РЕАЛИЗАЦИЯ СХЕМ АВТОМАТОВ МИЛИ НА СЧЁТЧИКАХ

Баркалов А. А., Титаренко Л. А., Ковалев С. А.

Предложен метод уменьшения числа элементов табличного типа (LUT) в схеме автомата Мили. В работе рассматривается метод кодирования стабильных пар логических условий. Этот подход позволяет уменьшить число аргументов в функциях возбуждения памяти. При этом уменьшается число входов блока замены логических условий дополнительными переменными. Рассмотрен пример синтеза и приведены результаты исследований.

Ключевые слова: микропрограммный автомат Мили, синтез, FPGA, LUT, пары логических условий, счётчик состояний.

Введение. Одной из наиболее важных проблем, связанной реализацией цифровых схем, является уменьшение аппаратных затрат [1]. Для решения этой проблемы необходимо учитывать особенности цифровой схемы и элементного базиса [2]. Одно из наиболее трудных проблем в этой области является оптимизация схем устройств управления. Это связано с их нерегулярной структурой [2]. В этой статье предлагается метод оптимизации схемы микропрограммного автомата (МПА) Мили, реализующего некоторый алгоритм управления. Мы сосредотачиваем усилия на уменьшении площади кристалла, занимаемой схемой МПА. Как известно, решение этой задачи позволяет уменьшить время задержки сигналов и потребляемую мощность [2-4]. Как правило, в схемах МПА Мили используются регистры, хранящие коды состояний [2]. В этой статье мы рассматриваем случай, когда регистр заменен счетчиком [5]. В качестве элементного базиса мы выбрали СБИС типа FPGA (field-programmable gate arrays) [6-7]. Этот базис является основным при реализации сложных цифровых систем [3]. При этом схема реализуется на элементах табличного типа, называемых LUT (look-up table) [6-7]. Для реализации схем счётчиков используются триггера, связанные через мультиплексоры с выходами элементов LUT.

Основной особенностью элементов LUT является ограниченное число входов (S_L). Например, для семейства Virtex-5 $S_L=5$, для Virtex-7 $S_L=6$ [7]. Считается что $S_L=6$ является оптимальным [8] с точки зрения площади, занимаемой элементом, и потребляемой им энергией.

Основной особенностью МПА Мили является зависимость его выходных сигналов (микроопераций) от входных сигналов (логических условий) и состояний [2]. Системы булевых функций (СБФ), представляющие микрооперации (M_0), могут зависеть от десятков аргументов [2]. Этот дисбаланс приводит к многоуровневым схемам МПА [9-10]. Для таких схем характерно значительное число межсоединений и, как следствие, значительная паразитная емкость [2]. Как известно, чем больше паразитная емкость, тем больше задержка сигнала и потребляемая мощность [11]. Для уменьшения числа аргументов в СБФ, мы предлагаем заменить логические условия (ЛУ) некоторыми дополнительными переменными. Такой подход является развитием идей из работы [12].

Синтез автоматов на счётчиках. Поведение автоматов может быть задано при помощи таблиц либо графов [2-3]. В данной работе мы используем язык граф-схем алгоритмов (ГСА) [2], обладающий значительной наглядностью для иллюстрации методов синтеза. Рассмотрим в качестве примера ГСА Γ_1 (рис.1).

Для отметки состояний на ГСА Γ_1 использован метод из работы [2]. Анализ ГСА Γ_1 позволяет найти множество состояний $A=\{a_1, \dots, a_8\}$, логических условий $X=\{x_1, \dots, x_4\}$ и микроопераций $Y=\{y_1, \dots, y_6\}$. Пусть $|A|=M$, $|X|=L$ и $|Y|=N$. В рассматриваемом примере $M=8$, $L=4$ и $N=6$. Кроме того, пути вида $\langle a_m, x_h, y_h, a_s \rangle$ определяют переходы между состояниями $a_m, a_s \in A \{ \}$. При этом a_m – исходное (текущее) состояние; a_s – состояние перехода; x_h – входной сигнал, определяющий переход $\langle a_m,$

a_s и равный конъюнкции некоторых ЛУ $x_i \in X$ (или их отрицаний); y_h – выходной сигнал, формируемый на переходе номер h ($h=1, N$).

