THE PRINCIPLES OF THE OPTIMIZATION OF CONTROL UNITS ON THE COUNTERS

Barkalov A.A.

Computer Science Departement, Donetsk State Technical University
barkalov@cs.dgtu.donetsk.ua

Abstract

Barkalov A.A. The principles of the optimization of control units on the counters. This paper considers the principles of optimization of control unit based on application of the splitting of information about object. As object can be the code of the automaton state or address of microinstruction. The main idea is to represent such information as concatenation of code of object and code of linear chain of objects. It permits to apply the methods of optimization of Moore automaton and decrease the amount of hardware in the control unit's circuit.

Introduction

One of the central part of any digital system is the control unit (CU), which implements an algorithm of the system's operation[1]. Control unit can be implemented as automaton with hardwired logic (AHL) [2], or automaton with programmable logic (APL) [3], or as compositional microprogram control unit (CMCU), which is composition of the AHL and APL[4]. Now programmable logical devices(PLD) are extensively used for implementation of the logical circuits of control units [5]. One of the important tasks in the design of control units is minimization of the amount of PLDs in its logical circuit. One of the ways to solve this problem is usage of the counter to keep the code of the automaton's state or microinstruction's address. It permits to decrease the amount of VLSI in the circuit of the excitation functions' formation[6]. In our article the principles of the optimization of the amount of PLDs are discussed which are mutual ones for APL, AHL and CMCU on counters.

1. Main definitions and algorithm of synthesis

Let control algorithm is represented as flow-chart $\Gamma=\Gamma(B, E)$, where B is the set of nodes, E is the set of arcs. The subsets Y_t of the set of microoperations $Y=\{y_1, ..., y_N\}$ are written in the operational nodes, the elements of the set of the logic conditions $X=\{x_1, ..., x_L\}$ are written in the logical nodes of flow-chart. Let us introduce the next definitions.

<u>Definition 1</u>. An object $o_m \in O=\{o_1, ..., o_M\}$ is state of the automaton with hardwired logic or microinstruction of the automaton with programmable logic or compositional microprogram control unit.

Definition 2. A linear sequence of objects(LSO) of the flow-chart Γ is finite vector α_g=<o_{g1}, ..., o_{gFg}>, such as for any pair of objects o_{gi}, o_{gi+1}, where i is the component number of the vector α_g, there is transition <o_{gi}, o_{gi+1}>.

A transition <o_i, o_j> is the path from operational node corresponding to the object o_i in the operational node corresponding to the object o_j, passing through some subset of conditional nodes(this subset may be empty one).

Definition 3. An object $o_q \in o^g$ is named an input of LSO, where o^g is the set of the components of LSO α_g, if there is transition <o_s,o_q>, such as $o_s \notin o^g$ or o_s is output of LCO α_g.

Definition 4.An object $o_q \in o^g$ is named an output of LSO α_g, if there is transition <o_q, o_s>, such as $o_s \notin o^g$ or $o_s \in o^g$ and if it has the component number less then the number of component o_q.

Let we determine for flow-chart Γ the set C={α_1, ..., α_G}. which is partition of the set of the flow-chart's objects O={o_1, ..., o_M} on LSO's. Let natural encoding of objects is fulfilled inside each LSO accordingly with condition

$$K(o_{gi+1}) = K(o_{gi}) + 1, \qquad (g = \overline{1, G}; i = \overline{1, F_g - 1}), \qquad (1)$$

where $K(o_m)$ is code of object $o_m \in O$, corresponding to the code of the state of AHL or microinstruction address of APL or CMCU. Now automaton can be represented as two combinational circuits KC1, KC2 and counter CT (Figure 1). Let us denote such automaton by the symbol U_1. An automaton U_1 operates in the next manner.

