TECHNICAL NOTE

AN EFFICIENT ALGORITHM FOR OUTPUT CODING IN PAL-BASED CPLDS

D. Kania

Institute of Electronics, Silesian Technical University
ul. Akademicka 16, 44-100 Gliwice, Poland, kdariusz@zeus.polsl.gliwice.pl

(Received: March 16, 2001 – Accepted: May 15, 2002)

Abstract One of the approaches used to partition inputs consists in modifying and limiting the input set using an external transcoder. This method is strictly related to output coding. This paper presents an optimal output coding in PAL-based programmable transcoders. The algorithm can be used to implement circuits in PAL-based CPLDs.

Key Words Partitioning, Coding, Decomposition, Programmable Logic Devices

چکیده یکی از راههای دسته بندی ورودیها، اصلاح و محدود کردن دسته ورودی با استفاده از یک رمز دهنده خارجی است که اکیدا به رمز گذاری خروجی مربوط است. این مقاله روش بهینه رمز گذاری خروجی در رمز گذار قابل برنامه نویسی بر پایه PAL را ارائه می دهد. الگوریتم ارائه شده می تواند برای بکارگیری مدار در CPLD پایه PAL استفاده شود.

1. INTRODUCTION

One of the methods of reducing the number of required module inputs is input coding [1,3]. The issue of coding becomes essential if a programmable PAL structure is to be used as an external transcoder [2]. PAL-based programmable chips have a limited number of terms connected to their OR gates. If a binary coding is used, due to uneven term allocation, only 2k+1 different words can be coded (k is the number of terms). It is of course possible to expand the number of terms and increase the number of code words by using additional outputs. However, there is a need for a different coding which will evenly use the products connected to the OR gates. What kind of coding can use the terms evenly? How many different words can be coded using m-outputs, if k-products are connected to every output? The present paper is an attempt to answer these questions.

2. THEORETICAL BACKGROUND

Let y be defined over the set I={I11,...,I1,I0} (Column A; Table 1). Let us try to find a partition of the function arguments using an 8-input transcoder.

Let w_k $(I_{s-1},...,I_{p+1},I_p)$ be the number of different words formed by the input variables $I_{s-1},...,I_{p+1},I_p$. If a set X1 exists such that X1⊂I and $w_k(X_1) < 2^{(\overline{\overline{X_1}}-1)}$, then it is possible to limit the number of inputs by using an external transcoder. If we assume that the transcoder has n_t-inputs, then our search for X1 will of course be limited to such subsets for which $\overline{\overline{X_1}} \le n_t$.

In our case n_t=8, so our search will begin with those subsets for which $\overline{\overline{X_1}}$=8. A subset X1={I11,I10,I8,I7,I5,I4,I2,I1} meeting all the conditions does indeed exist; it is therefore possible to propose the partitioning presented in Table 1 and Figure 1.

We shall now try to implement the transcoder in a programmable structure, in which 3 terms are connected to each output structure (e.g. a CPLD macrocell of the MAX 5000 series of devices by Altera). In what follows it will be assumed, for

TABLE 1. PLA Files Defining The Function y Before (y.pla) and After Argument Partitioning.

y.pla file ***A***	***transcoder*** ***B***	***y function circuits*** ***C***
.i 12 .o 1 .ilb I11,I10,I9,I8,I7,I6,I5,I4,I3,I2,I1,I0 .ob y .p 21 000000000000 1 100110110001 1 001001000001 1 100111111000 1 000001001000 1 111110111000 1 001000000110 1 111100001000 1 001001001111 1 111101000000 1 000001011011 1 110101110001 1 000001101100 1 111100111001 1 001001100100 1 110110111101 1 000100010010 1 111110110101 1 000101011011 1 111111111100 1 001101010011 1 .e	.i8 .o4 .ilb I11,I10,I8,I7,I5,I4,I2,I1 .ob a3,a2,a1,a0 .p10 00000000 0001 00000011 0010 00000101 0011 00001010 0100 00100101 0101 10111100 0110 11111100 0111 11100000 1000 11101100 1001 11111110 1010 .e	.i8 .o1 .ilb I9,I6,I3,I0,a3,a2,a1,a0 .ob y .p21 00000001 1 00010110 1 11010001 1 01100110 1 01100001 1 10100111 1 10000010 1 10101000 1 11110010 1 11001000 1 01110011 1 01011001 1 01100100 1 10111001 1 11000100 1 00111010 1 00000101 1 10011010 1 01110101 1 11101010 1 11010101 1 .e

