Multiplexer Based Circuit Synthesis with Area-Power Trade-Off

Sambhu Nath Pradhan[1] and Santanu Chattopadhyay[2]

[1] Dept. of ECE NIT-Agartala, and [2] Dept. of Electronics & ECE, IIT-Kharagpur, India
sambhu.pradhan@gmail.com, santanu@ece.iitkgp.ernet.in

Abstract. Due to the regularity of implementation, multiplexers are widely used in VLSI circuit synthesis. This paper proposes a technique for decomposing a function into 2-to-1 multiplexers performing area-power tradeoff. To the best of our knowledge this is the first ever effort to incorporate leakage into power calculation for multiplexer based decomposition. With respect to an initial ROBDD (Reduced Ordered Binary Decision Diagram) based representation of the function, the scheme shows more than 30% reduction in area, leakage and switching for the LGSynth91 benchmarks without performance degradation. It also enumerates the trade-offs present in the solution space for different weights associated with these three quantities.

Keywords: Power minimization, area-power trade off, multiplexer synthesis, ROBDD.

1 Introduction

Multiplexers have long drawn the attention of VLSI design and research community for circuit realization. This is mainly due to the high regularity of multiplexers that make the structure particularly suitable for VLSI implementation. Introduction of low-power regime has brought with it further challenges due to the increased usage of portable, handheld, battery-operated, plastic-packaged devices. This type of devices necessitate low-power consumption for increased battery life, low-cost etc. Due to this, researchers have been motivated to look into the multiplexer decomposition problems [3 - 7]. In these works, a large multiplexer is decomposed into a tree of 2-to-1 multiplexers. The problem of power-efficient decomposition has been handled in [3]. Here, given an arbitrary *n*-to-1 multiplexer with a fixed encoding for the data signal, minimum power balanced decomposition into a multiplexer tree consisting of 2-to-1 multiplexers has been done. Both uniform and nonuniform low power decompositions have been studied. Power dissipation of all the multiplexers in the tree is calculated from the ON probabilities of their outputs. In this work only single output functions have been considered. References [4] and [6] dealt with delay minimal decomposition of multiplexers. The main objective of [4] and [6] is to minimize the depth of the 2-to-1 multiplexer decomposed tree. Here, multi-output functions and power reduction issues are not considered. In [7] a method of estimating the signal

N. Meghanathan et al. (Eds.): CCSIT 2011, Part III, CCIS 133, pp. 410–420, 2011.

transition number in multiplexer tree has been proposed. It also describes a low power decomposition method based on signal transitions. In this work, the transition number or transition probability of input data signal is measured in advance. The drawback in this approach is that when the circuit is in working condition, we cannot say the exact transition probability or transition number of the data signal. It may be noted further that in [7] the authors have used a test vector generator to get an estimation of signal transitions. This is not very suitable since the input patterns fed to a circuit during its normal operation are generally substantially different from the test patterns. In normal operation, inputs are highly correlated, whereas, to maximize fault coverage, successive test patterns are made uncorrelated. Thus, until and unless the exact environment in which the circuit is going to be placed is known, it is not possible to predict the signal transitions.

Moreover, none of these techniques handled multi-output functions which are the most common structures in any digital system. The decomposition obtained in all the above works is in the form of tree only. Sharing of subfunctions at various levels of decomposition has not been considered. For multi-output functions, there exists a high chance of common subfunctions being shared by number of outputs. Authors in [12] have proposed a methodology to synthesize multi-output function using 2-1 multiplexers. The work also performed a trade-off between the area of the resulting circuit (in terms of number of 2-1 multiplexers) and the dynamic power consumed (measured as estimated switching activity over all multiplexers). However, the work ignored the leakage power consumed by the circuit under the premises that leakage power is quite insignificant as compared to the dynamic power. The International Technology Roadmap for Semiconductors (ITRS) projects an exponential increase in the leakage power with minimization of devices [2]. As the technology drops below 65 nm feature sizes, subthreshold leakage is expected to exceed the total dynamic power. As leakage current becomes the major contributor to power consumption, the industry must reconsider the power equation that limits system performances, chip sizes and cost.

In this paper we have presented a multiplexer targeted circuit synthesis scheme that thrives to attain the minimization of both dynamic and leakage power. To the best of our knowledge this is the first ever work that handles leakage power in multiplexer synthesis process. A trade-off has also been performed between the area of the resulting circuit and its estimated dynamic and leakage power consumptions.

Binary Decision Diagrams (BDDs) [1] are the natural representation of functions implemented using multiplexers. Each BDD node corresponds to a multiplexer. Figure 1(a) shows the BDD representation of a 2-output Boolean function, whereas, Figure 1(b) shows the corresponding multiplexer realization. Due to this natural correspondence, we have used Reduced Ordered BDD (ROBDD) representation of function to arrive at the area and power optimized multiplexer-based realization of the function.

The rest of the paper is organized as follows. In Section 2, different sources of power dissipation of CMOS circuits, estimation of switching activity and leakage power are given. Section 3 of this paper illustrates the multiplexer based circuit synthesis strategy. Section 4 shows the experimental result of our decomposition method.