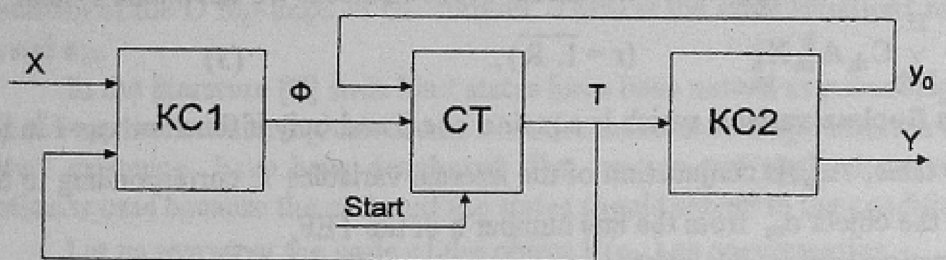

Figure.1.The structure of an automaton on the counter

By the signal «Start» the code of the flow-chart's initial object is written in counter CT. Let in the instant of time t (t=0, 1, 2,...) code $K(o_{gi})$ of the i-th object of the LSO $\alpha_g \in C$ is contained in counter. If object o_{gi} is not the output of the LSO α_g, then circuit KC2 forms microoperations Y_t and signal y_0 to increment counter. It is way to execute unconditional transition between the objects of the same LSO. If object o_{gi} is the output of the sequence α_g, then signal y_0 is not formed and the source of data for counter is circuit KC1 which forms the functions Φ. Excitation functions of the automaton memory Φ={φ_1, ..., φ_R}, where R=int($\log_2 M$), form in the counter CT an address of the current LSO. Performance is finished when the final object of flow-chart will be reached. The codes of the objects are identified by the internal variables from the set T={T_1, ..., T_R}.

To design such device U_1 the next method is proposed.

1. Formation of the set of the objects O. On this stage the flow-chart Γ is marked by the states [2] or by the microinstructions [4, 7].

2. Formation of the partition C of the set of objects. Each class of the set C corresponds to one LSO and to minimize the amount of chips in the circuit KC1 this partition C should satisfy to the condition

$$G \to \min. \qquad (2)$$

To solve this task we can apply the methods from the work [4].

3. Encoding of the objects. The goal of this stage is encoding of the objects which satisfies to the condition(1). To reach this goal we can use the following procedure[4]: formation of the vector $\alpha=\alpha_1*\alpha_2*...*\alpha_G$, where * is sign of concatenation, which includes all elements of the set O; determination of the number i of the vector α's component (i=0, 1,..., M-1);replacement of the decimal number i by its binary equivalent with $R=\mathrm{int}(\log_2 M)$ bits. These binary codes are the codes of the objects satisfying to the condition (1).

4. Formation of the table of the memory's excitation functions. Circuit KC1 forms functions $\varphi_r \in \Phi$, to get system Φ it is proposed to form the table of memory's excitation functions(TEF) for the objects which are the outputs of the LSO's. This table includes the following columns: o_m is the object of flow-chart which is an output of the LSO $\alpha_g \in C$; $K(o_m)$ is code of the object o_m; o_s is the object of flow-cart, in which there is transition from the object o_s; $K(o_s)$ is the code of the object o_s; x_h is input signal which determines the transition $<o_m, o_s>$ and which is conjunction of the some subset of input variables(or their negations), corresponding to the path from o_m into o_s in the flow-chart Γ; Φ_h is the set of excitation functions to switch the counter from code $K(o_m)$ into code $K(o_s)$; h =1,...,H is the number of transition. From this table we can obtain system

$$\varphi_r = \bigvee_{h=1}^{H} C_{rh} A_m^h X_h \qquad (r=\overline{1,R}), \qquad (3)$$

where C_{rh} is Boolean variable which is equal to one if and only if function $\varphi_r=1$ in the h-th line of the table, A_m^h is conjunction of the internal variables T, corresponding to the code $K(o_m)$ of the object o_m from the line number h of the TEF.

5. Formation of the table of the microoperations. Circuit KC2 forms the system of microoperations $Y \cup \{y_0\}$ and as the rule it is implemented on the ROM. Therefore table of microoperations(TMO) includes two columns: $K(o_m)$ is the code of the object o_m; $Y(o_m)$ is the set of microoperations corresponding to the object o_m. Signal y_0 is included in the all objects $o_m \in O$, except the outputs of the LSOs.

The problems of circuit's design on PLDs are well discussed in the literature, for example in [5], so it is not the subject of this article. Let us discuss the application of this approach to design the Moore automaton with counter on the some flow-chart ГСА Γ_1 which has 18 different states.

In this case the states from the set $A=\{a_1, ..., a_{18}\}$ are the objects, M=18, R=5. Using the approach from the article [4] we can get the set of the linear sequences of the states (LSS) $C=\{\alpha_1, ..., \alpha_6\}$, where $\alpha_1=<a_1,a_2,a_3>$, $\alpha_2=<a_4, a_5, a_6>$, $\alpha_3=<a_7, a_8, a_9, a_{10}>$, $\alpha_4=<a_{11}, a_{12}, a_{13}>$, $\alpha_5=<a_{14}, a_{15}, a_{16}>$, $\alpha_6=<a_{17}, a_{18}>$, G=6. The procedure of the states' encoding gives the next result: $K(a_1)$=00000, $K(a_2)$=00001, ..., $K(a_{18})$=10001, that is binary equivalent of the index m produces code $K(a_m)$ after subtraction of the one. The table of excitation functions is formed for the outputs of from the set $O(\Gamma_1)=\{\alpha_3, \alpha_6, \alpha_{10}, \alpha_{13}, \alpha_{16}, \alpha_{18}\}$ (Table 1). In our case this table has H=15 lines. The table of microoperations is formed elementary and we'll not discuss it.

