

AN ENTROPY-BASED ALGORITHM TO REDUCE AREA OVERHEAD FOR BIPARTITION-CODEC ARCHITECTURE

Po-Hung Chen, Shanq-Jang Ruan, Kuen-Pin Wu, Dai-Xun Hu, Feipei Lai, Senior Member, IEEE, Kun-Lin Tsai

Dept. of Electrical Engineering &

Dept. of Computer Science and Information Engineering

National Taiwan University Taipei 106, Taiwan.

flai@cc.ee.ntu.edu.tw

ABSTRACT

Bipartition-codec scheme has been used as one of the effective power reduction techniques in logic-level circuit design. It treats each output value of a combinational circuit as one state of an FSM, and extracts the most actively transitive states (output) and the corresponding input to build a subcircuit. After bipartitioning the circuit, the encoding technique is used to encode the highly active subcircuit for further power reduction. Although we can get a large amount of power reduction in the previous proposed bipartition algorithm, the area overhead is considerably large. In this paper, we propose an effective heuristic algorithm based on entropy, which offering a theoretical area model to resolve the area overhead problem in the bipartition-codec architecture. The experimental results show that the area can be averagely reduced by 16% with 1.8% marginal power increase compared to the previous proposed probabilistic-driven algorithm [2][3].

1. INTRODUCTION

In order to obtain power saving, much work has been done on dynamic power management. The technique is based on selectively disabling the input registers of logic circuits when some input conditions are met. The bipartition architecture treats each output value of a combinational circuit as one state of an FSM and the most transitive states will be extracted to build a small subcircuit [1]. The power saving is based on the observation that most time the small block is active and the big one is idle. Ruan, *et al.* further exploited the advantage of bipartition with the state encoding concept and named it bipartition-codec architecture [2][3].

On the other hand, power estimation always relates to low power design. Because it helps in selecting the one that is potentially more effective from the power standpoint, especially in the high level of abstraction. There are rich literatures on the high level and logic level power estimation. For instance, [4], [5], [6] presented a technique for estimating power based on entropy from information theory to formulate the relationship among input, output patterns and circuit area. It implies we can estimate the average switching frequency inside a combinational circuit by giving only input/output Boolean functional description.

This paper is an extended work of [2][3]. An entropy-based algorithm to reduce area overhead in bipartition-codec

architecture is proposed. The area model of the algorithm is estimated in terms of the amount of computation, which is based on concept of entropy from information theory, to formulate the relationship among input, output patterns and circuit area. To verify our results, our synthesis flow starts from the PLA specification to transistor-level implementation. We employed an accurate switch-level simulation to estimate the power by EPIC powermill. The experimental results show that area can be averagely reduced by 16% with 1.8% marginal power increase compared to the previous proposed probabilistic-driven algorithm [2][3]. Hence, based on the proposed entropy-based algorithm, we not only obtained the power saving, but also significantly reduced the area overhead in the bipartition-codec architecture [2][3].

The remainder of this paper is organized as follows. Section II presents the bipartition-codec architecture and its working principle. The entropy-based bipartition algorithm is described in section III. Experimental results are presented in section IV. Compared with the previous work, the proposed algorithm can obtain power reduction with less area increase. Finally, conclusions of the paper are presented in section V.

2. BIPARTITION CODEC ARCHITECTURE

The bipartition-codec architecture proposed in [2][3] treats each output vector of a combinational circuit as a state in an FSM. When the output vectors of the circuit cluster only around a few special output vectors, the probabilistic-driven algorithm transforms the circuit into two subcircuits: one is small with highly active probability and the other is big with low active probability. Under certain input vectors, the circuit switching activity will be confined to the small subcircuit with highly active probability. Consequently, encoding technique is applied to the most active subcircuit for further reducing the power consumption.

The power dissipation benefit of bipartition-codec architecture is due to the following three reasons: first the length of the registers, which are used to store the output of each stage, is reduced after encoding. Second, the Hamming distance of the register switching is smaller than before. Finally, the circuit switching activity of the combinational block is also reduced due to the bipartition. For the details of bipartition and bipartition-codec architectures, please refer to [1][2].