2 Sources of Power Dissipation

In order to develop techniques for minimization of power dissipation, it is essential to identify various sources of power dissipation and different parameters involved in each of them. CMOS has emerged as the technology of choice for low power applications and is likely to remain so in the near future. The power dissipated in CMOS circuits is classified into dynamic power and leakage power.

2.1 Dynamic Power Dissipation

Dynamic power dissipation in CMOS circuit occurs when the circuit is in working condition or active mode. Following are the two major constituents of dynamic power dissipation [10]: Switching power, Short-circuit power.

Amongst these, the switching power is the dominating contributor. This occurs due to the charging and discharging of load and parasitic capacitors.

$P_{dynamic} = \alpha_L . C_L . V_{DD}^2 . f + \sum_i \alpha_i . C_i . V_{DD} . (V_{DD} - V_T)$ here,α_L and α_i are the switching activity at the load and at the internal node of the circuit respectively. V_{DD} is the supply voltage, f is the frequency of operation, C_L and C_i are the load and internal gate capacitances respectively, V_T is the threshold voltage.

Switching activity estimation strategy

To compute the total switching activity in a BDD, we need to first find the switching activities at individual nodes and then sum them up to get the overall quantity. In the following, the process has been enumerated.

A Boolean function f can be expanded around one of its variable, say x, using Shannon Expansion [4] as follows.

$$f = x \cdot f_1 + x' \cdot f_0$$

where, f_1 and f_0 are the functions resulting from f, setting $x = 1$ and $x = 0$ respectively.

If $P_{on}[f]$ be the probability of the function f being ON, then,

$$P_{ON}[f] = P_{ON}[x].P_{ON}[f_1] + P_{ON}[x']. P_{ON}[f_0] = P_{ON}[x].P_{ON}[f_1] + (1-P_{ON}[x]). P_{ON}[f_0].$$

Similarly, if $P_{OFF}[f]$ be the probability of f being OFF, then,

$$P_{OFF}[f] = P_{OFF}[x].P_{OFF}[f_0]+ P_{ON}[x] .P_{OFF}[f_1]= (1-P_{ON}[x]).(1-P_{ON}[f_0]) + P_{ON}[x]. (1-P_{ON}[f_1]).$$

Now, a switching of f implies either of the following:

Case I : f was ON at time t and is OFF at its successive time instant.
Case II: f was OFF at time t and is ON at its successive time instant.

Thus, if $P_{switching}[f]$ denotes the transition probability of f, then,

$$P_{switching}[f] = P_{ON}[f].P_{OFF}[f] + P_{OFF}[f].P_{ON}[f] = 2.\ P_{ON}[f].P_{OFF}[f] = 2.\ P_{ON}[f].\ (1-P_{ON}[f]) \quad (1)$$

Assuming uniform input probabilities, that is ON and OFF probabilities of inputs like x to be 0.5, $P_{ON}[f]$ can further be simplified to

$$P_{ON}[f] = 0.5\ P_{ON}[f_1] + 0.5\ P_{ON}[f_0] \quad (2)$$

ON probability of any ROBDD node with variable x can be computed by recursively descending down the tree and using Equation 2. The recursion is terminated when a trivial ROBDD node, i.e., a constant node is reached. At a trivial ROBDD node $P_{ON}[0] = 0$ and $P_{ON}[1] = 1$. After calculating ON probability of a ROBDD node we can calculate the switching probability using Equation 1. Switching probabilities of all nodes are then summed up to get the overall switching activity.

2.2 Leakage Power Dissipation

Static power dissipation occurs due to various leakage mechanisms. A good review of leakage current is given in [9]. Among the various leakage components, two main leakage current components are sub-threshold leakage current and gate oxide tunneling current.

Leakage power estimation

In this section, leakage power calculation at each of the ROBDD nodes has been illustrated. Each ROBDD node can be implemented using two transmission gates connected to form a multiplexer. To calculate the leakage of such a structure, we have used the runtime mode leakage [15] considering all input probabilities. It may be noted that input probabilities are already computed in the switching activity estimation process in Section 2.1. Figure 2 shows the realization of a multiplexer using transmission gates. Table 1 gives input dependent leakage current values. It may be noted that due to the symmetrical structure of the transmission gate multiplexer, leakage current values for the patterns '000' and '100' are same. Similarly, for the patterns '011' and '111', leakages are same. Leakage current dissipation for each of the other patterns is also same for this symmetrical structure. Simulation for leakage power has been carried out using Cadence Spectre at 90 nm UMC technology. Length and width of PMOS and NMOS transistors are 90 nm and 180 nm respectively. The supply is 1 voltage. We have calculated the leakage current and hence leakage power of multiplexer using the state probabilities of the multiplexer inputs. As we know the state probabilities of the individual input nodes of the multiplexer, we can compute the probability of each input state of the multiplexer. The estimated leakage power after calculating the probability of each state would thus be given by,

$$P_{leakage} = V_{dd} \sum_k S_k * I_k \quad (3)$$

k is over all the possible eight input states of the multiplexer. S_k is the probability of state k and I_k is the leakage current in state k. V_{dd} is the supply voltage.

