08	2825	97.95	17.91	2819	103.92	16.88	0.99	0.94
misex3	1324	42.40	22.55	2032	61.56	32.19	1443	55.30	20.41	0.91	0.63
pdc	4498	55.13	44.21	3600	66.95	41.36	3472	76.41	40.21	0.91	0.97
s298	1738	82.88	16.32	1880	139.54	15.91	1113	110.66	14.46	0.89	0.91
seq	1605	43.39	27.30	1631	47.64	26.77	1619	46.18	26.72	0.98	1.00
spla	3725	57.86	39.85	2842	61.79	38.54	2818	61.75	36.58	0.92	0.95
Geometric Ave.	1858	47.99	28.82	1739	54.40	28.45	1629	54.96	25.63	0.89	0.91

5. EXPERIMENTAL RESULTS

In an initial experiment, all 16 designs were synthesized using SIS/DMIG, our original BDD tool (BDS-pga), and our new power-aware BDD tool. Following technology mapping with EMap using default settings, the designs were packed, placed, and routed by Quartus II with clock constraints. The results from this experiment are summarized in Table 1. Two main results are apparent from the table. For a fixed design clock frequency, our power-BDD synthesis tool achieves about a 13% overall energy

reduction versus a flow with SIS synthesis and about a 15% reduction versus a flow which includes our previous BDD synthesis approach. In the table, energy values are based on dynamic power values of the design (except I/Os) determined via simulation. Note that although we list energy results at 50 MHz (the clock speed used for simulation), the relative results for dynamic power and energy will not change if the clock frequency for all three mappings for each design is increased to the maximum value allowed among the three. For example, design **alu4** is constrained by a 7.65 ns period for SIS optimization so all

Table 3: Pre-map fanout and transition density values

	Average Fanout	% Diff	Ave. Trans. Density	% Diff
BDS-pga	2.05	0	0.30	0
BDS-power minus elim	2.07	1.0	0.27	-9.0
BDS-power minus decomp	1.91	-7.3	0.28	-6.2
BDS-power	1.93	-5.4	0.24	-18.0

designs could have been evaluated at this frequency with the same relative result. This constant energy ratio occurs since both dynamic power and energy due to dynamic power are linearly proportional to clock frequency.

This significant energy reduction comes at the cost of maximum achievable clock frequency. The peak clock frequency for our new BDD approach is about 11% worse than the peak frequency achievable with SIS.

This increase is not surprising since our eliminate and decomposition steps focus on reduced area and signal switching rather than depth reduction. We believe that for many applications, energy savings will be of utmost importance during design especially for low to moderate performance embedded applications.

It should be noted that although most designs for the Cyclone II experiments have a simulation coverage of close to 100% of design nodes, designs **ex1010** and **pd** only have coverage of 82% and 89% respectively, which may skew their results.

To verify our results for Cyclone II, we performed the same set of experiments using the VPR flow described in Section 4. As seen in Table 2, an 11% energy savings due to dynamic power reduction was also achieved with this flow versus the SIS flow. The delay average increase of 15% is also similar. The overall energy savings are likely lower for the 180 nm model used by VPR versus Cyclone II due to VLSI technology difference and since VPR does not model the same interconnect richness found in the commercial 90 nm Cyclone II device.

As mentioned earlier in this section, the SIS flow results (A) in Tables 1 and 2 were generated with default EMap technology mapping settings which attempt to simultaneously optimize for both power and performance. In an additional set of experiments, the netlists created by SIS synthesis were remapped by EMap under relaxed timing constraints so that average delay increased by 11%, the same amount achieved by power-aware BDD synthesis followed by default EMap technology mapping (Flow C). This remapping achieved an 8% area and 10% energy reduction versus the initial SIS flow (A). The energy savings are consistent with previous results [1] which examine power savings versus delay tradeoffs in technology mapping. In comparison to the delay-relaxed SIS/EMap flow, the power-aware BDD flow reduced area by 4% and energy by 3%.

In our experimentation, we only examine dynamic power since the static power of each design varies only a small amount per implementation for a fixed FPGA device. Although the logic area required by the designs varied based on the synthesis tools used,

the smallest package required to fit each design did not change. The area reduction achieved by the power-aware BDD synthesis tool may have more of an impact on static power if future FPGAs provide the capability to power down device regions. Although the percentage reduction in energy for Cyclone II (13%) is roughly the same as the amount of logic area reduction (12%) for the power-aware BDD synthesis flow, the use of power-aware eliminate and decomposition plays a role in energy reduction. From Table 1 it can be seen that if the original BDS-pga synthesis flow (B) is used, logic area is reduced by about 6% versus the SIS flow (A) but energy remains roughly constant.

The benefit of each step of the power-aware BDD synthesis approach can be seen in Table 3. This table lists average node fanout and transition density values for designs that have been synthesized but not technology mapped. In addition to synthesis with our original and new BDD tools, we also performed synthesis with the power-aware eliminate shut off (minus elim) and power-aware decomposition shut off (minus decomp). For those cases, their non-power-aware counterparts were used instead. As seen from the table, the power-aware eliminate was particularly important in reducing average fanout and power-aware decomposition reduces transition density. This effect was also seen in the *post-map* LUT designs created by EMap. The average post-map fanout for BDD-pga versus BDD-power is 2.91 vs. 2.73 and the average transition density is 0.23 vs. 0.21. The reduction of both of these values plays a role in reducing dynamic power along with design area reduction.