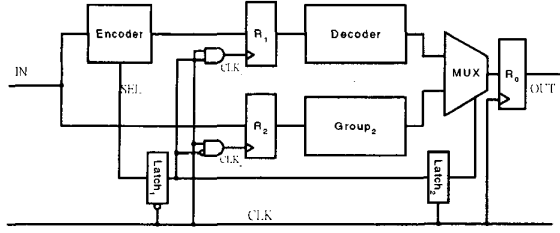


Figure 1. bipartition-codec architecture.

Given a logic function, we wish to apply the bipartition-codec architecture to implementation with an efficient algorithm not only for lower power dissipation but also lower area overhead. First, we should estimate the area cost of each bipartition combination and then choose the best combination as our solution for the bipartition-codec architecture. Thus an area estimation model is required. Some researches have been done on the relationship between circuit area complexity and its entropy (H), including [4-7]. For speed and efficiency considerations, we adopt the area estimation model proposed by Cheng *et al.* [7]. Using this model, we can estimate the area complexity of a circuit by its input, output and entropy.

To illustrate our algorithm, we first describe the area estimation model in the beginning of the next section.

3. TWO PHASE ENTROPY-BASED ALGORITHM

3.1 Entropy of a multiple output function

3.1.1 Fully Specified Functions

For a multiple-output, fully specified Boolean function f with n input and m output signals, there are 2^n input patterns and at most 2^m different output patterns. For each output pattern e_i , the probability that $f = e_i$ is [7]

$$P(f = e_i) = P_i = \frac{n_{e_i}}{2^n}, \quad (1)$$

where n_{e_i} is the appearance count of e_i in the truth table. The entropy H of f is a function of $P = P_1, P_2, \dots, P_{2^m}$ and is defined as:

$$H(P) = \sum_{i=1}^{2^m} P_i \cdot \log_2 \frac{1}{P_i}. \quad (2)$$

The value of $H(P)$ is always between 0 and m . When $P_1 = P_2 = \dots = P_{2^m} = 2^{-m}$, $H(P) = m$. When $P_i = 1$ and $P_j = 0$ for all $j \neq i$, $H(P) = 0$.

3.1.2 Partially Specified Functions

Given a partially specified function f_d with n_d don't cares terms in its truth table. The probability that $f_d = e_i$ is then altered as:

$$P(f_d = e_i) = P_i = \frac{n_{e_i}}{2^n - n_d}. \quad (3)$$

The calculation of the entropy of f_d is the same as that for fully specified functions using Eqn. (2).

In the information theory, the statistical behavior of a digital circuit is modeled by its entropy. A digital circuit transforms input signals into its output signals. This transformation may be considered as computational work. The computational work of an n -input digital combinational circuit is defined as:

$$\text{Computational Work} = 2^n \cdot H(P). \quad (4)$$

It assumes that the circuit area, which is proportional to the amounts of logic gates, should be also proportional to the computational work. In the following, we will use the literal count, represented by L , in a multiple-level minimized form as a measure of area complexity. The area complexity estimation model of a logic circuit is defined as:

$$L(n, d, P) = (1 - d) \cdot k \cdot 2^n \cdot H(P), \quad (5)$$

where n is the number of input signals, d is the fraction of don't care terms in the truth table, k is a proportional constant that varies with different n , and $H(P)$ is given by Eqn. (2). Total area cost is modeled as follows:

$$\begin{aligned} \text{TotalArea} &= \text{area}_{E1} + \text{area}_{D1} + \text{area}_{G2} + \text{overhead}() \\ &= L_{E1}(n_{E1}, d_{E1}, P_{E1}) + L_{D1}(n_{D1}, d_{D1}, P_{D1}) + \\ &\quad L_{E1}(n_{G2}, d_{G2}, P_{G2}) + \text{overhead}(), \end{aligned} \quad (6)$$