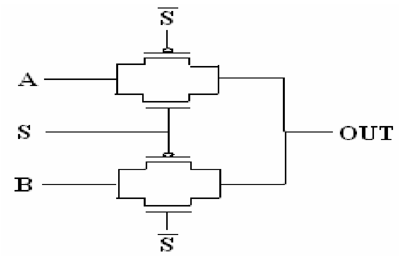

Fig. 2. Schematic of multiplexer

Table 1. Leakage current for different input values

S	A	B	Leakage current
0	0	0	0.01fA
0	0	1	2.44nA
0	1	0	2.44nA
0	1	1	0.02fA
1	0	0	0.01fA
1	0	1	2.44nA
1	1	0	2.44nA
1	1	1	0.02fA

3 Multiplexer Circuit Synthesis

In our approach, we have used a top-down heuristic [3] to decompose a given multi-output combinational function into 2-to-1 multiplexers. The heuristic has three objectives — minimize the area of the resulting circuit (in terms of number of 2-to-1 multiplexers needed), minimize the estimated switching activity at all multiplexers, and minimize leakage. The algorithm starts with a Reduced Ordered Binary Decision Diagram (ROBDD) based representation of the function using the BDD package *BuDDy* [13]. It then selects one of the variables around which the individual component functions of the multi-output function are decomposed using Shanon expansion. This generates a set of next-level functions. These next-level functions are again decomposed using one of the remaining primary input variables. The process continues till we have trivial subfunctions (constant 0, 1, or a single variable).

The major decision in the process outlined above is to select a variable around which decomposition is to be carried out at a level. Ideally, for each variable choice, we should decompose the resulting subfunctions using all possible sequences of remaining variables to get the minimum. However, the process is of exponential complexity. To simplify the process, to compare between two contending variables at any level of decomposition we do the following.

(i) Area complexities are compared by looking into the number of unique subfunctions generated at next level due to decomposition through each of the variables.

(ii) Dynamic and leakage powers are estimated as shown in Section 2.1 and 2.2 respectively.

Since we are targeting a power-area trade-off, we associate a cost function with each of the variables to evaluate their fitness to be used for decomposition. For variable *x*, its cost is given by,

$$cost(x)= w_1 \frac{area}{max\_area} + w_2 \frac{Switching}{max\_switching} + w_3 \frac{leakage}{max\_leakage}$$

Here, *"area"* is the number of next level unique subfunctions and *"max-area"* is the maximum number of next level unique functions amongst all the variables. Similarly, *"max-switching"* is the maximum of total switching activities in the set of next level subfunctions generated through decomposition around each of the contending variables, and "max-leakage" is the maximum leakage value. The weights w_1 , w_2 , w_3 can be set by the designer with $w_1 + w_2 + w_3$ being equal to 1. Next the algorithm for decomposition is given formally.

Algorithm Mux-Decomposition

```
Input : A multi output Boolean function
Output : A multiplexer based graph representation of the input function
1. Obtain the ROBDD representation of the given circuit
2. While(no_variable>1 and no_function>0)
   i) For( k=1 to no_variable ) do
        restricted_function[k]=restrict_function_by_variable(k)/*Shannon expansion around k */
        check for possible sharing.
        mux_count_function[k] = find_no_of_nextlevel_function(restricted_function[k])
        switching_activity[k] = total_switching_of_nextlevel_function(restricted_function[k])
        leakage_power[k]=tatal_leakage_power_of_nextlevel_function(restricted_function[k])
        cost_function[k]=weighted_average(mux_count_function[k],switching_activity[k],leakage_power[k])
     end for
     ii) variable_selected = find_variable_resulting_minimum_cost()
     iii) no_variable =no_variable -1
     iv) no_function = update_function(restricted_function(variable_selected) )
 end while
 3. Find multiplexer based graph representation
```

4 Experimental Results

In this section we present the results of our multiplexer based circuit synthesis heuristic. The decomposition procedure explained in Section 3 has been implemented using ROBDDs. The construction and manipulation of the ROBDDs for implementation are performed using BDD package *BuDDy* [13]. For the experiment, a set of LGSynth91 benchmark circuits [14] has been used. The experimentation of the algorithm has been done on these circuits on a Linux based platform with Pentium-IV processor having 3GHz clock frequency and 1 GB main memory. Simulation of multiplexer circuit has been carried out at 90 nm technology to get leakage power values.

First, we present a comparison of our multiplexer synthesis approach with the initial ROBDD results given by *BuDDy*. Here, BDD package*, BuDDy* tries to represent a Boolean function using the specified variable ordering, which, in the default case, is the order in which the inputs are specified. The package performs some reordering in the case of increase in the number of nodes beyond certain limit. It is never the area-optimized realization. Henceforth, initial area, switching and leakage obtained from this initial ROBDD will be termed as *initial_area*, *initial_switching* and *initial_leakage*. We also have compared our synthesis results with the inorder results obtained using *BuDDy*. To obtain *inorder* results we have followed the same algorithm described in Section 3 but in this case variable selection is not based on the minimum cost. Here, variable ordering is the same as the sequence in which inputs are specified in the circuit. Table 2 shows the area (in terms of no. of multiplexers), switching (switching activity) and leakage (nw) comparisons. As it can be seen from the table, our approach requires much lesser number of multiplexers compared to the initial and inorder approaches. Column 2 of Table 2 gives the number of multiplexers of the initial ROBDD. Column 3 shows the number of multiplexers needed for inorder decomposition. Saving of area in approach *(w_1=1 w_2=0 w_3 =0)* compared to