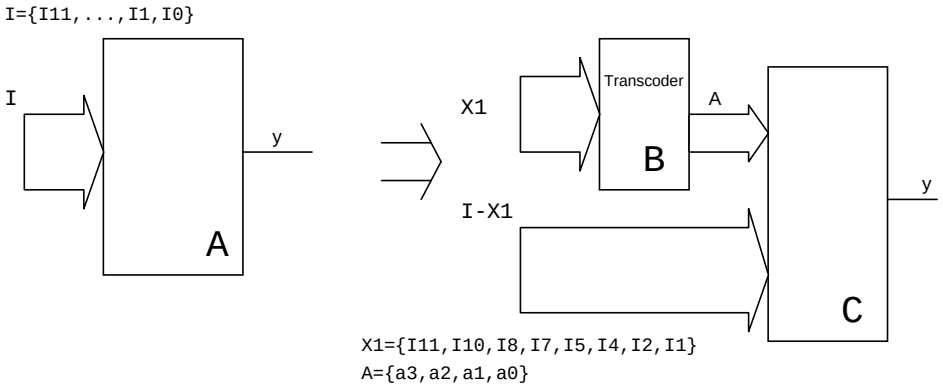

Figure 1. The example of partitioning the inputs using an external transcoder.

simplicity, that the outputs of the programmable transcoder are high active. Under this assumption, a "1" in a code word means that a term was used. It is not possible to implement the transcoder directly in such macrocells. It is necessary to expand the number of terms, which leads to using additional outputs. The resulting transcoder structure and the corresponding file are presented in Figure 2.

Can the individual words be coded better, using fewer outputs?

3. OUTPUT CODING IN PAL-BASED PROGRAMMABLE STRUCTURES

Programmable circuits have limited internal resources. Let k_t be the number of code words that can be obtained in a given programmable transcoder, m - the number of transcoder outputs and k - the number of terms connected to each output. Optimal coding means using a minimal number of terms while keeping this number evenly distributed among the individual outputs. Let us consider what maximum number k_t of code words can be obtained using a circuit with given parameters (m,k). To use the terms optimally, we should first use all the combinations "0 of m", "1 of m", "2 of m" etc. active outputs. Each of the blocks "i of m", where $0 \le i \le m$, makes use of a constant number of terms connected to each output (Table 2). If, for a given programmable transcoder, there exists a

```
.i11
.o7
.ilbI11,I10,I8,I7,I5,I4,I2,I1,a2',a1',a0'
.ob a2',a1',a0',a3,a2,a1,a0
.p20
00000000---  0010000
00000101---  0010000
00100101---  0010000
----------1  0000001
11111100---  0000001
11101100---  0000001
00000011---  0100000
00000101---  0100000
10111100---  0100000
---------1-  0000010
11111100---  0000010
11111110---  0000010
00001010---  1000000
00100101---  1000000
10111100---  1000000
--------1--  0000100
11111100---  0000100
11100000---  0001000
11101100---  0001000
11111110---  0001000
.e
```

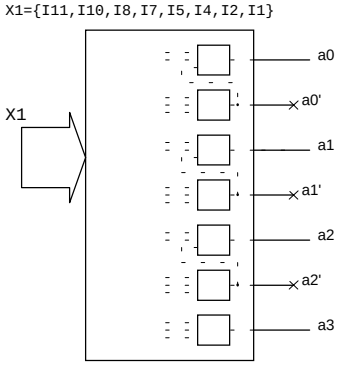

Figure 2. Description and structure of a PAL-based programmable transcoder after term expansion.

