World Applied Sciences Journal 25 (1): 69-77, 2013
ISSN 1818-4952
© IDOSI Publications, 2013
DOI: 10.5829/idosi.wasj.2013.25.01.7025

Application of Decomposition Method to Cyclic Finite State Machine Synthesis with Reconfigurable Time Parameters of Output Signals

*Vasily Grigorievich Rubanov, Elena Nikolaevna Korobkova
and Evgeniy Pavlovich Dobrinskiy*

Belgorod State Technological University named after V.G. Shukhov,
Russia, 308012, Belgorod, Kostyukov Street, 46

Submitted: Aug 17, 2013; **Accepted:** Sep 20, 2013; **Published:** Sep 24, 2013

Abstract: Article considers method of cyclic finite state machine synthesis with adjustable time parameters of output signals based on representation of designed automaton in the form of functional block composition of several interrelated separate blocks, each of which defines one of time parameters of generated pulse sequence. The initial state of the blocks, included in composition, treated as a logical 0 (even if the initial state is selected as a state other than zero). Any other condition of each block in composition treated as a logical 1. In this article performed synthesis of predetermined automata and given its scheme as an example for illustration of the proposed method.

Key words: Digital automaton · Decomposition · Logical functions · Synthesis · Minimization

INTRODUCTION

Interest in logic synthesis [1] renewed due to rapid development of programmable integrated electronics and their implementation in practice of digital devices design and not only at the level of computer-aided design, but at a lower level – "manual synthesis". Problem of "manual synthesis" remains relevant in digital devices synthesis based on small and medium integration level chips during development of standard components and units libraries with their subsequent usage at a higher level of computer-aided design based on PLD (Programmable Logic Devises) and FPGA (Field Programmable Gate Array) [2-4]. In this regard, researches, which are aimed to developing new components and units and improvement of design techniques, enhances their basic characteristics as well as reducing time and cost of development will never lose its relevance.

This work deals with cyclic finite state machines synthesis with adjustable time parameters of output signals used in development of pulse repetition frequency dividers, various sensors and cyclic generators of discrete time intervals, programmable interval timers, stepper motor controllers, control microprogrammed automata with adjustable microcode duration, tone generators in audio systems, lighting effects generators, generators of sequence diagrams and so on.

Researches and publications analysis about problem of blocks and units design, many of which are included in standard library, is the subject of many works, a simple listing of which is not a trivial task [2].

It should be noted that cyclic machines with adjustable output time parameters are not represented in spite of sufficient breadth of known standard components libraries, which are the main building blocks in the arsenal of digital devices developers.

Well-known design methods allow constructing any finite state machine for a specified algorithm. The construction of automaton as a finite state machine with rigid logic does not cause any problems for the case of fixed parameters of output signals. When designing cycle automaton with adjustable parameters, there are problems associated not even with automatic schemes synthesis as such, but finding minimal representation of functions that define configuration parameters specified for automaton, which depends from variables that determine state of automaton ($Q_{i-1}...Q_0$) and variables determine settings for specified parameters.

Corresponding Author: Vasily Grigorievich Rubanov, Belgorod State Technological University named after V.G. Shukhov, Russia, 308012, Belgorod, Kostyukov Street, 46.

Methods of synthesis proposed in works [5-7] based on functions representation in a generalized form that reduce number of definition domain points, which is important because it reduces dimension tables of designed device functioning and as a consequence reduces design time and also provides capability of presented functions analysis with further algorithm simplification of their transformation and optimization for a specified criteria. Statement of the problem is developing of optimization method for cyclic finite state machines design with adjustable time parameters of output signals based on decomposition of designed automaton.

Main Part: We propose a design method of the specified type automata, which simplifies procedure for presenting excitation functions and output functions, that is automaton synthesis procedure as a whole.

Key point of this method is representation of designed automaton in the form of a functional block composition of several interrelated individual blocks.

The theory of finite state machines partition into blocks allows analyzing any specified finite state machine as a unit for possibility of implementing it in the form of smaller finite state machines set is not new [8].

The simplest and most commonly used type of decomposition contains the main automaton and a group of subautomata performing functions assigned to them under the influence of signals generated by the main automaton. Known approach [8] makes no recommendations to the optimal decomposition by the number of blocks included in composition and functions entrusted to them. The end result is in a large extent will depend on designer experience.