Table 2. Area, switching and leakage comparison

Circuit	area			Switching			Leakage (nw)		
	Initial ROBDD	Inorder ROBDD	Our approach *(w_1=1 w_2=0 w_3=0)*	Initial ROBDD	Inorder ROBDD	Our approach *(w_1=0w_2=1 w_3=0)*	Initial ROBDD	Inorder ROBDD	Our approach *(w_1=0 w_2=0 w_3=1)*
alu2	257	252	200	108.82	106.32	90.39	289.57	277.52	288.47
alu4	1352	1342	718	543.31	538.31	281.41	1369.8	1345.67	1231.58
apex4	1021	1015	964	418.70	415.70	398.00	1104.8	1090.38	1143.83
apex5	2705	2611	1854	1242.74	1195.74	451.55	3263.5	3036.96	1851.82
aralis	229	222	166	45.22	41.72	29.79	132.47	115.59	115.59
bw	114	109	108	43.69	41.19	40.12	126.12	114.07	108.27
b12	91	76	67	30.55	23.05	25.03	103.09	66.92	70.27
cc	105	93	48	37.81	31.81	12.59	2270.0	91.57	42.40
cht	149	112	83	60.78	42.28	37.37	208.50	119.31	108.53
clip	254	247	94	109.25	105.75	38.46	8968.7	281.55	283.63
cm138a	17	16	16	1.82	1.32	1.32	6.03	3.62	3.62
count	219	184	168	91.00	73.50	88.81	282.78	198.39	241.17
cps	2329	2316	1923	421.65	415.15	280.60	1212.3	1180.96	1438.22
c8	145	125	100	64.92	54.92	43.43	4864.0	147.06	132.60
decode	118	94	66	42.30	30.30	25.55	147.07	89.21	77.30
duke2	976	969	398	143.51	140.01	88.91	372.34	355.47	247.39
ex1010	1079	1074	1072	306.55	304.05	297.85	789.39	777.34	767.60
ex4	1301	1235	563	484.67	451.67	214.10	1548.4	1389.26	1192.47
f51m	70	66	61	30.90	28.90	27.13	97.05	87.40	87.40
frg2	6520	6462	1238	2795.41	2766.42	229.24	7129.6	6990.10	1215.99
k2	2985	2951	1995	902.95	885.95	630.09	2450.2	2368.26	3461.43
lal	182	157	91	69.39	56.89	24.71	7390.5	162.89	137.11
maskmx	85	82	57	35.50	34.00	12.64	3565.3	94.74	54.74
misex1	47	37	36	19.37	14.37	13.87	65.34	41.23	46.55
misex2	140	127	111	24.74	18.24	13.19	1311.9	52.48	45.23
misex3c	847	838	484	326.07	321.67	159.05	852.8	831.07	1042.06
pbo2	226	215	187	76.11	70.61	61.48	6902.6	202.24	191.89
pbonew	326	315	284	111.09	105.60	88.07	329.8	303.28	258.76
pcle	93	78	62	35.39	27.89	8.35	2295	80.61	83.78
pcler8	145	129	85	60.02	52.02	23.56	186.78	148.20	164.69
pdc	705	683	667	139.89	128.89	67.15	424.07	371.04	307.09
pm1	50	43	43	14.01	10.57	8.34	1781.1	29.27	28.25
sao2	154	153	87	38.76	38.26	23.26	2032.9	97.30	111.36
sct	169	146	54	63.76	52.26	18.61	193.95	138.49	115.13
seq	142321	*	2000	27967.20	*	649.92	71608.13	*	5829.20
shiftc	102	90	79	39.16	33.16	32.16	3601.6	96.25	79.22
sp	625	605	585	136.94	126.94	126.82	406.60	358.35	678.06
spla	681	661	659	135.74	125.74	78.33	402.86	354.64	642.80
table3	941	934	1349	109.3	105.80	115.33	287.85	270.98	548.28
table5	873	866	742	147.59	143.62	118.21	398.77	379.56	600.51
term1	586	572	162	154.15	147.15	76.51	11059.7	359.44	648.43
ttt2	248	223	139	108.26	95.76	52.92	309.66	249.38	223.52
vda	4421	4417	752	1854.20	1852.02	282.48	4619.1	4604.70	4606.70
vg2	1059	1049	96	362.39	357.39	25.36	959.08	934.97	1211.21
x1	1583	1549	578	341.03	324.03	178.71	955.20	873.23	765.42
x2	73	70	30	28.57	27.07	10.09	82.21	74.98	59.62
x4	916	846	388	369.24	334.24	139.81	1063.7	894.9	754.71
Average % saving of our approach	**34.17**	**29.22**		**41.35**	**34.07**		**31.18**	**-03.05**	

initial and inorder ROBDD are *34%* and *29%* respectively. As inorder approach considers possible sharing at each level of decomposition, saving is lower than the initial approach. Switchings of initial, inorder and our approach *(w_1=0 w_2=1 w_3=0)* are noted in Coulumn 5, 6 and 7 respectively. From Table 2, it is seen that average savings in switching of our approach compared to initial and inorder are *41%* and *34%.* Average leakage saving compared to initial ROBDD is *31%* but on the average we cannot get leakage saving compared to inorder approach. In some circuits like, *apex5(39%),cc(54%),decode(13%), duke2(30%), ex4(14%), frg2(83%), maskmx(42%), misex2 (14%), pbonew(15%),pdc(17%), x2(20%), x4(16%),* leakage saving is there. The reason for not getting leakage saving for other circuits is, for only leakage based decomposition *(w_1=0 w_2=0 w_3=1)* area obtained is quite high and leakage power dissipation increases as area increases. So, in our leakage based decomposition, giving full weight to leakage only, leakage power is high (given in Table 4). It may be noted that for the large circuits like *seq*, BuDDy is unable to find the ROBDD when inputs are taken in the order specified in the circuit.

