

УДК 004.274

А.А. Баркалов<sup>1</sup>, д-р техн. наук, проф.,Л.А. Титаренко<sup>1</sup>, д-р техн. наук, проф.,И.Я. Зеленева<sup>2</sup>, канд. техн. наук, доц.<sup>1</sup>Університет Зеленогурський, г. Зелена Гура, Польща<sup>2</sup>Запорозький національний технічний університет, г. Запорозжє, Україна

E-mail: A.Barkalov@ie.uz.zgora.pl

## Реализация совмещенного микропрограммного автомата в базисе FPGA

Предложен метод реализации логической схемы совмещенного микропрограммного автомата в базисе FPGA. Схема автомата реализуется на встроенных блоках памяти и элементах табличного типа. Определены условия целесообразности применения предложенного метода. Приведен пример синтеза совмещенного микропрограммного автомата по исходной граф-схеме алгоритма.

**Ключевые слова:** совмещенный микропрограммный автомат, FPGA, LUT, встроенные блоки памяти, граф-схема алгоритма, синтез логической схемы.

### Введение

Устройство управления является важной частью практически любой цифровой системы [1]. При реализации схемы устройства управления (УУ) необходимо оптимизировать его характеристики. К важнейшим характеристикам относится площадь кристалла, занимаемая схемой УУ. Как показано в работах [2, 3], уменьшение площади ведет к уменьшению таких показателей, как время распространения сигнала и потребляемая энергия. Методы уменьшения площади в значительной степени зависят от модели УУ и элементного базиса [4].

В настоящей работе рассматривается метод решения этой задачи при реализации схемы совмещенного микропрограммного автомата (СМПА) в базисе FPGA (field-programmable-gate-array). Выбор модели СМПА объясняется практически полным отсутствием статей, связанных с синтезом схем СМПА. Как правило, в литературе рассматриваются либо автоматы Мили, либо автоматы Мура. Однако СМПА часто используется в практике проектирования цифровых систем [5]. Выбор базиса FPGA обусловлен его высокой популярностью. При этом считается, что FPGA будут широко использоваться в ближайшие десять лет [6].

### Особенности модели совмещенного автомата и базиса FPGA

Совмещенный автомат может быть представлен в виде восьмикомпонентного вектора:

$$S = \langle A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, \alpha_1 \rangle \quad (1)$$

Вектор (1) складывается из следующих компонентов:

$A = \{\alpha_1, \dots, \alpha_m\}$  - множество внутренних состояний;

$X = \{x_1, \dots, x_l\}$  - множество логических условий;

$Y^1$  - множество выходных переменных автомата Мили;

$Y^2$  - множество выходных переменных автомата Мура;

$\delta$  - функция переходов;

$\lambda_1$  - функция выходов автомата Мили;

$\lambda_2$  - функция выходов автомата Мура;

$\alpha_1 \in A$  - начальное состояние.

Функция  $\delta$  - служит для нахождения состояния перехода  $\alpha_s \in A$  в зависимости от текущего состояния  $\alpha_m \in A$  и вектора входных переменных:

$$\alpha_s = \delta(\alpha_m, X) \quad (2)$$

Множество выходных переменных  $Y = Y^1 \cup Y^2$  состоит из  $N_1 = |Y^1|$  переменных автомата Мили и  $N_2 = |Y^2|$  переменных автомата Мура. При этом  $Y^1 \cap Y^2 = \emptyset$  и  $N_1 + N_2 = N$ .

Функция  $\lambda_1$  определяет выходные переменные  $y_n \in Y^1$ :

$$y_n = \lambda_1(\alpha_m, X) \quad (3)$$

Функция  $\lambda_2$  определяет выходные переменные  $y_n \in Y^2$ :

$$y_n = \lambda_2(\alpha_m) \quad (4)$$

В настоящей работе СМПА представляется граф-схемой алгоритма (ГСА) [1]. Для нахождения функций (2) - (4) необходимо построить по ГСА прямую структурную таблицу (ПСТ). Эта таблица рассматривается более детально в нашей работе.