Для автоматов практического объема [2], $L \approx 50$, $N \approx 100$, $M \approx 200$, $H \approx 2000$. Этот факт определяет значительную сложность СБФ, задающих схему МПА.

Рис.1. Отмеченная ГСА Γ_1

Для реализации схемы МПА необходимо закодировать состояние $a_m \in A$ двоичными кодами $K(a_m)$ [2]. Для кодирования состояний используются элементы множества внутренних переменных $T = \{T_1, \dots, T_R\}$. Коды $K(a_m)$ хранятся в специальном регистре состояний (RGS), состоящим из R двухтактных триггеров типа D [9]. Для смены содержимого RGS используются функции возбуждения памяти (ФВП), образующие множество $\Phi = \{D_1, \dots, D_R\}$.

Схема МПА задается СБФ

$$\Phi = \Phi(T, X); \quad (1)$$

$$Y = Y(T, X); \quad (2)$$

Каждая функция $D_r \in \Phi$ и $y_n \in Y$ реализуется как совокупность элементов LUT, называемой нами LUTer. Регистр RGS распределен между LUTer $D_1, \dots, D_R$. Для обнуления RGS используется импульс Start. Импульс Clock разрешает смену состояния (содержимого RGS). Всё это

определяет структурную схему МПА Мили U_1 (рис.2).

Рис.2. Структурная схема МПА U_1

Символ LUTerΦ соответствует блоку, включающему подсхемы LUTer $D_1, \dots, D_R$ и распределенный регистр RGS. Символ LUTerY соответствует блоку, состоящему из подсхем LUTer $Y_1, \dots, Y_n$. При этом LUTer Y_n реализует схему для МО $y_n \in Y$.

Для уменьшения числа термов в СБФ (1), регистр RGS можно заменить счетчиком состояний (CTS). При этом некоторые переходы осуществляются путем увеличения содержимого счётчика. Как правило, этот подход используется для автоматов Мура [5]. Однако, в [12] предлагалось заменить RGS на счётчик для МПА Мили. Это ведет к МПА Мили U_2 (рис.3).

Рис.3. Структурная схема МПА U_2

В автомате U_{20} , блок БФВП формирует СБФ (1) и сигнал $i_1 \# CT := CT + 1$. Блок БМО формирует СБФ (2) и сигнал $i_2 \# CT := CT + 1$. Блок БФВП формирует i_1 для безусловных переходов, а БМО - для безусловных [12].

Однако в случае FPGA, уменьшение числа термов не влияет на число LUT элементов в схеме МПА [3,9]. Для решения этой задачи необходимо уменьшать число аргументов в реализуемых СБФ. В настоящей работе мы предлагаем один из возможных подходов к решению этой задачи.

Основная идея предложенного метода. Для реализации функции СБФ f_i , имеющей $NL(f_i)$ литералов, достаточно одного элемента LUT, если следующие условия имеют место:

$$NL(f_1) \leq S_L. \quad (3)$$

При нарушении условия (3) для реализации схемы требуется несколько элементов. В этом случае схема имеет не менее двух уровней.

Если (3) нарушается, то мы предлагаем два метода, одновременное применение которых приводит к оптимизации схемы МПА Мили на счётчике. Это следующие методы:

1. Замена ЛУ $x_1 \in X$ дополнительными переменными $P_g \in P = \{P_1, \dots, P_G\}$, где $G \ll L$.

2. Кодирование стабильных пар ЛУ, определяющих переходы $\langle a_m, a_s \rangle$, элементами множества $Z = \{z_1, \dots, z_{RL}\}$. Пусть N_P - число стабильных пар ЛУ. Тогда для их кодирования достаточно RL переменных:

$$R_L = \lceil \log_2 N_P \rceil. \quad (4)$$

Пусть функции $z_r \in Z$ формируются блоком БМО. Тогда МПА Мили на CTS может быть представлен в виде U_3 (рис.4).