Next, we present the result of area-power trade-offs achievable in our approach. For this purpose we have varied w_1(area weight), w_2(dynamic power weight) and w_3(leakage power weight) in a range of 0 to 1. Table 3 and 4 present some of the example cases for the values of w_1, w_2 and w_3. In particular we have presented the results for w_1, w_2, w_3 values of *(1,0,0), (0.5,0.5,0), (0,1,0), (0,0.5,0.5), (0,0,1), (0.5,0.25,0.25), (0.5,0,0.5).* Last column in Table 4 shows the maximum CPU time required among all these decompositions. To summarize the results, we have computed the number of circuits having minimum area, switching and leakage value for each weighted decomposition. For example, out of 47 circuits, 23 circuits give minimum area value for the weight *(1,0,0).* Though, average saving in area with respect to initial ROBDD for all 47 circuits for the weight *(0.5,0.5,0*) is also the minimum, number of circuits having minimum area value, in this case, is 16. For other cases, number of circuits having minimum area is further lesser. So, for only area based decomposition, area weight should be 1 and the maximum area saving can be achieved. But switching and leakage power are high in this case. In fact, switching is *16%* higher than its minimum value and leakage is 14% higher than the minimum leakage value. We have also computed some ratios. For getting area ratios, the area value has been divided by the initial area value of the initial ROBDD of each circuit. Similarly, for all the decompositions, area, switching and leakage ratios have been obtained with respect to initial area, switching and leakage of the initial ROBDD. Average of all these ratios have been noted in the second last row of Tables 3 and 4. Also, for clear understanding, average ratios for all the decompositions have been noted in Table 5. From Table 5, it is observed that minimum switching is obtained for the decomposition *(0,1,0).* So, maximum number of circuits results in minimum switching for full weight to switching activity and in this case, area value is only *3%* higher compared to their minimum values. Saving in switching activity for this weight *(0,1,0)* is *43%.* It is interesting to note that for this decomposition, leakage value is also at its minimum. Though it is expected that leakage power dissipation be at its minimum for the weight *(0,0,1)* (decomposition is only leakage based), this is not true. Decomposition for the weight *(0,0,1)* results in maximum increase in area. In fact, with respect to initial ROBDD it requires *1%* more area. Average leakage saving for the weight (*0,0,1*) is *33%* which is not the minimum among all other decompositions. This happens because, as area increases, leakage power also increases, so we cannot get maximum leakage saving for the decomposition *(0,0,1).* As observed from Tables 3and 4, the best results are obtained for the weight *(0, 1, 0)* for which switching and leakage both are at minimum and area is only *3%* higher compared to its minimum value. So, one can choose this weight, where total power dissipation becomes the critical factor.

It may also be noted that for the weight *(0.5, 0.5, 0)* average area is at minimum, switching and leakage results are also good. In this case, switching and leakage values are *9%* and *12%* higher compared to their respective minimum results. So, we can decompose the circuits for weight *(0, 1, 0)* or *(0.5,0.5,0)* for which area, switching and leakage values are within acceptable limits.

However, it must be mentioned that these weight can best be adjusted by the designer based upon the target technology, design constraints, and optimization criterions.

Comparison of area and switching results of our approach with the previous reported work [16] is shown in Table 5. In [16] experiment has been carried out using few circuits, noted in first column of the table. The nodes of a BDD are mapped to adiabatic multiplexers. As a pair of complementary nodes in the BDD have been replaced by single adiabatic multiplexer, number of nodes in the mapped BDD is reduced. Power improvement is due to the usage of adiabatic logic. It may be noted