where area_{E1} , area_{D1} and area_{G2} represent the area of Encoder, Decoder and Group₂ after being synthesized by SIS, respectively. The last term overhead() in the formula consists of one $\lceil \log_2 n \rceil$ -bit register, one n -bit register, one m -bit register, one m -tuple 2-to-1 multiplexer and two 1-bit latches. Note that, although the input number of Decoder is much less than that of Encoder or Group₂, we apply the same k to Decoder, Encoder and Group₂ due to the negligible area complexity of Decoder compared with the other two blocks. Furthermore, since the same k is applied to all the three of them, we can even reduce the area complexity model to:

$$L(n, d, P) = (1 - d) \cdot 2^n \cdot H(P). \quad (7)$$

Then we can use Eqn. (7) in our algorithm as the cost function and find the minimum cost partition. Note that, although Eqn. (7) is not an accurate model to estimate the real area, it can be applied to making a comparison of area complexity among different low power schemes. In the next paragraph, we describe the proposed algorithm.

First, we can perform a pre-selection procedure to find a set of states as candidates for phase II. Second, we apply the area cost model to a greedy-selection algorithm for the appropriately arranging states of Encoder, Decoder and Group₂ with minimum area cost. We assume that the total number of states is w . The heuristic bipartition algorithm works as follows:

Phase I preselection

In our proposed algorithm, we have to find a set of states as the candidates for phase II. We define the pre-selection rule according to Ruan's probability-driven algorithm [2][3]. It assumes that clustering the higher probability states into a small

circuit implies higher power reduction. Hence, we sort the output states according to the state probability. Two variables **avg_prob** (average state probability) and **min_prob** (minimum state probability) are defined as follows:

$$\text{avg_prob} = 1/w,$$

$$\text{min_prob} = 1/2.$$

We then select the states whose probabilities are greater than *avg_prob* as the candidates for phase II. If the sum of state probabilities of these candidates is less than 1/2, we add one more state into the candidates and check again until the summation is larger than or equal to 1/2. The algorithm of phase I (pre-selection phase) is shown below:

```

Preselect ( $S = \{s_1, s_2, \dots, s_w\}$ )
{
  min_prob = 1/2;
  avg_prob = 1/w;
  candidates =  $\Phi$ ;
  probability = 0;
  for each ( $s_i \in S$ )
  {
    if (probability( $s_i$ ) > avg_prob)
    {
       $S = S - \{s_i\}$ ;
      candidates = candidates  $\cup \{s_i\}$ ;
      probability = probability + probability( $s_i$ );
    }
  }
  while(probability  $\leq$  min_prob)
  {
    new_state = sel_max_count(S);
    candidates = candidates  $\cup \{new\_state\}$ ;
    probability = probability + probability(new_state);
     $s = s - \{new\_state\}$ ;
  }
  return candidates;
}

```

Phase II greedy selection

After choosing a set of states as the candidates, we continue to phase II greedy selection. To determine the area cost we have the following steps.

1. Cluster all states of the candidates selected during phase I in Group₁, leaving the remainder in Group₂. [1]
2. Encode Group₁ to get two circuits, Encoder and Decoder. [2][3]
3. Apply the area model of Eqn. (7) and Eqn. (8) into the three partial circuits Encoder, Decoder, and Group₂ and sum up their areas into TotalArea, which will be considered as area cost.

There are three variables *Minimal*, *Candidate* and *Working* in our algorithm. All of these variables are sets of states. *Minimal* keeps the states in Encoder, Decoder and Group₂ such that the architecture has minimal area; *Candidate* contains the states that may be placed in Encoder, Decoder and Group₂; and *Working* is an auxiliary set variable. Initially, the three variables are all

empty sets. To find the minimum cost solution quickly, we apply the following greedy selection method:

```

greedyselect(candidates)
{
  while (candidate  $\neq \Phi$ )
  {
    for (each  $s_i \in$  candidates and all other  $s_j (\neq s_i) \in$  candidates)
    {
      if (area(working  $\cup \{s_i\}$ )  $\geq$  area(working  $\cup \{s_j\}$ ))
      {
        working = working  $\cup \{s_i\}$ ;
      }
    }
    if minimal =  $\Phi$  or area(working) < area(minimal)
    {
      minimal = working;
    }
    candidates = candidates -  $s_l - s_i$  for all  $l$ , where  $p_l > p_i$ ;
  }
  return minimal;
}

```

To verify the performance of the algorithm, we make an experiment on it with various MCNC benchmarks, and the experimental results are shown in the next section.