Для реализации схемы СМПА можно использовать три компонента FPGA: элементы табличного типа LUT (look up table), встроенные блоки памяти ЕВМ (embedded memory block) и программируемые межсоединения [7,8]. Элементы LUT представляют собой ОЗУ, имеющее S входов и один выход. При этом число входов определяет максимальное число аргументов булевой функции, которая может быть реализована на одном LUT элементе. Как правило, число входов  $S \leq 6$  [7,8].

Блоки ЕВМ представляют собой ОЗУ, имеющее  $S_A$  входов и  $t_F$  выходов. Общая емкость блоков является константой и определяется следующей формулой:

$$V_0 = 2^{S_A * t_F} \quad (5)$$

Параметры  $S_A$  и  $t_F$  могут быть выбраны из следующего множества пар вида  $\langle S_A, t_F \rangle$ :  $\langle 15, 1 \rangle$ ,  $\langle 14, 2 \rangle$ ,  $\langle 12, 8 \rangle$ ,  $\langle 11, 16 \rangle$ ,  $\langle 10, 32 \rangle$  и  $\langle 9, 64 \rangle$ .

### Тривиальная реализация совмещенного автомата в базе FPGA

Для реализации схемы совмещенного микропрограммного автомата по ГСА Г необходимо выполнить некоторые промежуточные этапы:

1. Отметить ГСА состояниями СМПА
2. Закодировать состояния  $a_m \in A$  двоичными кодами  $K(a_m)$  разрядности  $R$ .
3. Построить ПСТ автомата.
4. Получить системы функций, соответствующие (2)-(4).

Для кодирования состояний  $a_m \in A$  используются внутренние переменные  $T_r \in T$ , где  $|T|=R$ . Закодируем состояния минимальным числом внутренних переменных:

$$R = \lceil \log_2 M \rceil \quad (6)$$

Для хранения кодов состояний используется регистр (RG), включающий  $R$  триггеров. Как правило, при синтезе в базе FPGA используются D триггеры [5]. Для изменения содержимого RG необходимо задать функции возбуждения  $D_r \in \Phi$ , где  $\Phi = \{D_1, \dots, D_r\}$ . Изменение содержимого RG происходит по сигналу синхронизации Clock. Для установки в RG кода состояния  $a_1 \in A$  используется импульс Start.

После выполнения этапов 1-2, ПСТ формируется с использованием правил [1]. ПСТ является основой для получения систем:

$$\Phi = \Phi(T, X) \quad (7)$$

$$Y^1 = Y^1(T, X) \quad (8)$$

$$Y^2 = Y^2(T) \quad (9)$$

Система (7) определяет функцию (2), система (8) – функцию (3) и система (9) – функцию (4).

Назовем термином LUTer схему, состоящую из элементов LUT. Схему из блоков ЕВМ будем называть ЕМBer. Существует две тривиальные модели СМПА для базиса FPGA, показанные на рис.1.

Для модели  $U_1$  (рис.1,а) характерно использование только элементов LUT для реализации систем (7) – (9). В случае модели  $U_2$  (рис.1,б) системы (7) – (9) реализуются на блоках ЕВМ. В обоих случаях регистр RG в явном виде отсутствует. В модели  $U_1$  триггеры распределены между логическими элементами (ЛЭ), в которые входят элементы LUT. В модели

$U_2$  используется возможность синхронизации блоков ЕВМ [5,7,8].

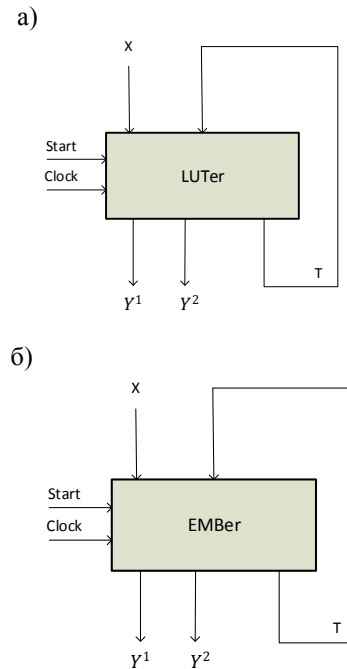


Рисунок 1 - Тривиальные модели совмещенного автомата в базе FPGA:  $U_1$ (а) и  $U_2$ (б).

Недостатком моделей  $U_1$  и  $U_2$  является избыточность реализации для практических примеров. Это значит, что для тривиальной (одноуровневой) реализации схем автоматов реальной сложности [1] требуется больше логических элементов, чем для многоуровневых структур [5].