Table 3. Weighted decomposition results

Circuit	$w_1=1.0$ $w_2=0.0$ $w_3=0.0$			$w_1=0.5$ $w_2=0.5$ $w_3=0.0$			$w_1=0.0$ $w_2=1.0$ $w_3=0.0$			$w_1=0.0$ $w_2=0.5$ $w_3=0.5$		
	Mux	Switching	leakage (nw)	mux	switching	leakage (nw)	mux	Switching	leakage (nw)	mux	switching	leakage (nw)
alu2	200	90.04	241.68	202	90.39	241.91	202	90.39	241.91	203	89.82	240.19
alu4	718	183.49	751.78	718	183.5	751.78	727	281.41	756.21	1255	454.04	1248.3
apex4	964	398.00	1038.9	964	398.0	1038.95	964	398.00	1038.95	1079	445.03	1135.3
apex5	1854	626.47	1662.4	1919	460.6	1251.39	2596	451.55	1250.28	2249	297.89	911.77
aralis	166	32.01	85.74	179	29.34	83.24	188	29.79	85.54	188	29.79	85.54
bw	108	40.72	112.68	109	40.12	115.84	109	40.12	115.84	114	42.19	117.01
b12	67	24.50	72.74	70	25.03	71.78	70	25.03	71.78	77	25.24	70.96
cc	48	18.50	54.08	45	16.95	46.99	42	12.59	38.23	45	13.09	39.64
cht	83	36.75	100.36	90	38.44	105.10	92	37.37	102.77	109	39.73	109.64
clip	94	38.95	110.74	94	38.46	108.96	94	38.46	108.96	165	65.51	175.93
cm138a	16	1.32	3.62	16	1.32	3.62	16	1.32	3.62	16	1.32	3.62
count	168	82.00	202.52	192	88.50	222.41	193	88.81	223.09	142	48.17	125.48
cps	1923	348.55	972.38	1500	208.9	601.06	1973	280.60	691.15	1999	284.28	700.10
c8	100	43.43	113.31	100	43.43	113.31	100	43.43	113.31	106	46.43	123.56
decode	66	24.55	72.18	68	24.61	72.33	78	25.55	74.74	76	26.77	75.64
duke2	398	89.41	251.64	621	84.97	228.54	657	88.91	234.59	786	109.09	292.49
ex1010	1072	302.87	772.97	1072	302.7	772.33	1059	297.85	756.64	1051	293.82	747.69
ex4	563	171.21	487.83	586	214.1	607.36	586	214.10	607.36	590	198.59	564.61
f51m	61	27.13	74.97	61	27.13	74.97	61	27.13	74.97	61	27.13	74.97
frg2	1238	438.39	1261.9	1086	236.5	600.18	1161	229.24	581.36	1304	250.56	631.89
k2	1995	642.35	1697.8	1964	627.7	1658.71	2013	630.09	1668.07	2500	781.17	2019.4
lal	91	29.93	95.49	85	25.08	78.93	94	24.71	72.25	84	23.37	64.77
maskmx	57	23.37	64.43	41	14.97	42.97	37	12.64	36.86	62	18.72	53.59
misex1	36	13.99	37.92	36	13.87	38.22	36	13.87	38.22	40	15.05	40.84
misex2	111	17.55	50.38	92	13.19	38.07	92	13.19	38.07	95	14.12	40.32
misex3c	484	165.71	448.90	491	167.9	454.4	479	159.05	438.01	870	315.92	855.34
pbo2	187	64.46	187.40	187	61.58	179.49	190	61.48	173.91	190	48.55	142.4
pbonew	284	97.84	278.26	284	97.84	278.26	267	88.07	255.74	290	84.85	245.58
pcle	62	23.02	60.27	33	8.35	22.41	33	8.35	22.41	45	9.96	26.39
pcler8	85	33.15	94.70	90	23.56	62.25	90	23.56	62.25	101	24.71	64.74
pdc	667	126.85	359.32	657	117.9	326.22	718	67.15	180.10	948	76.01	191.46
pm1	43	12.43	35.72	43	11.45	36.72	41	8.34	24.17	39	8.04	22.26
sao2	87	23.26	64.57	97	25.98	73.07	97	23.26	65.44	97	23.26	65.44
sct	54	18.75	54.13	59	17.28	49.46	70	18.61	52.11	124	34.35	95.99
seq	2000	569.99	1557.6	2451	665.5	1810.77	3217	649.92	1623.85	3691	567.19	1414.2
shiftc	79	29.96	86.46	79	29.96	86.46	87	32.16	91.94	73	26.39	80.72
sp	585	121.8	336.22	572	108.7	299.85	667	126.82	351.52	720	120.96	330.88
spla	659	128.39	360.05	636	106.55	295.51	621	78.33	213.5	991	84.23	213.73
table3	1349	277.18	724.91	1479	323.4	826.89	810	115.33	314.96	1318	178.4	459.14
table5	742	120.68	331.14	746	118.36	329.16	749	118.21	328.67	1156	215.18	568.41
term1	162	47.03	133.22	138	39.95	112.64	434	76.51	207.43	779	144.73	367.16
ttt2	139	56.87	154.21	131	48.24	139.36	141	52.92	152.31	132	45.57	129.56
vda	752	286.35	766.28	748	278.99	749.79	770	282.48	765.33	1933	779.42	1943.0
vg2	96	28.22	74.66	87	25.36	67.44	87	25.36	67.44	591	176.6	468.48
x1	578	168.94	482.82	588	164.6	471.19	662	178.71	508.53	704	165.59	460.05
x2	30	10.15	29.69	30	10.15	29.69	31	10.09	29.92	38	12.46	37.50
x4	388	141.16	435.55	399	138.08	422.58	437	139.81	395.43	613	190.89	509.32
Average wrt initial	**0.64**	**0.66**	**0.48**	**0.64**	**0.62**	**0.47**	**0.66**	**0.57**	**0.42**	**0.78**	**0.65**	**0.48**
No. of circuits having min. value	**23**	**9**	**11**	**16**	**19**	**16**	**15**	**19**	**17**	**5**	**12**	**9**

that power saving in adiabatic logic can be achieved only during low-frequency operation. However, it should be noted that, experiments in [16] has been carried out using 180 nm technology library. Thus, a direct comparison with our approach is not possible. We have noted the % improvement values reported in [16]. The average percentage saving in area achieved there [16] is much lower than our result. Though the average switching power result in [16] is good, most of the circuits like pcle(76.40%), ttt2(51.12%), vda(84.78%) and x2(64.68%) show higher saving in our approach compared to [16]. Unfortunately, there is no leakage power result available in [16]. So, we are unable to compare leakage result.