4. EXPERIMENTAL RESULT

The entropy-based bipartition algorithm has been implemented in C++ on a SUN Sparc station. We used SIS [8] to synthesize our partition results and estimated the power by PowerMill. Several random logic circuits taken from MCNC PLAs are used to demonstrate our algorithm. In the experiment, 5v supply voltage and a clock frequency of 20MHz were assumed. The severe script of SIS was used to optimize the benchmarks.

Table I presents the power dissipation of the probabilistic-driven algorithm and our entropy-based bipartition algorithm for the bipartition-codec architecture on a subset of the MCNC PLAs, respectively. The original column is the total power (**TP**) implemented by conventional architecture. The **TP** and **PR%** of the probabilistic-driven and the entropy-based bipartition algorithm columns are the total power dissipation and power improvement computed as $100(\text{Power}_{\text{original}} - \text{Power}_{\text{bipartition-codec}}) / \text{Power}_{\text{original}}$. The area of original and bipartition-codec architectures implemented by both algorithms is tabulated in Table II. **TA** represents the total area size implemented by original, probabilistic-driven and our entropy-based bipartition algorithms in the table. The percentage of area increased **AI%** is computed as $100(\text{Area}_{\text{bipartition-codec}} - \text{Area}_{\text{original}}) / \text{Area}_{\text{original}}$.

From Tables I and II we can observe that over half of the MCNC benchmarks, power reduction of the proposed algorithm are larger than or equal to that of the probability-driven algorithm [2][3], but all of the area of the proposed algorithm are less than that of probability-driven algorithm.

As a result, while implementing the bipartition-codec architecture, our algorithm can significantly reduce area with almost the same power improvement as in [2][3].

5. CONCLUSION

Bipartition-codec architecture is advantageous when few output occurred frequently in a combinational circuit, but the increased

circuit area significantly decreases the practicability of the circuit. Moreover the larger the chip size, the higher the chip cost. In this paper, we have proposed an effective entropy-based bipartition algorithm to keep the low power dissipation and minimize the area so that the area is less than that of previous work [2][3].

In our algorithm, we first select the candidate states in the order of state probability and then we employ a greedy procedure to select the states from some candidates for minimizing the area, treating each output vector of a combinational block in a combinational circuit as a state in an FSM. Here we estimate the area size with the statistical behavior of each partition block, which is characterized by its entropy. The experimental results show that the proposed algorithm has done a better work on reducing the area overhead while maintaining competitive power consumption in the bipartition-codec architecture than the previous work [2][3].

6. REFERENCES

- [1] S.-J. Chen, R.-J. Shang, X.-J. Huang, S.-J. RUAN, and F. Lai, "bipartition and synthesis in low power pipelined circuits," IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E81-A, no. 4, pp. 664-671, Apr. 1998.
- [2] S.-J. Ruan, R.-J. Shang, S.-J. Chen, X.-J. Huang, and F. Lai, "A bipartition-codec architecture to reduce power in pipelined circuits," in Proc. of IEEE/ACM Int. Conf. on Computer Aided Design, pp. 84-89, Nov. 1999.
- [3] S.-J. Ruan, R.-J. Shang, S.-J. Chen, X.-J. Huang, and F. Lai, "A bipartition-codec architecture to reduce power in pipelined circuits," submitted to IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (accepted).
- [4] M. Nemani and F. N. Najm, "High-level area and power estimation for VLSI circuits," IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, No. 6, pp. 697-713, Jun. 1999.
- [5] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation and optimization," IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol. 17, pp. 1061-1079, Nov. 1998.
- [6] M. Nemani and F. N. Najm, "Towards a high-level power estimation capability," IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, No. 6, pp. 588-598, Jun. 1996.
- [7] K.-T. Cheng and V. D. Agrawal, "An entropy measure for the complexity of multi-output boolean functions," in Proc. of 27th ACM/IEEE Design Automation Conf., pp. 302-305, 1990.
- [8] SIS: A System for Sequential Circuit synthesis is implemented by Electronics Research Laboratory in Department of EE and CS, University of California, Berkley, 4 May 1992.