In the proposed method each of blocks included in the composition determines one of the time parameter of generated pulse sequence which depends on variety of generator and may be fixed or adjustable (programmable) in its predetermined range of values. Consequently, the number of blocks is the number of generated pulse sequence parameters (for this type generators is generally less than the number of four – five), number of states for separate block (or one greater) are fixed parameter values or is the maximum value of tunable parameters for specified set defined by this block.

Each block is a typical cyclical finite digital machine with general or same initial state for all blocks.

Any typical cyclical automata can be used as initial blocks: binary, summing binary-coded decimal, subtracting or reversible counters, counting circuits with any conversion factor and with any encoding of states [9], cyclic automata made by shift registers [10] (unitary encoding counters, Johnson's counters, linear counters) and others which typically have inputs of synchronous or asynchronous zero state setting.

The initial state of blocks included in composition treated as a logic zero (even if the initial state is state other than zero). Any other condition of each of blocks in the composition treated as logic one level.

In principle, it can be selected any general condition for blocks included in the composition as initial. However, since blocks included in the composition can have overlapping and non-overlapping conditions the most conveniently is taking initial state corresponding to zero of all digits of each composition blocks. If each of blocks is performed by flip-flops with asynchronous inputs zero setting ("Reset"), then each of blocks and, therefore, the whole automaton can be set to zero initial state immediately after the power supply is on for some period of time, which is given by an external resistor and capacitor.

When there is no such inputs or in the case when launch must be synchronous, then is possible to allocate a separate input for each block for signal "Reset", such that when signal is active at this input with the arrival of the first (after the power supply is on) timing pulse, all of blocks move to zero.

Proposed representation of projected finite state machine in the form of composite of two or more types of blocks, each of which is independent from the number of stages and its states, allows us to treat the incoming blocks in the composition as a simple memory elements having two states: zero (if value of all bits is equal to block zero) and the state of logic one level (if at least one of block's bit is one) and the projected automaton as a whole – as a device consisting of two (or more) of these elements.

In this case, function defining state of each block included in the composition can be represented as a logical sum of signals Q_i from outputs of memory elements (flip-flops) forming each block:

$$F = V_{I=0}^{2^{n-1}-1} Q_i,$$

where n – number of block stages including in the composition.

Since the presented function coincides with inverse value of the transfer function P of reversible counter, selecting the initial blocks, which is included in the composition based on requirement of minimum structure of the automaton preference, should be given to reversible counter.

If on the basis of other requirements other cyclical automata used as blocks (binary, summarizing binary-coded decimal counters, counting circuits with any conversion factor with any encoding states, cyclic automata performed by a shift register), then will be required additional n-inputs OR-element to implementation of function determining status of each block included in the composition.

Some species of this class automaton for one or several blocks included in the composition require presence of additional outputs, an active signal of which identify the state block prior to zero and based on the algorithm of their functioning. In typical summarizing binary counters function of indicator performs overflow signal (transfer), which identifies the state of $N = 2^n – 1$ (n-number of counter stages). In typical subtracting counter logical zero state preceded by logical one level, therefore, the indicator function fulfills with minterm defining this state $(m_1 = Q_{n-1}\bar{Q}_{n-1}\bar{Q}_{n-2}\cdots\bar{Q}_1\bar{Q}_0)$. Thus, an additional element that implements this minterm will require for organization of such output. If cyclical automata configured on the shift register with shifting towards lower digits used as block that defines some parameter, then indicator function also serves with minterm defining logical one level state of register. In case of using a shift register with shift towards most significant digits the indicator function performs minterm defining the contents of the register equal to one in the most significant digit and zeros in all other digits, that is $(m_1 = Q_{n-1}\bar{Q}_{n-1}\bar{Q}_{n-2}\cdots\bar{Q}_1\bar{Q}_0)$. Similarly we can define an indicator function for any kind of used block based on the nature of its transitions.

Design of specified finite state machine begins with its verbal description followed by drawing up state table as in case of classical methods. Total number of state table sets (including unused) is 2^n for classical design methods where n is the number of memory elements (flip-flops, digits) of designed automaton as a whole.