Table 1. The table of Moore automaton's excitation functions

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Φ_h	h
a_3	00010	a_4	00011	x_1x_2	D_4D_5	1
		a_7	00110	$x_1\neg x_2$	D_3D_4	2
		a_9	01000	$\neg x_1x_3$	D_2	3
		a_{11}	01010	$\neg x_1\neg x_3$	D_2D_4	4
a_6	00101	a_{14}	01101	x_4	$D_2D_3D_5$	5
		a_{17}	10000	$\neg x_4$	D_1	6
a_{10}	01001	a_{14}	01101	x_4	$D_2D_3D_5$	7
		a_{17}	10000	$\neg x_4$	D_1	8
a_{13}	01100	a_{14}	01101	x_4	$D_2D_3D_5$	9
		a_{17}	10000	$\neg x_4$	D_1	10
a_{16}	01111	a_1	00000	1	-	11
a_{18}	10001	a_4	00011	x_1x_2	D_4D_5	12
		a_7	00110	$x_1\neg x_2$	D_3D_4	13
		a_9	01000	$\neg x_1x_3$	D_2	14
		a_{11}	01010	$\neg x_1\neg x_3$	D_2D_4	15

2. The method of distribution of code

As you can see from Table 1 the transitions for the states a_3 and a_{18} are equal ones because there are transitions into the same states with formation the same excitation functions of the D flip-flops of the counter. There is the same situations for the states a_6, a_{10} and a_{13}.

In the literature [8] such kind states have been named as pseudoequavalent states and the methods of the logical circuit's Moore automaton's optimization based on these states' existence have been developed. But we can not apply these methods in our particular case because the codes of the states should satisfy to the condition(1).

Let us represent the code of the object $K(o_m)$ as concatenation

$$K(o_m)=K(\alpha_g)*K_i \quad (m=\overline{1,\ M};\ i=\overline{1,\ F_g}), \qquad (3)$$

where $K(\alpha_g)$ is the code of LSO with R_1=int($\log_2 G$) bits, K_i is the code of the i-th component of the vector α_g with $R_2=\max(R_2^1,\ ...,\ R_2^G)$ bits, where $R_2^g=\mathrm{int}(\log_2 F_g)$ is the amount of bits to encode the components of the LSO $\alpha_g \in C$. Now condition (1) is transformed in the condition

$$K_{i+1}=K_i+1 \qquad (i=\overline{1,\ F_g-1}). \qquad (4)$$

From the analysis of the formulas (3)-(4) it is clear that we can choose the codes of LSO independently on the components' codes. An automaton based on the rules (3)-(4) includes counter CT to keep the component code and register RG to keep the LSO's code (Figure 2).

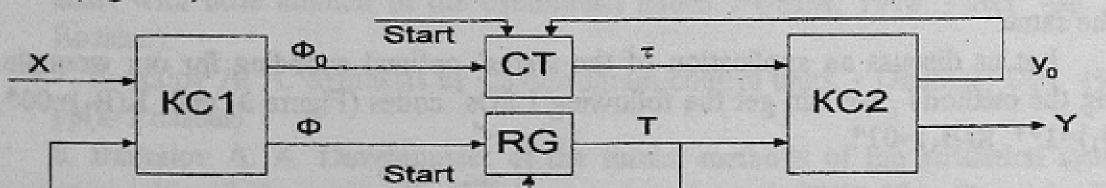

Figure 2. The structure of the automaton with distribution of the codes

Let us call an automaton based on rules (3)-(4) as automaton with distribution of the codes and denote it by the symbol U_2. An automaton U_2 operates in the following manner.

By the signal «Start» the zero code of the initial LSO is written in register RG and zero code of its first component is written in the counter CT. If pair $<T, \tau>$ determines the code of the object which is not the output of LSO, then circuit KC2 forms the set of microoperations and signal y_0 to increment the content of the counter by 1 accordingly with (4). Here $T=\{T_1, ..., T_{R1}\}$ is the set of internal variables to encode LSOs $\alpha_g \in C$, $\tau=\{\tau 1, ..., \tau_{R2}\}$ is set of internal variables to encode the components. If pair $<T, \tau>$ determines the code of the LSO's output, then signal y_0 is not formed. Circuit KC1 forms the set of the excitation functions $\Phi=\Phi(T, X)$ to write the code of the next object $\alpha_g \in C$ in register and functions $\Phi_0=\Phi_0(T, X)$ to write the code of component-input of LSO $\alpha_g \in C$ in the counter.