| Table I The simulation results for power dissipation | | | | | |
|--|--------|--------------------------------|------|---------------|------|
| Circuits | Origin | Bipartition-codec architecture | | | |
| | | Probability-driven [2] | | Entropy-based | |
| | TP | TP | PR% | TP | PR% |
| bw | 4443 | 3908 | 12 | 3699 | 17 |
| sao2 | 3361 | 1255 | 63 | 1392 | 59 |
| con1 | 1506 | 1072 | 29 | 1176 | 22 |
| misex1 | 2238 | 1555 | 31 | 1474 | 34 |
| rd84 | 3176 | 2649 | 17 | 2920 | 8 |
| rd73 | 2362 | 2230 | 6 | 2316 | 2 |
| cm82a | 1249 | 858 | 31 | 774 | 38 |
| table5 | 7946 | 5748 | 28 | 6831 | 14 |
| cm85a | 2535 | 1757 | 31 | 1757 | 31 |
| cm138a | 1299 | 445 | 66 | 445 | 66 |
| cm163a | 3573 | 2688 | 25 | 2688 | 25 |
| inc | 2840 | 2856 | -0.6 | 2712 | 5 |
| t | 1007 | 477 | 53 | 501 | 50 |
| cu | 3144 | 1470 | 53 | 1538 | 51 |
| C17 | 1005 | 519 | 48 | 500 | 50 |
| cm42a | 839 | 803 | 4 | 795 | 5 |
| x2 | 2442 | 1646 | 33 | 1919 | 21 |
| rd53 | 1499 | 1067 | 29 | 1112 | 26 |
| decod | 1313 | 889 | 32 | 959 | 27 |
| cm162a | 3109 | 2369 | 24 | 2369 | 24 |
| cmb | 3265 | 811 | 75 | 811 | 75 |
| | | average | 32.7 | average | 30.9 |
| *Power is in μW | | | | | |

| Table II The simulation results for area size | | | | | |
|---|--------|--------------------------------|-------|---------------|-------|
| Circuits | Origin | Bipartition-codec architecture | | | |
| | | Probabilistic-driven [2] | | Entropy-based | |
| | TA | TA | AP% | TA | AP% |
| bw | 694 | 1060 | 53 | 941 | 36 |
| sao2 | 571 | 725 | 27 | 675 | 18 |
| con1 | 209 | 384 | 84 | 300 | 44 |
| misex1 | 340 | 441 | 30 | 399 | 17 |
| rd84 | 531 | 774 | 46 | 714 | 35 |
| rd73 | 370 | 740 | 100 | 605 | 64 |
| cm82a | 178 | 417 | 134 | 367 | 106 |
| table5 | 3203 | 4312 | 35 | 4145 | 29 |
| cm85a | 383 | 312 | -18.5 | 312 | -18.5 |
| cm138a | 217 | 248 | 14 | 248 | 14 |
| cm163a | 475 | 716 | 51 | 716 | 51 |
| inc | 463 | 797 | 72 | 726 | 57 |
| t | 132 | 244 | 85 | 193 | 46 |
| cu | 485 | 637 | 31 | 593 | 22 |
| C17 | 132 | 223 | 69 | 193 | 46 |
| cm42a | 186 | 357 | 92 | 307 | 65 |
| x2 | 371 | 533 | 44 | 482 | 30 |
| rd53 | 216 | 395 | 83 | 356 | 65 |
| decod | 231 | 436 | 89 | 390 | 69 |
| cm162a | 444 | 654 | 47 | 654 | 47 |
| cmb | 440 | 435 | -1.1 | 435 | -1.1 |
| | | average | 56 | | 40 |
| *area is in $128 \mu m^2$ | | | | | |