In proposed method total number of sets in table are 2^{n+k}, n is number of blocks, k – number of additional outputs (if there is a necessity for their presence), which values are independent from the number of used blocks digits, which significantly reduces the total number of state tables sets in comparison with it classical representation.

As a further illustration to following procedure of drawing up tables it is recommended to present time charts showing the expected behavior of designed automaton for one or more sets of control signals that determine parameters of output pulse sequence. As a verbal description of the automaton behavior is not mathematically rigorous, then graph of automaton constructed before going to the operation table.

A distinctive feature of the proposed method is that the graph transitions of the automaton is represented as a closed ring equal to the number of composition blocks with a common vertex corresponding to the initial (zero) state of each block.

In this representation each subsequent state S^{n+1} of single block in (n + 1)-th tact does not depend on its particular state S^n in this tact, but on the state of all blocks and signal values at the additional outputs if they are available. Block state characterized by signal value P at its main output (at transfer output P in case of usage subtracting counter) and the automaton state in general in (n + 1)-th tact determined by signals at all outputs of blocks in the n-th tact.

Under each clock cycle one or more blocks can remain in the same state (storage mode), can move to the next state in accordance with operation algorithm of this block (sequential mode transition), can move to any state of its sets defined by external control word (parallel loading mode), can move to the initial (zero) state (zero mode).

Each of blocks must have appropriate inputs to allow these transitions and formation of the active signal (equal to 0 or 1 depending on the selected scheme) on one of inputs provides selected mode. In particular, typical reversible counters have inputs such as typical chip ALS 5694, which is a reversible four-digit binary counter, which has an input for clock pulses supply C, input for settings to addition / subtraction mode U, input to enable synchronous parallel load L and supply inputs for downloadable data $D_0 – D_{n-1}$, input to enable count mode P_0, synchronous (SR) and asynchronous (AR) inputs for setting to zero state, output of overflow P_4. If other cyclical automata (typical or atypical) used as blocks and it does not have sufficient number of control inputs, then firstly it must be necessary to organize a lot of them, which are enough to allow the required transitions.

Number and types of inputs for blocks depend on the algorithm of automaton, in particular, blocks that define tunable parameters must have input to enable synchronous parallel load L, supply input for downloadable data $D_0 – D_{n-1}$, input to enable counting mode (sequential transition) P_0 and input of asynchronous setting to zero state R. For blocks that define fixed parameters, one must have input permits the sequential transition (count number) and input of asynchronous setting to zero state R.

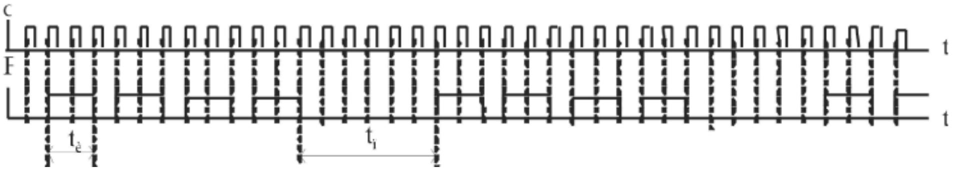

Fig. 1: Time diagrams illustrating the operation algorithm

The presence of input for settings on addition / subtraction mode *U* is necessarily if typical microchip of reversible counter used as the subtracting counter (because chip of typical subtracting counter is not released by industry) and in the case of accordance with operation algorithm of automaton it becomes necessary to passing from summing to subtraction mode and vice versa. The presence of synchronous setting input to zero state is optional.

After choosing the required number and type of blocks determined in accordance with the descriptive algorithm (given diagram) states table and excitation functions table combined with it (values of control signals ensuring transition from considered state to the following) in accordance with given graph. State table contains (n + 1) columns (n – number of blocks, k – number of additional outputs), in which are listed all combinations of signal values (variables) at outputs of blocks (even if some of combinations in automaton operation are not used) and n columns, each line of which commenting on block setting mode for this combination of signal values.

Table of excitation functions (control) contains the number of columns equal to the total number of control signals plurality "$L, D_0 - D_{n-1}$, P_0, R, U" for all blocks included in the composition. Table fills in accordance with given comment of blocks reaction (setting up modes) for each combination and it should be borne in mind that priority assigned to each control input, so if active value of control signal (value of the excitation function) at the input, which has a higher priority, has stamped in some row of the table for some block, then the signal at the other inputs of this block with a lower priority can be put down to 0 or 1 (sign of redundancy).