number j such that $\sum_{i=0}^{j}\binom{m-1}{i}=k$ and $w_k(X1)<\sum_{i=0}^{j+1}\binom{m}{i}$, then the coding algorithm is relatively simple. It consists in choosing an arbitrary set of combinations "i of m"; this set will have $w_k(X1)$+1 elements, where 0 ≤ i ≤ j+1. The algorithm becomes more complicated if no number j meets the condition $\sum_{i=0}^{j}\binom{m-1}{i}=k$. In this case, we can present the concept of optimal coding differently. In the first step we find a value of j, which meets the inequalities $\sum_{i=0}^{j-1}\binom{m-1}{i}<k<\sum_{i=0}^{j}\binom{m-1}{i}$. Now a part of the required number of words can be obtained using all the combinations "i of m", where 0 ≤ i ≤ j. This step yields $\sum_{i=0}^{j}\binom{m}{i}$ different combinations. When these have been coded, the transcoder still has a certain number $k_r=k-\sum_{i=0}^{j-1}\binom{m-1}{i}$ of unused terms at each output. These terms can be used to code more words. With k_r unused terms connected to every OR gate we can code a maximum number of "j+1 of m" combinations. The total number of terms used by every combination is j+1. With m outputs and k_r unused terms at every output we can code a maximum of $\left[\frac{mk_r}{j+1}\right]$ additional words. The total number of possible words is therefore $\sum_{i=0}^{j}\binom{m}{i}+\left[\frac{mk_r}{j+1}\right]$. The above allows us to state that

$$k_t=\sum_{i=0}^{j}\binom{m}{i}+\left[\frac{m(k-\sum_{i=0}^{j-1}\binom{m-1}{i})}{j+1}\right] \quad (1)$$

where j is a number satisfying the inequality

$$\sum_{i=0}^{j-1}\binom{m-1}{i}<k\le\sum_{i=0}^{j}\binom{m-1}{i} \quad (2)$$

Assume that we want to code $w_k(X1)$=10 words using a PAL-based programmable transcoder with k=3 products per output Table 1and Figure 1. It is possible to code $w_k(X1)$ words if the coefficient of the transcoder is $k_t>w_k(X1)$. From Equation 1, we can determine the number of outputs necessary to code a given number of words. Knowing the values of m and k we can determine the parameter j meeting Equation 2. The creation

```
.i11
.o7
i8
.o4
.ilb I11,I10,I8,I7,I5,I4,I2,I1
.ob a4,a3,a2,a1,a0
.p10
00000000 00001
00000011 00010
00000101 00100
00001010 01000
00100101 10000
10111100 00011
11111100 01100
11100000 10001
11101100 00110
11111110 11000
.e
```

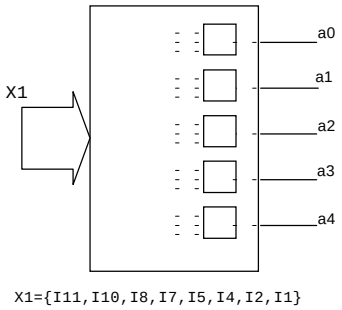

Figure 3. Description and structure of a PAL-based programmable transcoder after optimal coding algorithm.

of code words occurs in two stages. In the first stage, all "i of m" combinations are generated, where $0 \leq i \leq j$. In the second stage, the remaining words are coded as "j+1 of m" combinations. The first stage requires no special explanations. The purpose of the second stage is to find a maximum number of "j+1 of m" combinations when k_r terms per output are available. The optimal PAL-based programmable transcoder structure is presented in Figure 3.

4. CONCLUSIONS

The word-coding algorithm presented above was implemented in a program Decomp assisting the decomposition of combinational circuits. In the present paper the algorithm was described for structures with an equal number of terms connected to each output and with high-active outputs. It is, however, also directly applicable to output coding in circuits with varied number of terms and programmable output type (e.g. 22V10).

5. REFERENCES

1 Devadas, S., and Newton, R., "Exact Algorithm for Output Encoding, State Assignment, and Four-Level Boolean Minimization", ***IEEE Transactions on Computer-Aided Design***, Vol. 10, No. 1, (1991), 13-27.

2 Kania, D., "Coding Capacity of PAL-Based Logic Blocks Included in CPLDs and FPGAs", ***IFAC Workshop on Programmable Devices and Systems***, PDS 2000, Ostrava, February 8-9, ***Published for the IFAC by PERGAMON, An Imprint of Elsevier Science***, (2000), 164-169.

3 Malik, S., Lauter, U., Brayton, R., K., and Vincentelli, A., S., "Symbolic Minimization of Multilevel Logic and the Input Encoding Problem", ***IEEE Transactions on Computer-Aided Design***, Vol. 11, No. 7, (1992), 825-843.