Рис.4. Структурная схема МПА Мили U_3

В схеме на рис.4 блок LUTerP реализует СБФ

$$P = P(Z, X). \quad (5)$$

Функции (5) используются как аргументы функции

$$\Phi = \Phi(T, P); \quad (6)$$

$$i_1 = i_1(T, P). \quad (7)$$

Функции (6)-(7) реализуется блоком LUTerPhi, соответствующем БФВП (рис.3). Блок LUTerY формирует СБФ

$$Y = Y(T, P); \quad (8)$$

$$Z = Z(T, P); \quad (9)$$

$$i_2 = i_2(T, P). \quad (10)$$

Блок LUTerY соответствует блоку БМО (рис.3).

Переменные $T_r \in T$ кодируют состояния $a_m \in A$. Как правило, для МПА, реализуемых на элементах LUT, число R определяется как

$$R = \lceil \log_2 M \rceil. \quad (11)$$

Мы будем использовать (11) в нашем методе.

В настоящей статье предлагается метод синтеза МПА U_3 , который включает следующие этапы:

1. Отметка ГСА Γ состояниями автомата Мили.
2. Создание множества линейных цепей состояний.
3. Кодирование состояний МПА.
4. Формирование множества стабильных пар логических условий.
5. Кодирование пар ЛУ и замена ЛУ.
6. Формирование таблиц переходов МПА U_3 .
7. Формирование СБФ (5) – (10).
8. Реализация схемы МПА в заданном базисе.

Пусть символ $U_i(\Gamma_j)$ означает, что модель U_i используется для синтеза схемы, начиная с ГСА Γ_j . Рассмотрим пример синтеза МПА $U_3(\Gamma_1)$.

Примеры синтеза МПА U_3 . Центральное место в предлагаемом методе занимает второй этап. Определим линейную цепь состояний (ЛЦС) как вектор d_i ($i = \overline{1, I_C}$), элементами которого являются состояния. При этом для любой пары соседних компонент существует условный или безусловный переход из компонента с меньшим индексом в компонент с большим на единицу индексом. В результате выполнения этого этапа, формируется разбиение Π_C множество состояний A на I_C цепей. Для оптимизации схемы, необходимо найти разбиение $\Pi_C = \{d_1, \dots, d_{I_C}\}$ с минимально возможным числом ЛПС [5].

Для автомата $U_3(\Gamma_1)$ множество A включает $M=8$ состояний. Используя метод [5], найдём $\Pi_C = \{d_1, d_2\}$, где $d_1 = \langle a_1, a_2, a_3, a_4, a_6 \rangle$, $d_2 = \langle a_5, a_7, a_8 \rangle$.

Состояния $a_m \in A$ необходимо закодировать так, чтобы для любой пары компонент выполнялось условие

$$K(a_i) = K(a_{i-1}) + 1. \quad (12)$$

Назовём такое кодирование натуральным.

Для нашего примера из (11) следует, что $R=3$, $T = \{T_1, T_2, T_3\}$ и $\Phi = \{D_1, D_2, D_3\}$. Закодируем состояние так, как показано на рис.5.

$T_1 T_2 T_3$	000	001	010	011	100	101	110	111
	a_1	a_2	a_3	a_4	a_6	a_5	a_7	a_8

Рис.5. Кодирование состояний автомата

В рассматриваемом примере существует две пары логических условий: $B_1 = \langle x_1, x_2 \rangle$ и $B_2 = \langle x_3, x_4 \rangle$. Это следует из анализа ГСА Γ_1 . Итак, $N_p=2$ и из (4) следует, что $R_L = 1$, $Z = \{z_L\}$. Закодируем пары произвольным образом: $K(B_1) = 0$ и $K(B_2) = 1$.

Таблица замены ЛУ строится тривиальным образом. Её столбцы отмечены кодами $K(B_n)$, а строки переменными $p_g \in P$. Для рассматриваемого примера $G=2$, $P=\{p_1, p_2\}$. Один из вариантов замены показан в табл.1.

Из табл.1 можно найти СБФ (5). Для нашего примера имеем:

$$\begin{cases} p_1 = \bar{z}_1 x_1 \vee z_1 x_3; \\ p_2 = \bar{z}_1 x_2 \vee z_1 x_4. \end{cases} \quad (13)$$

Система (13) определяет схему блока LUTerP.