Thus, obtained table allows finding required excitation function (control) defining signals sent to the control inputs of each block. In finding the excitation functions unused and redundant combination extends the definition of value of 0 or 1 based on requirement of minimizing structure of automaton or risk eliminating. Excitations functions and their scheme of realization are quite simple as the number of units equal to the number of parameters of the output pulse sequence does not exceed two – five.

Output signal more often get from output of one of blocks or is a function of signals from outputs of several blocks.

Let's consider periodic synthesis generator of pulse packets sequence with fixed number of them in a pack equal to four, tunable pause between packets, reconfigurable fixed duration and packet pulses pause equal to one cycle as an example to proposed method illustration.

In Figure 1 shown time diagrams (epures) illustrating operation algorithm of designed generator for variant of settings: pulse width in a pack t_p = 2T, pause between pulses in the pack is T, number of packet pulses is four, length of pause between packets is 6T.

Analyzing given diagrams, at first glance might be concluded that the projected automaton characterized by four parameters: variable pulse duration in the pack, fixed pause of packet pulses equal to one cycle, fixed number of packet pulses equal to four, tunable pause between packets. However, as the pause between pulses in a pack is equal to one cycle, then functions defining these two parameters can be blamed to a single block. The best option of such block will be subtracting counter with input of enabling synchronous parallel load L, inputs for supplying of downloaded data $D_0 - D_{n-1}$ and input of asynchronous zero setting. Number of counter digits and consequently digit binary word $B = b_{n-1}...b_1b_0$ applied to inputs $D_0 - D_{n-1}$ determines adjustment range of pulse duration. If the typical digital counter used as this block, then pulse duration will range from T to $(2^n - 1)T$, where T is the period of time pulses supplied with output of the external clock generator.

Number of pulses in a pack is equal to four and fixed, consequently, functions defining this option could be assigned to any cyclic automaton with four states having an input enabling consecutive transition (count) and input of asynchronous zero setting.

Pause duration between packets is tunable, consequently, functions defining this option could be assigned to the subtracting counter with input to enable synchronous parallel load L, inputs for supplying of downloaded data $D_0 - D_{n-1}$ and input of asynchronous zero setting.

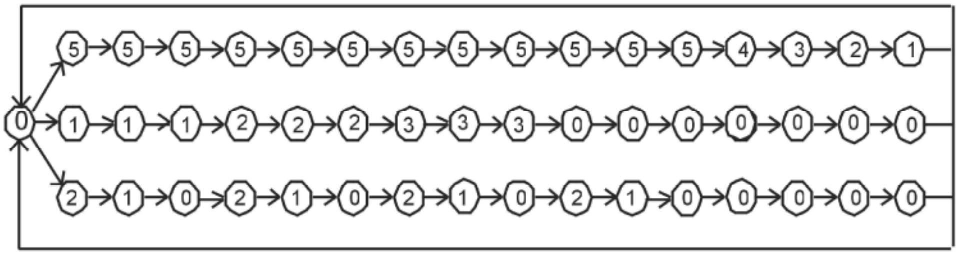

Fig. 2: Transitions graph of the automaton

The number of counter digits and, consequently, digit binary word $P = p_{n-1}...p_1p_0$ applied to inputs $D_0 - D_{n-1}$, determines the range adjustment of pause duration between impulse packets. If typical digit counter used as this block, then pause duration between impulse packets will be in the range from T to 2^nT.

Let's represent automaton algorithm with preset parameters in the form of a graph for more detailed perception of it (Fig. 2).

Graph consists of three rings with a common vertex, corresponding to the initial (zero) state of all (three) blocks (counters): bottom ring – transition graph of the block, which determines the pulse and pauses duration between them, middle ring – transition graph of the block, which determines number of pulses in pack, top ring – transition graph of the block, which determines duration of pause between impulse packets. Vertices of presented graph represented in the form of circles inside of which written state of the counter. In order to simplify presentation of the graph value of signal is not written at outputs of blocks, which is equal to 0 if condition of the block is zero and is equal to 1 if its status is not zero.