To address the issue of increased circuit delay for designs mapped with our new power aware BDD synthesis tool, an additional delay resynthesis approach was attempted on designs following the BDD synthesis shown in Figure 1 and before EMap. This approach uses tree height reduction [12] on a design after power-aware BDD decomposition. Functionally equivalent gates in the circuit paths are collapsed together and re-decomposed into 2-input gates using DMIG, a tree-height reduction decomposition algorithm. A Huffman encoding procedure [12] is used in DMIG to reduce the depth of the logic gate network.

With this approach, an average 2% improvement in maximum delay versus results in Table 1 is achieved for BDS-power, but the maximum delay results are still 9% larger on average than the SIS flow. The energy saving over SIS is reduced from 13% to 10% on average while the area result remains the same.

6. CONCLUSION

This work presents a power-aware BDD-based synthesis system for FPGAs. Our BDD synthesis system performs partial collapsing during logic optimization to reduce node fanout. Additionally, signal switching information is used during logic decomposition to achieve decompositions that are both area and power efficient. Both of these goals have been shown to favorably assist subsequent power-aware technology mapping.

Several advancements warrant evaluation as next steps. It may be possible to limit node depth during decomposition by using a more global view of BDDs. Alternatively, it may be possible to initially explore algebraic decompositions along critical paths followed by BDD-based decomposition along non-critical paths.

7. ACKNOWLEDGMENTS

This work was funded by a grant from Altera Corporation. We thank Julien Lamoureux from the University of British Columbia for providing the EMap software. We acknowledge the efforts of Mohammed Alhussein in preparing the final version of the paper.

8. REFERENCES

- [1] J. Anderson and F.N. Najm, "Power-Aware Technology Mapping for LUT-Based FPGAs," *IEEE Int. Conf. on Field Programmable Technology*, Dec. 2002, pp. 211-218.
- [2] A. H. Farrahi and J. Sarrafzadeh, "FPGA Technology Mapping for Power Minimization," *Int. Workshop on Field-Programmable Logic and Applications*, 1994, pp. 66-77.
- [3] G. Holt and A. Tyagi, "Minimizing Interconnect Energy Through Integrated Low-Power Placement and Combinational Logic Synthesis," *Proc. ISPD*, Apr. 1997, pp. 48-53.
- [4] S. Iman and M. Pedram, "POSE: Power Optimization and Synthesis Environment," *Proceedings of the 33rd Design Automation Conference*, June 1996, pp. 21-26.
- [5] J. Lamoureux and S.J.E. Wilton, "On the Interaction between Power-Aware FPGA CAD Algorithms," *Proc. ICCAD*, Nov. 2003.
- [6] C. Lennard, P. Buch, and A. Newton, "Logic Synthesis using Power-Sensitive Don't Care Sets," *Proc. of Int. Symposium on Low Power Electronics and Design*, 1990, pp. 293-296.
- [7] H. Li, W.-K. Mak and S. Katkooori, "LUT-Based FPGA Technology Mapping for Power Minimization with Optimal Depth," *IEEE Computer Society Workshop on VLSI*, 2001.
- [8] P. Lindgren, M. Kerttu, M. Thornton, and R. Drechsler, "Low Power Optimization Technique for BDD Mapped Circuits," *Proc. ASP-DAC*, Jan. 2001, pp. 615-621.
- [9] F. N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits," *IEEE Transactions on Computer-Aided Designs of Integrated Circuits and Systems*, vol. 12, no. 2, Feb. 1993, pp. 310-323.
- [10] K. Poon, A. Yan, and S. Wilton, "A Flexible Power Model for FPGAs," *International Conference on Field-Programmable Logic and Applications*, Sep. 2002.
- [11] S. Roy, A. Harm, and P. Banerjee, "PowerShake: A Low Power Driven Clustering and Factoring Methodology for Boolean Expressions," *Proceedings of Design, Automation and Test in Europe Conference*, Feb. 1998.
- [12] E. M. Sentovich et al., "SIS: A System for Sequential Circuit Synthesis," *UC Berkeley, Memorandum No. UCB/ERL M92/41, Electronics Research Laboratory*, May 1992.
- [13] Z.-H. Wang, E.-C. Liu, J. Lai and T.-C. Wang, "Power Minimization in LUT-Based FPGA Technology Mapping," *Proc. ASP-DAC*, Jan. 2001, pp. 635-640.
- [14] C. Yang and M. Ciesielski, "BDS: A BDD-Based Logic Optimization System," *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Sys.*, vol. 21, no. 7, July 2002.
- [15] N. Vemuri, P. Kalla, and R. Tessier, "BDD-based Logic Synthesis for LUT-Based FPGAs," *ACM Transactions of Design Automation of Electronic Systems*, vol. 7, no. 4, Oct. 2002, pp. 501-52.