В настоящей работе мы предлагаем использовать принцип гетерогенной реализации [5] для уменьшения числа элементов в схеме СМПА. Очевидно, что уменьшение числа элементов ведет к уменьшению площади кристалла FPGA, занимаемой схемой СМПА.

### Основная идея предлагаемого метода и пример его применения

Анализ систем (7) – (9) показывает, что функции  $\Phi$  и  $Y^1$  зависят от одних и тех же переменных. В системе  $Y^2$  отсутствует зависимость переменных  $x_e \in X$ . Представим СМПА в виде композиции двух комбинационных схем и регистра RG (рис.2).

Схема КС1 реализует системы (7) – (8), а схема КС2 – систему (9). Принцип гетерогенной реализации заключается в том, что разные части схемы реализуются на разных логических элементах [9]. В случае использования FPGA такими элементами являются внутренние элементы LUT и блоки ЕВМ. Выбор элементов зависит от сочетания характеристик автомата и элементного базиса.

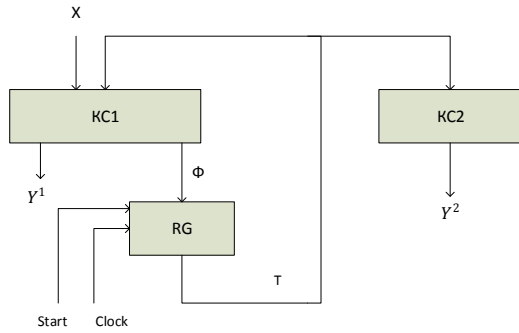


Рисунок 2 - Структурная схема совмещенного автомата

Пусть следующие условия выполняются для некоторого СМПА:

$$2^{L+R} * (N_1 + R) \leq V_0; \quad (10)$$

$$2^{L+R} * (N + R) > V_0; \quad (11)$$

$$R \leq S. \quad (12)$$

Условие (10) свидетельствует о том, что системы (7) – (8) могут быть реализованы на одном блоке ЕМВ. Условие (11) показывает, что для реализации систем (7) – (9) требуется больше одного блока ЕМВ. Условие (12) свидетельствует о том, что каждая из функций  $y_n \in Y^2$  реализуется на одном элементе LUT.

Таким образом, целесообразно использовать ЕМВ для реализации схемы КС1 и LUT для КС2. Эта идея представлена структурной схемой  $U_3$  (рис.3). Для реализации RG используются элементы LUT, связанные с D-триггерами.

В настоящей работе предлагается метод синтеза автомата  $U_3$  по исходной ГСА. Этот метод включает следующие этапы:

1. Формирование множества состояний А.
2. Кодирование состояний  $a_m \in A$ .
3. Формирование ПСТ совмещенного автомата.
4. Формирование таблицы блока ЕМВ.
5. Формирование таблиц элементов блока LUTer.
6. Реализация схемы автомата.

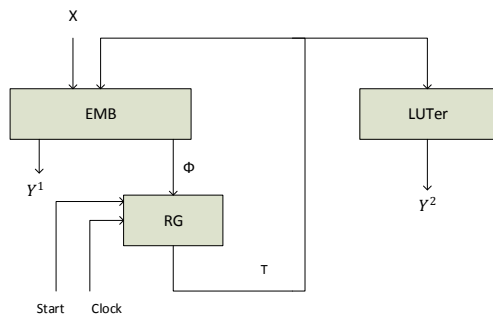


Рисунок 3 - Структурная схема СМПА  $U_3$

Рассмотрим применение этого метода на примере ГСА  $\Gamma_1$  (рис.4). На дугах ГСА  $\Gamma_1$  показаны переменные  $y_n \in Y^1$ , а в операторных вершинах – переменные  $y_n \in Y^2$ .