Effect on performance: To see the effect on performance maximum levels of *initial*, *inorder* and *weigted* decomposed BDD have been considered. This maximum level depicts the critical path delay of the circuit, as in each level the decomposition is performed around one input variable. The number of levels is same for *initial* and *inorder* decomposed BDD. In our decomposition approach, at each level a suitable variable is selected and after checking the sharing, decomposition is carried out. Hence, critical path delay corresponds to the number of input variables used to decompose the full function. As there is no change in the number of variables, the critical path delay remains unaltered.

Table 4. Weighted decomposition results

Circuit	w_1=0.0 w_2=0.0 w_3=1.0			w_1=0.50 w_2=0.250 w_3=0.250			w_1=0.5 w_2=0.0 w_3=0.5			CPU time (micro Second)
	mux	switching	leakage (nw)	mux	switch ing	leakage (nw)	mux	switching	leakage (nw)	Max-time among all the decompositions
alu2	263	110.17	288.47	202	90.39	241.91	203	89.99	240.79	7.0
alu4	1300	469.15	1231.58	888	331.33	883.70	1142	425.57	1126.93	85.0
apex4	1071	446.89	1143.83	1079	445.03	1135.34	1079	445.03	1135.34	29.0
apex5	2151	740.76	1851.82	1870	372.61	1065.05	1799	528.59	1456.98	2973.0
aralis	255	41.72	115.59	179	29.34	83.24	180	36.34	101.36	2.0
bw	109	40.19	108.27	109	39.85	106.31	109	40.19	108.27	0.001
b12	76	24.94	70.27	73	23.97	70.29	75	26.96	77.07	3.0
cc	47	15.20	42.40	49	17.05	48.73	49	16.55	47.90	3.0
cht	112	39.84	108.53	100	39.59	110.79	116	42.25	120.62	17.0
clip	258	105.43	283.63	113	46.39	128.38	168	68.70	186.80	4.0
cm138a	16	1.32	3.62	16	1.32	3.62	16	1.32	3.62	0.01
count	268	96.81	241.17	121	42.15	112.63	142	48.20	125.48	51.0
cps	2464	500.99	1438.22	1527	209.98	603.62	1470	310.40	830.42	29.4
c8	112	49.43	132.60	106	46.43	121.75	107	46.93	125.06	10.0
decode	74	27.52	77.30	74	26.65	76.55	74	27.52	77.30	1.0
duke2	633	93.40	247.39	626	114.27	308.53	584	76.57	200.31	39.0
ex1010	1076	300.8	767.60	1051	293.82	747.69	1063	297.31	759.40	42.0
ex4	1115	394.53	1192.47	596	198.91	564.76	593	179.7	503.63	1830.0
f51m	66	28.90	87.40	61	27.13	74.97	61	27.13	74.97	1.0
frg2	2292	481.35	1215.99	1138	249.14	631.06	1242	277.01	710.62	30407.0
k2	3991	1386.202	3461.43	2230	689.27	1813.36	2387	753.66	1946.73	4180.0
lal	161	46.80	137.11	93	27.37	79.26	84	23.37	64.80	10.0
maskmx	71	19.77	54.74	42	14.57	42.54	62	19.85	56.38	4.0
misex1	45	17.01	46.55	40	15.05	40.84	43	16.26	42.37	0.01
misex2	119	16.23	45.23	94	13.58	37.79	118	16.74	48.19	7.0
misex3c	1063	396.65	1042.06	547	180.59	495.53	878	319.36	861.80	32.0
pbo2	271	68.55	191.89	190	62.38	184.32	185	47.75	138.96	7.0
pbonew	335	89.65	258.76	275	86.40	250.63	290	86.43	250.76	0.01
pcle	108	32.01	83.78	33	8.35	22.41	76	22.01	58.85	3.0
pcler8	208	64.48	164.69	86	21.73	57.57	101	24.71	64.74	21.0
pdc	1085	118.86	307.09	674	80.34	215.71	935	102.71	272.43	177.0
pm1	44	10.07	28.25	41	10.365	32.34	40	8.45	23.36	2.0
sao2	154	40.35	111.36	110	28.23	80.42	110	28.23	80.42	3.0
sct	147	43.04	115.13	68	18.86	53.97	99	25.55	70.67	5.0
seq	1139	2239.45	5829.20	2920	618.94	1566.13	4337	970.38	2556.73	2225.0
shiftc	79	27.26	79.22	73	26.39	80.72	73	26.39	80.72	1.0
sp	1102	241.92	678.06	597	122.11	335.99	863	159.06	417.76	43.0
spla	1314	228.69	642.8	706	126.79	356.92	867	95.73	246.54	232.0
table3	1391	215.93	548.28	1455	273.66	710.48	1249	171.16	441.64	56.0
table5	1339	233.94	600.51	1295	279.64	742.42	1624	356.69	927.36	186.0
term1	1048	250.23	648.43	245	58.44	158.56	415	106.25	288.04	123.0
ttt2	238	85.55	223.52	149	53.98	151.38	123	44.33	121.77	16.0
vda	4417	1852.02	4606.70	1077	421.38	1088.92	1903	764.48	1910.30	116.0
vg2	1419	459.46	1211.21	87	25.36	67.44	94	27.05	73.67	93.0
x1	1119	277.43	765.42	663	161.55	447.48	672	192.08	532.61	686.0
x2	61	22.37	59.62	32	10.49	30.41	38	12.46	37.58	0.1
x4	809	285.8	754.71	428	144.95	408.38	562	188.20	519.76	1147.0
Average wrt initial	**1.01**	**0.92**	**0.67**	**0.68**	**0.64**	**0.48**	**0.76**	**0.69**	**0.51**	
No. of circuits having min. value	**1**	**1**	**3**	**9**	**12**	**11**	**8**	**5**	**5**	

