Modelling and Synthesis of Printed Circuit Boards Testing Systems based on Timed Hard Petri Nets

Sudacevschi Viorica, Ababii Victor, Calugari Dmitri, Bordian Dimitrie
Technical University of Moldova, Informatics and Systems Engineering, Chisinau, R. Moldova,
Viorica.sudacevschi@calc.utm.md, Victor.ababii@calc.utm.md,
Dmitri.calugari@scec.utm.md, Dimitrie.bordian@scec.utm.md

Abstract **- This paper presents a synthesis method for delay time evaluation in the printed circuit boards based on Timed Hard Petri Nets. For the specification and modeling of the delay time evaluation system, Timed Synchronous Petri Nets (TSPN) are used, which allow conflicts identification and exclusion related to both the processes time synchronization and processes timing constraints. The developed TSPN model of the delay time evaluation system comprises the test signal generator and the time delay analyzer. The transition to the hardware description of the system is achieved by translating the TSPN into Timed Hard Petri Net (THPN). THPN consists of processing elements and logical connections between them. For each processing element analytical model and AHDL code were developed. The implementation of the delay time evaluation system was done by direct mapping of the THPN into the Field-Programmable Gate Array (FPGA) circuits. FPGA architectures present advantages such as high parallelism, control processing speed-up and reconfigurability option. The direct mapping method has a linear algorithmic complexity and is not affected by state explosion problem. The transparent correspondence between the elements of the initial specification and the components of the resultant circuit ensures that the timing constraints under which the evaluation system is designed are respected.**

Keywords: *Printed Circuit Boards, Timed Hard Petri Nets, Delay Time, HDL, FPGA.*

I. Introduction

The advantages of FPGAs, such as high integration scale and their high speed processing time with the reconfigurability option make this type of circuits an attractive solution for many types of applications. One of the perspective directions is the use of reconfigurable circuits for parametric testing of printed circuit boards [1-3]. The interest in this approach is based on the ability to simultaneously test multiple dependent signals [4] and to perform, if necessary, the real-time reconfiguration of the testing system.

Some common problems that appear in the design of delay time evaluation systems in printed circuit boards are the synchronization of the test signal generator and difficulties with test signals acquisition. Certain operations are executed within predetermined time limits, so that processing is time constrained. In these cases, time is the basic measure, and time constraints require a very high accuracy which can be achieved by using parallel or concurrent data processing techniques [4].

The implementation of parallel data processing algorithms requires verification of the correct functioning and occurrence of conflicts, which can lead to serious errors. For this purpose, modern methods and techniques are used based on the application of Timed Petri Nets [5, 8, 9]. These techniques allow conflicts identification and exclusion related to both the processes time synchronization and processes timing constraints.

The classic implementation methods of printed circuit boards test systems based on logical synthesis have a number of disadvantages, namely: high computational complexity, system specification only at low abstraction levels, the structure of the resulting circuits does not correspond to the structure of the functional model [1-3]. The direct mapping techniques of the system model in the circuit exclude these disadvantages, which is particularly important for real-time systems with timing constraints.

In the paper is proposed a synthesis method of systems for evaluation of delay time in printed circuit boards based on Timed Hard Petri Nets (THPN). The implementation of the delay time evaluation system is done by direct mapping of the THPN into the reconfigurable hardware architecture (FPGA). The direct mapping method has a linear algorithmic complexity, is not affected by state explosion, so large digital systems can be constructed at low cost. Direct mapping facilities checking of the functional correctness of the implementation and ensures that the timing constraints under which the evaluation system is designed are respected, because of the transparent correspondence between the elements of the initial specification and the components of the resultant circuit.