Таблица 1. Таблица замены ЛУ для $U_3(\Gamma_1)$

$\begin{matrix} Z \\ P \end{matrix}$	0	1
P_1	x_1	x_3
P_2	x_2	x_4

Таблица переходов МПА U_3 содержит следующие столбцы: a_m , $K(a_m)$, a_s , $K(a_s)$, P_h , Y_h , Φ_h , Z_h , h . При этом в столбце Y_h записываются не только МО $y_n \in Y$ формируемые при переходе из a_m в a_s , но и переменные i_1 (для безусловных переходов) и i_2 (для безусловных переходов). Столбец P_1 содержит конъюнкцию дополнительных переменных $P_g \in P$ (или их отрицаний), определяющую переход номер h ($h = \overline{1}, N$). Столбец Φ_h содержит ФВП, принимающее значение 1 для переходов, которые не удовлетворяют условию (12).

Для нашего примера $N=12$. Таблица переходов МПА $U_3(\Gamma_1)$ приведена в табл.2.

Таблица 2. Таблица переходов МПА $U_3(\Gamma_1)$

a_m	$k(a_m)$	a_s	$k(a_s)$	P_h	Y_h	Φ_h	Z_h	h
a_1	000	a_2	001	1	$y_1 y_2 i_1$	-	-	1
a_2	001	a_3	010	1	$y_3 i_1$	-	-	2
a_3	010	a_4	011	p_1	$y_4 i_2$	-	-	3
		a_4	011	$\bar{p}_1 p_2$	$y_2 y_5 i_2$	-	-	4
		a_5	101	$\bar{p}_1 \bar{p}_2$	y_6	$D_1 D_2$	-	5
a_4	011	a_6	100	1	$y_3 i_1$	-	-	6
a_5	101	a_7	110	1	$y_1 y_2 i_1$	-	-	7
a_6	100	a_1	000	1	$y_1 y_6$	-	-	8
a_7	110	a_8	111	1	$y_4 i_1$	-	-	9
a_8	111	a_1	000	p_L	y_3	-	z_1	10
		a_1	000	$\bar{p}_1 p_2$	$y_2 y_6$	-	z_1	11
		a_5	101	$\bar{p}_1 \bar{p}_2$	y_6	$D_1 D_3$	z_1	12

Столбец Z_h включает переменные $z_r \in Z_1$, определяющие пару ЛУ. Очевидно $z_1 = 1$ для состояния a_8 , переходы из которого зависят от ЛУ из пары B_2 . Это дает систему (9), которая для нашего примера определяется как

$$Z_1 = T_1 T_2 T_3. \quad (14)$$

Функции для блоков LUTerФ LUTerY и формируются из таблицы переходов. Блок LUTerФ определяется системами

$$D_1 = D_3 = \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 \bar{p}_2 \vee T_1 T_2 T_3 \bar{p}_1 \bar{p}_2. \quad (15)$$

$$i_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \vee \bar{T}_1 \bar{T}_2 T_3 \vee \dots \vee T_1 T_2 \bar{T}_3. \quad (16)$$

Блок LUTerY определяется системой (14) и системами

$$\begin{cases} y_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \vee T_1 \bar{T}_2 T_3 \vee T_1 \bar{T}_2 \bar{T}_3; \\ \vdots \\ y_6 = \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 \bar{p}_2 \vee T_1 \bar{T}_2 \bar{T}_3 \vee T_1 T_2 T_3 \bar{p}_1. \end{cases} \quad (17)$$

$$i_2 = \bar{T}_1 T_2 \bar{T}_3 p_1 \vee \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 p_2. \quad (18)$$

Последний этап синтеза связан с использованием стандартных пакетов фирм-производителей FPGA [6,7] и VHDL - Verilog

- моделей автоматов [11, 13]. Мы не рассматриваем этот этап в нашей статье.