Table 5. Comparison of our approach to the approach [16]

Circuit	area				switching			
	Initial ROBDD	Our approach (w_1=1 w_2=0 w_3=0)	% saving of our approach	% saving in [181]	Initial ROBDD	Our approach (w_1=0 w_2=1 w_3=0)	% saving of our approach	% saving in [181]
count	219	168	23.29	32.65	91	73.5	2.41	57.22
f51m	70	61	12.86	28.57	30.9	28.9	12.20	54.62
k2	2985	1995	33.16	19.06	902.95	885.95	30.22	48.58
pcle	93	62	33.33	18.6	35.39	27.89	76.40	48.29
ttt2	248	139	43.95	19.35	108.26	95.76	51.12	48.77
vda	4421	752	82.99	17.8	1854.2	1852.02	84.76	47.78
x2	73	30	58.90	17.5	28.57	27.07	64.68	40.59
Averag e % saving			**41.21**	**21.93**			**45.97**	**49.41**

5 Conclusion

The approach described in this paper performs an area-power trade-off in multiplexer based circuit synthesis. It takes both the dynamic and leakage power into consideration, which, to the best of our knowledge is the first ever approach in multiplexer-based realization of multi-output Boolean functions. As the leakage power is much significant at 90 nm or lower technology, the proposed method can be well applied for the leakage aware multiplexer-based realization of Boolean functions in these technologies.

References

[1] Akers, S.B.: Binary decision diagrams. IEEE Trans on Computers C-27(6), 509–516 (1978)

[2] Thompson, Packan, P., Bohr, M.: MOS Scaling: Transistor Challenges for the 21st Century. Intel Technology Journal, Q3 (1998)

[3] Narayanan, U., Leong, H.W., Chung, K., Liu, C.L.: Low Power Multiplexer Decomposition. In: International Symposium on Low Power Electronics and Design, pp. 269–274 (1997)

[4] Thakur, S., Wong, D.F., Krishnamoorthy, S.: Delay Minimal Decomposition of Multiplexers in Technology Mapping. In: Design Automation Conference, pp. 254–257 (1996)

[5] Murgai, R., Brayton, R.K., Sangiovanni- Vincentelli, A.: An improved Synthesis Algorithm for Multiplexer-based PGA's. In: Design Automation Conference, pp. 404–410 (1995)

[6] Rudell, R.: Logic Synthesis for VLSI Design, PhD Thesis, U.C. Berkley (April 1989)

[7] Kim, K., Ahn, T., Han, S.Y., Kim, C.S., Kim, K.H.: Low-power multiplexer decomposition by suppressing propagation of signal transitions. In: IEEE International Symposium on Circuits and Systems, vol. 5, pp. 85–88 (2001)

[8] Anis, M., Elmasry, M.: Multi-Threshold CMOS Digital Circuits Managing Leakage Power, p. 14. Kluwer academic publishers, Dordrecht (2003)

[9] Roy, K., Mukhopadhyay, S., Mahmoodi-Meimand, H.: Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits. Proceedings of the IEEE 91(2), 305–327 (2003)

[10] Roy, K., Prasad, S.C.: Low-Power CMOS VLSI Circuit Design. John Wiley & Sons, Inc., Chichester (2000)

[11] Lindgren, P., Kerttu, M., Thornton, M., Drechsler, R.: Low Power Optimization Technique for BDD Mapped Circuits. In: Asia South Pacific Design Automation Conference, pp. 615–621 (2001)

[12] Satyanarayana, D., Chattopadhyay, S., Sasidhar, J.: Low Power Combinational Circuit Synthesis targeting Multiplexer based FPGAs. In: Proceedings of 17th International Conference on VLSI Design (2004)

[13] BDD packages BuDDY, `http://www.itu.dk/research/buddy`

[14] Computer-Aided benchmarking Laboratory, `http://www.cbl.ncsu.edu/benchmarks`

[15] Lee, D., Blaauw, D., Sylvester, D.: Runtime Leakage Minimization through probability-Aware Dual Vt or Dual-Tox Assignment. In: Asia-Pacific Desig Automation Conference, pp. 399–404 (2005)

[16] Paul, G., Pradhan, S.N., Bhattacharya, B.B., Pal, A., Das, A.: BDD-based synthesis of logic functions using adiabatic multiplexers. International Journal on Systemics, Cybernetics, and Informatics (IJSCI) 1, 44–49 (2006)