Such organization leads to the decreasing the complication of the circuit KC1 thanks the following factors:

1. If condition

$$R_1<R \qquad (5)$$

is satisfied, then the amount of the outputs of the circuit KC1 of the automaton U_2 is less, than similar parameter in the automaton U_1.

2. Independence between the codes of LSOs and components permits to decrease the amount of the terms to be implemented by the circuit KC1 (U_2) thanks the application of the known Moore automaton's optimization methods [8].

Let us introduce relation β on the set of the flow-chart's LSOs such that $\alpha_i\beta\alpha_j$ if the operational nodes corresponding to the outputs of LSOs α_i and α_j are connected with the input of the same node. Relation β determines partition $\pi_\beta=\{B_1, ..., B_I\}$ on the set C. To minimize the circuit KC1 we should unique express each class $B_i \in \pi_\beta$ using the following methods [8]:

- as generalized interval of the R_1-dimensional Boolean space (optimal encoding of objects);
- by interval of the R_3-dimensional Boolean space where R_3=int($\log_2 I$) (transformation of the object's codes in the codes of the classes of objects);
- by supplement objects(transformation of the initial flow-chart).

In the first and second cases TEF is replaced by the transformed TEF which can be created by the replacement of the column o_m by the column B_i, column $K(o_m)$ by the column $K(B_i)$ and elimination of K-1 from K equivalent lines.

In our example there is $\pi_\beta=\{B_1, B_2, B_3\}$,where $B_1=\{\alpha_1, \alpha_6\}$, $B_2=\{\alpha_2, \alpha_3, \alpha_4\}$, $B_3=\{\alpha_5\}$, R_1=3, R_2=2 (any LSS includes not more than four elements). In this particular case condition

$$R_1+R_2=R \qquad (6)$$

is satisfied, that is the hardware amount for circuits KC2 in the automata U_1 and U_2 will be the same.

Let us discuss an application of the states' optimal encoding for our example. Using the methods , we can get the following LSOs' codes (Figure 3) and $K(B_1)$=00*, $K(B_1)$=1**, $K(B_1)$=01*.

T_1 \ T_2T_3	00	01	11	10
0	α_1	α_6	α_5	*
1	α_2	α_3	α_4	*

Figure 3. Optimal encoding of the Moore automaton's states

The encoding of components accordingly with (4) is trivial process: the code of the i-th component is equal to the R_2-bit binary equivalent of the number i-1. The transformed TEF(Table 2) has H_1=7 lines, that is in our case circuit KC1 of the automaton U_2 has less inputs and implements less terms than circuit KC1 in the automaton U_1.

Table 2. Transformed TEF of the automaton U_2

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
B_1	00*	a_4	10000	x_1x_2	D_1	1
		a_7	10100	$x_1\neg x_2$	D_1D_3	2
		a_9	10110	$\neg x_1x_3$	$D_1D_3D_4$	3
		a_{11}	11100	$\neg x_1\neg x_3$	$D_1D_2D_3$	4
B_2	1**	a_{14}	01100	x_4	D_2D_3	5
		a_{17}	00100	$\neg x_4$	D_3	6
B_3	01*	a_1	00000	1	-	7

Conclusion

The investigations of the automata based on these principles have shown that if conditions (5)-(6) are satisfied then the amount of hardware in the U_2 is less than in the automata U_1. This gain is increasing when the number of operational nodes is increased and the number of LSOs is decreased. The gain for the control algorithms with 400-600 nodes is approximately 28%.

Literature

1. Stallings W. Computer Organization and Architecture. - N. J.: Prentice Hall, 1996 - 682 pp.
2. Baranov S.I. Synthesis of the microprogram automata. - L.: Energy, 1979. - 232 c.(in Russian).
3. Salislury A. Microprogrammable Computer Architectures. - N. Y.: Am. Elcerien, 1976 - 152 p.
4. Barkalov A. A. Microprogram control unit as composition of automata with microprogram and hardwired logic.//AVT,1983.-N4.-pp.36-41(in Russian).
5. Solovoyov V.V. Design of the functional units of digital circuits on the programmable logic devices. - Minsk: Bestprint, 1996. - 252 c.(in Russian).
6. Kirpichnicov V.M., Sclyarov V.A. Synthesis of microprogram automata on the flow-chart with little amount of the conditional nodes. //USIM, 1978. - №1. -pp. 77-82(in Russian).
7. Barkalov A. A. Synthesis of microprogram control units. - Donetsk: DPI, 1992. - 48 pp(in Russian).
8. Barkalov A. A. Development of the formal methods of the structural synthesis of compositional automata. Thesis...doct. techn.sciences: 05.13.08 - Donetsk: DonSTU, 1994. - 301 pp(in Russian).