II. Synthesis Algorithm for Delay Time Evaluation in the Printed Circuit Boards

Let's consider the system, the main task of which is to calculate the delay time induced by propulsion of electrical signals in the printed circuit boards. The functional diagram of the system is shown in Fig. 1 and includes: $\boldsymbol{TG}$ - test signals $\boldsymbol{U^{Out}}$ generator; $\boldsymbol{DTE}$ - the time delay evaluator of the input signals $\boldsymbol{U^{In}}$; $\boldsymbol{S}$ - synchronization signal of the time delay evaluation unit; $\boldsymbol{PCB}$ - printed circuit board; $\boldsymbol{EI}$ - the influence of the outside environment on the printed circuit board with noise signals $\boldsymbol{U^{EI}(t)}$.

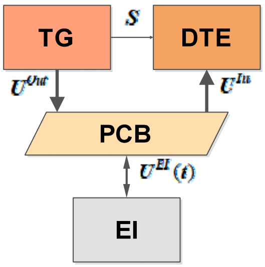


Fig. 1. Functional diagram of a printed circuit board delay time evaluation system.

The abstract model of the printed circuit board $\boldsymbol{PCB}$ is shown in Fig. 2, where: $\boldsymbol{OXY}$ - the coordinate system that determines the position of the objects on the board; $U^{Out}=\left\{U^{Out}_{x,y}(t), \forall x=\overline{0,X}, y=\overline{0,Y}\right\}$ - the set of test signals that are defined in space P and time t; $U^{Out}=\left\{U^{In}_{x,y}(t), \forall x=\overline{0,X}, y=\overline{0,Y}\right\}$ - the set of signals for delay time evaluation that are defined in space P and time t; $\theta=\left\{\tau_{x,y}, \forall x=\overline{0,X}, y=\overline{0,Y}\right\}$ - delay time generated by printed circuit board $\left[U^{Out}_{x,y}(t), U^{In}_{x,y}(t)\right]$, D - mathematical model that determines the ratio between $U^{Out}_{x,y}(t)$ and $U^{In}_{x,y}(t)$ [4].

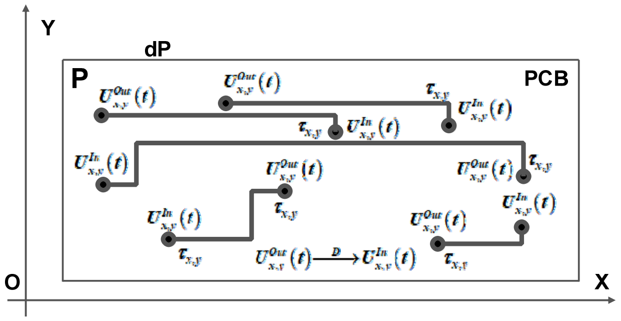


Fig. 2. The abstract model of the printed circuit board.

Fig. 3 presents a block diagram of the synthesis algorithm for modeling and implementation of the delay time evaluation system in printed circuit boards based on Hard Petri Nets.

The synthesis algorithm includes the following steps:

$\boldsymbol{VPNP}$ - the Petri Net modeling environment [5,6] which includes the following operations: $\boldsymbol{PNMI}$ - manual input $\boldsymbol{MI}$ or uploading the Timed Synchronous Petri Net model from the storage device $\boldsymbol{DS}$; $\boldsymbol{PNMS}$ - Petri Net modeling; $\boldsymbol{PEPNM}$ - evaluation of the functional and performance parameters of the TSPN model that determines the functionality of the delay time evaluation system in the printed circuit boards.

$\boldsymbol{HPNC}$ - environment for translating Timed Synchronous Petri Net model into Timed Hard Petri Net [7-9] which includes the following operations: $\boldsymbol{APNM}$ - parameter analysis of the TSPN model in order to extract the connections between the functional elements for Timed Hard Petri Net; $\boldsymbol{T\ PNM\ to\ HPNM}$ - translating the TSPN model into a Timed Hard Petri Net; $\boldsymbol{HDLG}$ - generation of the Hardware Description Language (HDL) code of the system that is converted in a gate level netlist and can be implemented into FPGA circuit.