Let's assume that after power supply is on for a period of time determined by speed of hardware component, installation of them has taken to initial (zero) position. Signals at the outputs of all three blocks $P_3 = P_2 = P_1 = 0$. Value of these signals (zero set of variables P_3 P_2 P_1 = 000) provides setting of first and third counters to mode of synchronous parallel load and second counter- to mode of sequential transition to the first non-zero state.

At the time of entry first timing pulse, first counter transitions to a state defined by value of four-digit binary word $P_3P_2P_1P_0$ = 0010 supplied to its inputs of parallel load $D_3D_2D_1D_0$, that is the state two. Second block transitions to the first non-zero state, that is to one, if it is used ordinary two-bit summing counter as a second counter. Third counter transitions to a state determined by value of four bit word $p_3p_2p_1p_0$ = 0101 supplied to its inputs of parallel load $D_3D_2D_1D_0$,, that is in the state five. Logic one level generated at outputs $P_3P_2P_1$ of all three blocks ($P_3 = P_2 = P_1 = 1$) as result of this transition. Proceeding from given graph, there is shown that the value of these variables (seventh set of variables $P_3P_2P_1$ = 111) provides configuration of first counter to counting mode, second and third – to storage mode and then at the time of next timing pulse entry first counter (subtracting) passes in state B-1, that is to 0001, but states of second 0001 and third 0101 counters will remain unchanged. Arguing the same way, there are defined modes and corresponding transitions for all vertices of given graph with the note that there is no necessity for additional outputs that identify individual state of blocks.

Results of these considerations are presented in the form of state table (the first three columns – sets of variables defined by values of signals P_3,P_2,P_1 at outputs of blocks, the second three columns – configuration modes of counters defined by these sets) and table of excitation functions (control) combined with it $\overline{L}_1, \overline{E}_{01}, E_2, \overline{L}_3, \overline{E}_3$ (last five columns). In representation of table modes and corresponding values of excitation functions are marked with * for unused and redundant sets.

To find analytic representation of excitation functions we use a method based on their representation in Karnaugh maps and function values of unused and redundant sets (#) will redefine with value 0 or 1 based on requirements for obtaining minimum of disjunctive normal form (DNF) for functions $\overline{L}_1, \overline{E}_{01}, E_2, \overline{L}_3, \overline{E}_3$.

Source and redefined map of excitation function (control) $\overline{L}_1$ of first counter are shown in Fig. 3a, b.

While extending the definition of function values $\overline{L}_1$ for first and third sets value with logic one and for second set with value of zero (Fig. 3b), we find the minimum DNF of excitation function

$$\overline{L}_1 = P_1 \vee P_3\overline{P}_3.$$

Source and redefined map of excitation function (control) $\overline{E}_{01}$, of first counter are shown in Fig. 4a, b.

Table 1: Table of states and excitation functions

Output signals			Counter modes				Values of excitation functions			
P_1	P_2	P_2	CT1	CT2	CT3	$\bar{L}_1$	$\bar{E}_{01}$	E_2	$\bar{L}_3$	$\bar{E}_3$
0	0	0	Load	Count	Load	0	*	1	0	*
0	0	1	*	*	*	*	*	*	*	*
0	1	0	*	*	*	*	*	*	*	*
0	1	1	*	*	*	*	*	*	*	*
1	0	0	Store	Store	Count	1	1	0	1	0
1	0	1	Count	Store	Store	1	0	0	1	1
1	1	0	Load	Count	Store	0	*	1	1	1
1	1	1	Count	Store	Store	1	0	0	1	1