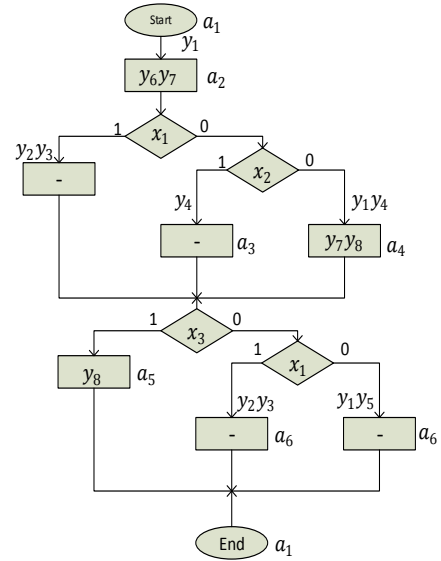


Рисунок 4 - Исходная ГСА  $\Gamma_1$ , отмеченная состояниями совмещенного автомата

Поясним принцип отметки состояний. Из-за наличия системы (9) СМПА должен быть отмечен, как автомат Мура [1]. Однако отметки могут совпадать для вершин, у которых:

- а) нет переменных  $y_n \in Y^1$ , а также
- б) выходы которых связаны со входом одной и той же вершины ГСА.

Например, отметка  $a_3$  встречается два раза, как и отметка  $a_6$  (рис.4).

Следующие множества и их параметры могут быть получены для ГСА  $\Gamma_1$ :

$A = \{a_1, \dots, a_6\}$ ,  $M=6$ ,  $X = \{x_1, x_2, x_3\}$ ,  $L=3$ ,  $Y^1 = \{y_1, \dots, y_5\}$ ,  $N_1=5$ ,  $Y^2 = \{y_6, y_7, y_8\}$ ,  $N_2=3$  и  $N=8$ . Так как  $M=6$ , то формула (6) определяет число разрядов кода состояния  $R=3$ . Это определяет множество кодирующих переменных и функций возбуждения памяти триггеров:  $T = \{T_1, T_2, T_3\}$  и  $\Phi = \{D_1, D_2, D_3\}$  соответственно.

Пусть схема СМПА будет реализована в базе FPGA, имеющем блоки ЕМВ с конфигурацией  $64*8$  и элементы LUT с  $S=3$ . Проверка условий (10) – (12) показывает, что они выполняются. Следовательно, модель  $U_3$  может быть использована в рассматриваемом примере.

Так как система (7) реализуется на ЕМВ, то она представляется таблицей истинности. Следовательно, коды состояний не влияют на аппаратные затраты. Закодируем состояния  $a_m \in A$  тривиальным образом:  $K(a_1)=000, \dots, K(a_6)=101$ .

Прямая структурная таблица СМПА строится по правилам [1] и включает следующие столбцы:

- $a_m$  - текущее состояние;
- $K(a_m)$  - код состояния  $a_m \in A$ ;
- $a_s$  – состояние перехода;
- $K(a_s)$  - код состояния  $a_s \in A$ ;

$X_h$  - входной сигнал, определяющий переход  $(a_m, a_s)$ ;

$Y_h^1$  - набор входных переменных, формируемых при переходе  $(a_m, a_s)$ ;

$\Phi_h$  - набор функций возбуждения памяти, принимающих единичные значения для переключения памяти из  $K(a_m)$  в  $K(a_s)$ ;

$h$  - номер перехода.

Кроме того, в столбце  $a_m$  записываются переменные  $y_n \in Y^2$ , формируемые в состоянии  $a_m \in A$ .

Для рассматриваемого примера ПСТ имеет  $N=10$  строк (табл.1). При этом  $(a_5, a_1)$  и  $(a_6, a_1)$  не включены в ПСТ, так как они выполняются по сигналу Clock. При этом надо помнить, что в состоянии  $a_5$  формируется выходной сигнал  $y_8$ .

Таблица 1

Прямая структурная таблица СМПА

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X_h$	$Y_h^1$	$\Phi_h$	$h$
$a_1$	000	$a_2$	001	1	$y_1$	$D_3$	1
$a_2$ $y_6 y_7$	001	$a_3$	010	$x_1$	$y_2 y_3$	$D_2$	2
		$a_3$	010	$\bar{x}_1 x_2$	-	$D_2$	3
		$a_4$	011	$\bar{x}_1 \bar{x}_2$	$y_1 y_4$	$D_2 D_3$	4
$a_3$	010	$a_5$	100	$x_3$	$y_3 y_5$	$D_1$	5
		$a_6$	101	$\bar{x}_3 x_1$	$y_2 y_3$	$D_1 D_3$	6
		$a_6$	101	$\bar{x}_3 \bar{x}_1$	$y_1 y_5$	$D_1 D_3$	7
$a_4$ $y_7 y_8$	011	$a_5$	100	$x_3$	$y_3 y_5$	$D_1$	8
		$a_6$	101	$\bar{x}_3 x_1$	$y_2 y_3$	$D_1 D_3$	9
		$a_6$	101	$\bar{x}_3 \bar{x}_1$	$y_1 y_5$	$D_1 D_3$	10