Заключение. Замена регистров состояний счётчиками позволяет уменьшить число элементов LUT в схемах автоматов. Это возможно при условии уменьшения числа аргументов в реализуемых СБФ. Для подобного уменьшения, мы предлагаем модифицировать известный метод замены логических условий. Вместо отдельных условий мы предлагаем кодировать их пары. Такой подход имеет смысл, я если число кодирующих переменных остаётся меньше числа внутренних переменных, кодирующих состояния автомата.

Предложенный метод был исследован, используя стандартные тестовые примеры из библиотеки [14]. Исследования проводились с использованием пакета Vivado фирмы Xilinx и нашего программного пакета

ABSYNTH [5]. Пакет [5] генерирует VHDL модели автоматов на уровне языка регистровых передач. В качестве элементного базиса использованы микросхема Virtex - 7 фирмы Xilinx.

Микросхемы семейства Virtex - 7 включают элементы LUT, имеющие $S_L = 6$. Исследования показали, что замена регистров счётчиками и кодирования пар ЛУ позволяет уменьшить аппаратные затраты для схем 35 автоматов (из 48). При этом максимальный выигрыш достигал 19%, а средний - 12%.

Дальнейшее направление наших исследований связано с применением различных методов структурной декомпозиции [5,9] для реализации схем МПА Мили на счётчиках. Кроме того, мы планируем применить эти методы для схем, реализуемых на отличных от FPGA базисах.

Литература

- [1] Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб:БХВ. – Петербург, 2002. – 608с.
- [2] S. Baranov. Logic synthesis for control automata. – Boston: Kluwer Academic Publishers, 1994. - 257 pp.
- [3] I. Skliarova, V. Sklyarov, A. Sudnitson. Design of FPGA-based circuits using hierarchical finite state machines. – Tallinn: TUT Press, 2012. – 240pp.
- [4] R. Czerwinski, D. Kania. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. – Berlin: Springer, 2013. – 182pp.
- [5] A. Barkalov, L. Titarenko, J. Bieganski. Logic synthesis for finite state Machines Based on Linear Chains of States. – Berlin: Springer, 2018. – 218pp.
- [6] Altera: www.altera.com
- [7] Xilinx: www.xilinx.com
- [8] C. Maxfield. The Design Warriors Guide to FPGAs – Orlando: Academic Press, 2004. – 542pp.
- [9] V. Sklyarov, I. Skliarova, A. Barkalov, L. Titarenko. Synthesis and Optimization of FPGA-based Systems. - Berlin: Springer, 2014. – 432pp.
- [10] G. DeMicheli. Synthesis and Optimization of Digital Circuits. – N.Y.: McGraw-Hill, 1994. – 636pp.
- [11] J. Grout. Digital System Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 784 pp.
- [12] A. Barkalov, L. Titarenko, S. Chmielewski. Design of CPLD-based Mealy FSMs with counters. – proceeding of MOCAS 18, 2018. – pp. 1-4.
- [13] Соловьев В. В. Проектирование цифровых систем на основе ПЛИС. – М.: Горячая линия – ТЕЛЕКОМ, 2001. – 636 стр.
- [14] S. Yang. Logic Synthesis and Optimization benchmarks user guide. MCNC. – North Carolina, 1991. – 43pp.

References

- [1] Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб:БХВ. – Петербург, 2002. – 608с.

- [2] S. Baranov. Logic synthesis for control automata. – Boston: Kluwer Academic Publishers, 1994. – 257 pp.
- [3] I. Skliarova, V. Sklyarov, A. Sudnitson. Design of FPGA-based circuits using hierarchical finite state machines. – Tallinn: TUT Press, 2012. – 240pp.
- [4] R. Czerwinski, D. Kania. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. – Berlin: Springer, 2013. – 182pp.
- [5] A. Barkalov, L. Titarenko, J. Bieganski. Logic synthesis for finite state Machines Based on Linear Chains of States. – Berlin: Springer, 2018. – 218pp.
- [6] Altera: www.altera.com
- [7] Xilinx: www.xilinx.com
- [8] C. Maxfield. The Design Warriors Guide to FPGAs – Orlando: Academic Press, 2004. – 542pp.
- [9] V. Sklyarov, I. Skliarova, A. Barkalov, L. Titarenko. Synthesis and Optimization of FPGA-based Systems. – Berlin: Springer, 2014. – 432pp.
- [10] G. DeMicheli. Synthesis and Optimization of Digital Circuits. – N.Y.: McGraw-Hill, 1994. – 636pp.
- [11] J. Grout. Digital System Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 784 pp.
- [12] A. Barkalov, L. Titarenko, S. Chmielewski. Design of CPLD-based Mealy FSMs with counters. – proceeding of MOCAST 18, 2018. – pp. 1-4.
- [13] Соловьев В. В. Проектирование цифровых систем на основе ПЛИС. – М.: Горячая линия – ТЕЛЕКОМ, 2001. – 636 стр.
- [14] S. Yang. Logic Synthesis and Optimization benchmarks user guide. MCNC. – North Carolina, 1991. – 43pp.