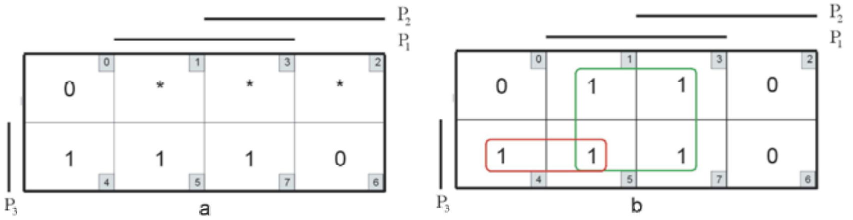

Fig. 3: Source and redefined maps of first counter function $\bar{L}_1$

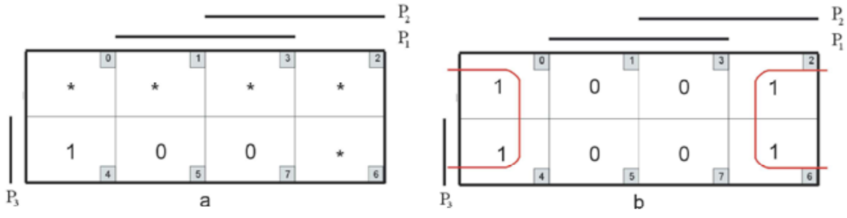

Fig. 4: Source and redefined maps of first counter function $\bar{E}_{01}$,

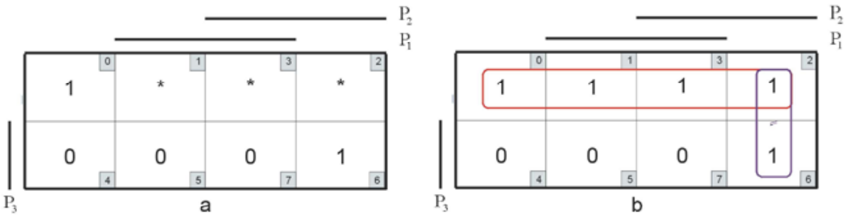

Fig. 5: Source and redefined maps of second counter function E_2

Extending the definition of function values $\bar{E}_{01}$, at zeroth, second and sixth sets with value of logic one and at first and third sets with value of logic zero (Fig. 4b) we find minimum DNF of excitation function $\bar{E}_{01} = \bar{P}_1$.

Source and redefined map of excitation function (control) E_2 of second counter are shown in Fig. 5a, b.

Extending the definition of function values E_2 at first, second and third sets with value of logical one (Fig. 5b), we find minimum DNF of function with subsequent conversion to its implementation on two-input NAND elements:

$$P_2 = \bar{P}_3 \vee P_2\bar{P}_1 = \overline{P_3\overline{P_2\overline{P}_1}}$$

Source and redefined map of excitation function $\bar{L}_1$ of third counter are shown in Fig. 6a, b.

Extending the definition of function values at first, second and third sets with value of 0 (Fig. 6a), we find minimum DNF of excitation function:

$$\bar{L}_3 = P_3$$

Source and redefined maps of excitation function $\bar{E}_{03}$ of third counter are shown in Fig. 7a, b.

Extending the definition of function values $\bar{E}_{03}$ at first, second and third sets with value of logical one and at zeroth set with value of logic zero (Fig. 7b), we find minimum DNF of excitation function $\bar{E}_{03} = \bar{P}_1 \vee P_2$.