Прямая структурная таблица используется на следующем шаге синтеза для построения таблицы ЕМВ. Таблица ЕМВ имеет столбцы  $K(a_m)$  и  $X$ , определяющие адрес ячейки памяти, и столбцы  $Y^1$  и  $\Phi$ , определяющие её содержимое. Переходы из каждого состояния  $a_m \in A$  задаются в таблице ЕМВ (табл.2) одинаковым числом строк  $N(a_m)$ :

$$N(a_m) = 2^L \quad (13)$$

В рассматриваемом примере  $N(a_m) = 8$

Формат таблицы ЕМВ для рассматриваемого примера представлен в табл.2. Он задает часть таблицы, соответствующей состоянию  $a_2 \in A$ .

В табл.2 столбец  $q$  показывает номер ячейки, а столбец  $h$  показывает соответствующий номер строки ПСТ. Как видно из табл.2, строки 13-16 соответствуют  $x_1$ , строки 11,12 -  $\bar{x}_1 x_2$  и 9,10 -  $\bar{x}_1 \bar{x}_2$ .

Так как условие (12) выполняется, то LUTer реализуется на трех элементах LUT. Таблицы этих элементов строятся тривиальным образом. Каждая таблица имеет входы  $K(a_m)$  и

выход  $y_n \in Y^2$ . Единицы в ячейках таблицы соответствуют кодам состояний, для которых формируется данная переменная. Таблица для функции  $y_6$  представлена в табл.3.

Таблица 2

Фрагмент таблицы ЕМВ

$K(a_m)$	$X$	$Y^1$	$\Phi$	$q$	$h$
$T_1 T_2 T_3$	$x_1 x_2 x_3$	$y_1 y_2 y_3 y_4 y_5$	$D_1 D_2 D_3$		
001	000	10010	011	9	4
001	001	10010	011	10	4
001	010	00000	010	11	3
001	011	00000	010	12	3
001	100	01100	010	13	2
001	101	01100	010	14	2
001	110	01100	010	15	2
001	111	01100	010	16	2

Таблица 3

Таблица LUT элемента для функции  $y_6$

$T_1$	0	0	0	0	1	1	1	1
$T_2$	0	0	1	1	0	0	1	1
$T_3$	0	1	0	1	0	1	0	1
$y_6$	0	1	0	0	0	0	0	0
$q$	1	2	3	4	5	6	7	8

Последний этап предложенного метода связан с записью таблиц истинности в соответствующие элементы LUT и программированием межсоединений. Этот этап выполняется с использованием стандартных промышленных пакетов [5].

### Заключение

В настоящей работе предлагается метод синтеза совмещенного микропрограммного автомата на FPGA. Схема СМПА реализуется на элементах табличного типа и встроенных блоках памяти. Число элементов в схеме зависит от сочетания параметров СМПА и элементного базиса. В работе определены условия применения данного метода.

Если условие (10) нарушается, то для реализации систем (7) - (8) требуется более одного блока ЕМВ. Анализ библиотеки [10] показал, то это условие (10) выполняется для 81% стандартных автоматов. Если условие (12) нарушается, то усложняется схема блока LUTer.

Направлением дальнейших исследований является разработка методов синтеза СМПА при нарушении условий (10) и/или (12). Для этого могут быть использованы различные методы [5], адаптированные к условиям совмещенного автомата.