Надійшла до редакції 25.04.2019 р.

Баркалов Александр Александрович – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (ул. Подгорная, 50, 65-246 Зелёна-Гура, Польща).
E-mail: a.barkalov@iie.uz.zgora.pl.

Титаренко Лариса Александровна – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (ул. Подгорная, 50, 65-246 Зелёна-Гура, Польща).
E-mail: i.tittarenko@iie.uz.zgora.pl.

Ковалев Сергей Александрович – к. т. н., доцент, Донецький національний технічний університет (пл. Шибанкова, 2, г. Покровськ, 85300, Україна).

E-mail: sergiy.kovalov@donntu.edu.ua

IMPLEMENTATION OF MEALY FINITE-STATE MACHINE ON THE COUNTERS

One of the most important problems associated with the implementation of digital circuits is the reduction of equipment costs. In order to optimize the circuits of control units having irregular structures, methods are considered that reduce the crystal area when implementing the Mealy scheme for Microprogram Automata (MPA). The scheme is implemented on FPGA-type VLSI and the solution of this problem allows to reduce the signal delay time and power consumption.

Typically, Mealy MPA schemes use registers that store state codes. This article looks at the case where the register is replaced by a counter. The scheme is based on table-type elements called LUTs (look-up table). Triggers that are connected via multiplexers to the outputs of the LUT elements are used to implement the counter circuits.

The main feature of LUT elements is a limited number of inputs (5-6). It is considered that the number of inputs equal to 6 is optimal in terms of the area occupied by the element and the energy consumed by it. Since the output signals in MPA Miles depend on the input signals (logical conditions) and states, the system of Boolean functions (SBF) describing the output signals has dozens of arguments. This results in multi-level MPA circuits and, as a consequence, an increase in power consumption and delay time of the signals in the circuit.

To reduce the number of terms in the SBF, the status register can be replaced with a status counter. In this case some transitions are made by increasing the contents of the counter. A method for reducing the number of table type elements (LUTs) in a Mealy circuitry by encoding stable pairs of logical conditions is also proposed. This approach reduces the number of arguments in memory excitation functions. This reduces the number of inputs of the block replacement logic conditions with additional variables. Synthesis example is considered and studies for standard test cases using the Xilinx Vivado package and the ABSYNTH software package have been conducted. Studies have shown that the replacement of registers with counters and coding of logical conditions pairs can reduce the hardware costs for circuits of 35 automata (out of 48). At the same time the maximum gain reached 19%, and the average - 12%.

Keywords: *Mealy FSM, synthesis, FPGA, LUT, pairs of logical conditions.*

Barkalov Aleksandr Aleksandrovich – Prof. Dr., University of Zielona Gora, Inst. Computer Engin. and Electronics (Podgorna Street 50, 65-246 Zielona Gora, Poland).

E-mail: a.barkalov@iie.uz.zgora.pl.

Titarenko Larisa Aleksandrovna – Prof. Dr., University of Zielona Gora, Inst. Computer Engin. and Electronics (Podgorna Street 50, 65-246 Zielona Gora, Poland).

E-mail: i.tittarenko@iie.uz.zgora.pl.

Kovalov Serhii Oleksandrovych – PhD, Associate Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: sergiy.kovalov@donntu.edu.ua