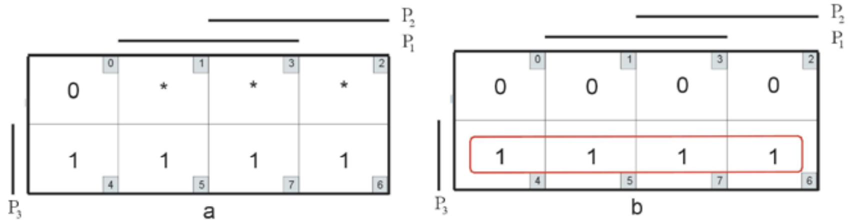

Fig. 6: Source and redefined maps of third counter function $\bar{L}_1$

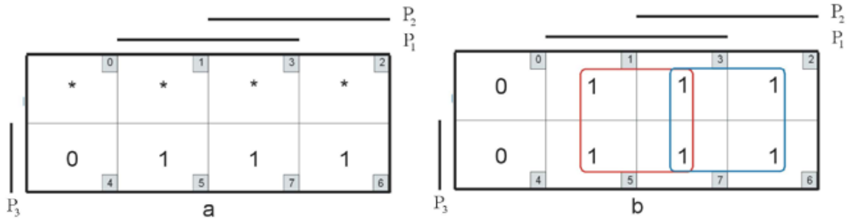

Fig. 7: Source and redefined maps of third counter function $\bar{E}_{03}$

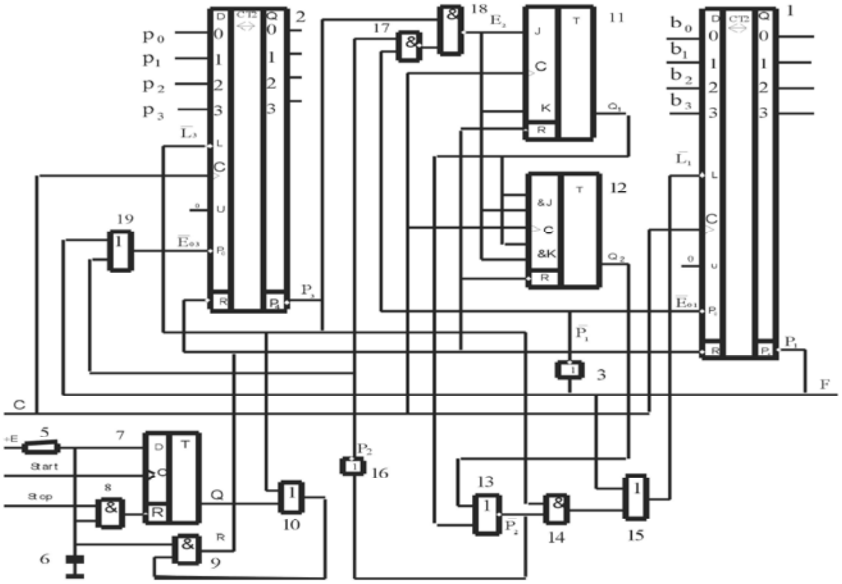

Fig. 8: Generator circuit

However, if construct a circuit just in accordance with obtained expressions, then its usage is practically limited since beginning and ending of the output pulse sequence generation associated with a moment of power supply on / off. In real digital devices transfer of all elements of its memory (flip-flops) occurred to pre-specified (usually zero) state when the power source is up. This state is often initial state of automaton, but for beginning of functioning apart from having a detection circuit of switching power supply up to provide this transition, practically there are always required not only start-up input, but often reset input supplying, active signal values of which provides these operations.

If automaton designed to generate a periodic pulse sequence, then one of the main requirements for the reset circuit is to prevent distortion of pulse sequence generation until the end of its full cycle, regardless of receipt moment of reset signal. In our automaton start and reset functions are entrusted to a separate block – start / stop device configured at synchronous D-trigger with input of asynchronous zero setting. Generator circuit shown in Fig. 8 and constructed in accordance with given remark and the above representation of control functions $\bar{L}_1, \bar{E}_{01}, E_2, \bar{L}_3, \bar{E}_3$.

Generator contains four interconnected blocks and groups of elements providing control of blocks in accordance with a predetermined functioning algorithm. First block (1) – reversible four-digit binary counter configured to subtraction mode ($U = 0$), second block – two-digit summarizing counter made on two JK-triggers 11, 12, third block (2) – reversible three-digit binary counter configured on subtraction mode, fourth block – start / stop device which contains synchronous D-trigger 7 with input of asynchronous setting to zero state R, chain composed of series connected resistor 5 and capacitor 6, two-input AND elements 8, 9 and two-input OR element 10.

Elements providing control of blocks can also be divided into groups: a group of elements generating parallel load enable signal for the first counter $\overline{L}_1$ –two-input NOR element 13, two-input AND element 15, two-input OR element 16, element generating enable signal of count mode for the first counter $\overline{E}_{01}$ – inverter 3, group of elements generating enable signal of count mode for the second counter E_2– inverter 14, two-input NAND element 17, two-input NAND element 18, element generating enable signal of count mode for the third counter $\overline{E}_{03}$ – two-input OR element 19.

Inputs $D_3D_2D_1D_0$ of the first counter form inputs b_3b_2 b_1b_0 of generator settings to predetermined output pulses duration (t_i). Inputs $D_3D_2D_1D_0$ of the third counter form inputs $p_3p_2p_1p_0$ of generator settings to predetermined pause duration between impulse packets (t_p).

Input C – supply port of clock pulses from the output of external generator. Input Start – input of run pulse supply. Input Stop – input of stop pulse supply for generation output pulse sequence. Output overflow for the first counter forms output of generator F.