**Список использованной литературы**

1. Baranov S. Logic and System Desing of Digital Systems. – Tallinn: TUT Press, 2008. – 267 p.
2. De Micheli G. Synthesis and Optimization of Digital Circuits. – New York: Mc Graw-Hill, 1994. – 636 p.
3. Соловьев В.В. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев, А. Климович. – М.: Горячая Линия – Телеком, 2008. – 376 с.
4. Czerwinski R., Kania D. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. – Berlin: Springer, 2013. – 172 p.
5. Sklyarov V., Sklyarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA – based Systems. – Berlin: Springer, 2014. – 432 p.
6. Sklyarova I., Sklyarov V., Sudnitson A. Design of FPGA – based circuits using Hierarchical Finite State Machines. – Tallinn: TUT Press, 2012. – 240 p.
7. www.altera.com
8. www.xilinx.com
9. Баркалов А.А. Принципы оптимизации логической схемы микروпрограммного автомата Мура / А.А. Баркалов // Кибернетика и системный анализ. – 1998. – №1. – С. 65 – 72.
10. Yang S. Logic Synthesis and optimization benchmarks user guide. – North Carolina: Microelectronics Center of North Carolina, 1991. – 43 p.

Надійшла до редакції 25.09.2015

**О.О. БАРКАЛОВ<sup>1</sup>, Л.О. ТИТАРЕНКО<sup>1</sup>, І.Я. ЗЕЛЕНЬОВА<sup>2</sup>**

<sup>1</sup>Університет Зеленогурський (Польща)

<sup>2</sup>Запорізький національний технічний університет (Україна)

**РЕАЛІЗАЦІЯ СУМІЩЕНОГО МІКРОПРОГРАМНОГО АВТОМАТА В БАЗИСІ FPGA**

Запропоновано метод реалізації логічної схеми суміщеного мікропрограмного автомата в базисі FPGA. Схема автомата реалізується на вбудованих блоках пам'яті і елементах табличного типу. Визначено умови застосування запропонованого методу. Наведено приклад синтезу автомата по вихідній граф-схемі алгоритму.

**Ключові слова:** суміщений мікропрограмний автомат, FPGA, LUT, вбудовані блоки пам'яті, граф-схема алгоритму, синтез логічної схеми.

**A.A. BARKALOV<sup>1</sup>, L.A. TITARENKO<sup>1</sup>, I.J. ZELENJOVA<sup>2</sup>**

<sup>1</sup>University of Zielona Góra (Poland)

<sup>2</sup>Zaporizhzhya National Technical University (Ukraine)

**IMPLEMENTING OF COMBINED FINITE STATE MACHINE WITH FPGA**

A method for implementing logic circuit of combined finite state machine (CFSM) on base FPGA is proposed. The control unit is an important part of almost any digital system. In a process of implementing the circuit of control unit it is necessary to optimize its characteristics. The most important characteristic is the chip area occupied by the circuit of control unit. Reduction of the chip area leads to decreasing of characteristics such as the signal propagation time and power consumption. Methods of reducing the required chip area significantly depend on the model of control unit and the element basis. Choosing of the model CFSM for consideration in this article is explained by the almost complete absence of publications related to the synthesis schemes of combined finite state machine. As a rule, in the literature either Mealy or Moore machines are considered. However CFSM is often used in practice in the design of digital systems. Choice of element basis FPGA for the implementation of logic circuit CFSM due to high popularity of FPGA nowadays. It is assumed that the FPGA will be widely used in the next ten years. For the implementation of the scheme combined finite state machine it's possible to use the three internal components of FPGA: the elements of a table-type LUT (look up table), embedded memory blocks EBM (embedded memory block) and programmable interconnects. The number of elements required for the scheme depends on a combination of characteristics CFSM and parameters of elemental basis. Clearly, reducing the number of elements leads to reducing the area of the chip FPGA, occupied by the circuit CFSM. In this article we propose to use the principle of a heterogeneous implementation to reduce the number of elements in the scheme CFSM. The principle of a heterogeneous implementation means that different parts of the circuit are implemented on the different logical cells. In the case of using FPGA such elements are the internal elements and blocks, such as LUT and EMB. Selecting of elements depends on the combination of the characteristics of the machine and the elemental basis. The article also defines the conditions of the appropriateness of the proposed method - namely, the permissible ratio of the parameters of an initial graph-scheme of algorithm and parameters of the built-in blocks FPGA. An example of the synthesis of the combined finite state machine on the original graph-scheme of algorithm using the FPGA is considered.

**Keyu words:** combined finite state machine, FPGA, LUT, embedded memory blocks, graph-scheme of algorithm, logical circuit synthesis.