Summary: First proposed representation of designed automaton in the form of functional block composition of several interrelated individual blocks, the initial state of which treated as a logic 0 (even if as the initial state chosen state other than zero). Any other state of each of input blocks in the composition is treated as a logic one, thereby, ensuring independence number of sets from the number of used blocks digits. Any other condition of each of the units in the composition is treated as a logic one, thereby, ensuring the independence of sets number from number of bits in used blocks. Article considers method of synthesis. There was synthesized periodic sequence of generator pulse packets with fixed number of them in pack with reconfigurable pause between packets, reconfigurable and fixed-paused pulse duration in the pack according to the proposed method. There is shown circuit of generator made on chips of small and medium-level integration.

CONCLUSIONS

Presented scientific novelty of the work. Proposed method of synthesis digital automata with adjustable time parameters of output signals based on representation of designed automaton in the form of functional block composition of several interrelated separate blocks, each of which defines one of time parameters for generated pulse sequence.

Practical value of results is that unlike conventional approaches to digital automata synthesis, total number of state table sets does not depend on capacity of used blocks that considerably simplifies synthesis procedure in comparison with known classical methods.

Direction of further researches involves the development of technology-aided design of digital automata with adjustable time parameters of output signals in the form of functional block composition of several interrelated individual blocks based on programmable logic of integrated circuits.

Research performed under the grant 8.4656.2011 of the Ministry of Education and Science of the Russian Federation on theme: "Development and research design techniques for controlled mobile logistics equipment having the property of survivability" and grant A-20/12 under the program of strategic development of BSTU named after V.G. Shukhov.

REFERENCES

1. Uaykerli Dzh, F., 2002. Proyektirovaniye tsifrovykh ustroystv (Design of digital devices). Postmarket, pp: 544.
2. Solov'yev, V.V., 2001. Proyektirovaniye tsifrovykh sistem na osnove programmiruyemykh logicheskikh integral'nykh skhem (Design of digital systems based on programmable logic integrated circuits). Goryachaya liniya-Telekom, pp: 636.
3. Sentovich, E.M., K.J. Singh, L. Lvagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton and A. Sangionvanni-Vincentelli, 1992. SIS: A system for sequential circuit synthesis. UCB/ERL M92/41, Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkley.
4. Villa, T., T. Kam, R.K. Brayton and A. Sangionvanni-Vincentelli, 1998. Synthesis of Finite State Machines: Logic Optimization. Kluwer Academic Publishers, Boston.

5. Kharchenko, V.S., Ye. N. Korobkova, N.G. Korobkov and V.G. Rubanov, 2008. Obobshchonnyye logicheskiye funktsii i sistemy na programmiruyemoy logike (Generalized logical functions and systems on a programmable logic). Ministerstvo obrazovaniya i nauki Ukrainy, Natsional'nyy aerokosmicheskiy universitet im. N.Ye. Zhukovskogo «KHAI», pp: 351.
6. Rubanov, V.G. and Ye. N. Korobkova, 2009. Logicheskoye proyektirovaniye tsifrovykh ustroystv, osnovannoye na predstavlenii funktsiy v obobshchonnoy forme (Logical design of digital devices based on representation of functions in generalized form). Belgorod, Izd-vo BGTU.
7. Rubanov, V.G. and Ye. N. Korobkova, 2009. Metody analiza i sinteza tsifrovykh ustroystv (proyektirovaniye tsifrovykh elementov avtomatiki i vychislitel'noy tekhniki) (Methods of analysis and synthesis of digital devices (digital elements design of Automation and Computer Engineering)). Belgorod, Izd-vo BGTU, pp: 293.
8. Yakobson, G.E., 1971. Parallel'naya dekompozitsiya konechnykh avtomatov (Parallel decomposition of finite state machines). AVT, 1: 1-7.
9. Amann, R. and U. Baitinger, 1974. Optimal state chains and state codes in finite state machines. IEEE Transactions on CAD, 8(2): 153-170.
10. Devyatkov, V.V., 1974. Metody realizatsii konechnykh avtomatov na sdvigovykh registrakh (Methods of finite state machines implementation with shift registers). Energiya, pp: 80.