$\boldsymbol{Quartus-II}$ - the HDL compile environment and the FPGA configuration that includes the following operations: $\boldsymbol{A\ HDL}$ - analysis of the HDL code; $\boldsymbol{C\ FPGA}$ - configuration of the FPGA device.

$\boldsymbol{PCB\ DTE}$ - evaluation of the delay time in the printed circuit board.

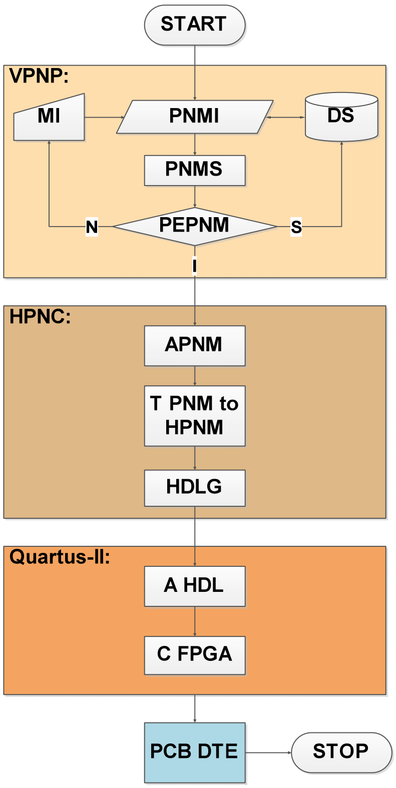


Fig. 3. Synthesis algorithm.

III. Functional Diagram of the System for Delay Time Evaluation in the Printed Circuit Boards

Functional diagram of the system for evaluation of the delay time in the printed diagram boards is presented in Fig. 4.

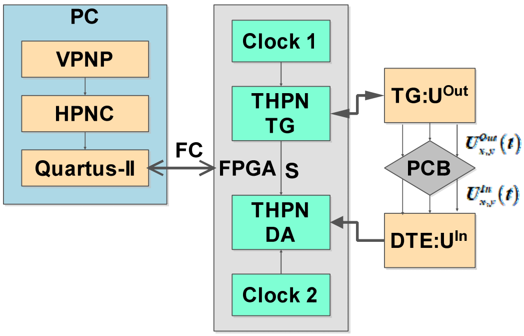


Fig. 4. Functional diagram of the system.

The functional diagram includes the following modules:

$\boldsymbol{PC}$ - the personal computer with $\boldsymbol{VPNP}$ environment for Petri Net modeling [6], $\boldsymbol{HPNC}$ compiler and the design tool $\boldsymbol{Quartus-II}$;

$\boldsymbol{FPGA}$ - reconfigurable circuit for implementation of the test generator based on Timed Hard Petri Nets $\boldsymbol{THPN\ TG}$ with clock generator $\boldsymbol{Clock}\ 1$, and delay time analyzer based on Timed Hard Petri Nets $\boldsymbol{THPN\ DA}$ with clock generator $\boldsymbol{Clock}\ 2$;

$\boldsymbol{TG : U^{Out}}$ - the set of connectors for contacting the surface of the printed circuit board $\boldsymbol{PCB}$ for test signal transmission $\boldsymbol{U^{Out}_{x,y}(t)}$;

$\boldsymbol{DTE : U^{In}}$ - the set of connectors for contacting the surface of the printed circuit board $\boldsymbol{PCB}$ for the acquisition of input signals $\boldsymbol{U^{In}_{x,y}(t)}$.

IV. Timed Synchronous Petri Net Model

The Timed Synchronous Petri Net model was developed in $\boldsymbol{VPNP}$ environment [6]. It is used as a design entry for Timed Hard Petri Net model generation. The model for delay time evaluation system consists of $\boldsymbol{M}$ homogeneous channels running in parallel. In Fig. 5 submodels for channels $\boldsymbol{1}$ and $\boldsymbol{M}$ are presented. Each submodel generates the delay time of input signals $\boldsymbol{U^{In}_{x,y}(t)}$ and produces the testing output signals $\boldsymbol{U^{Out}_{x,y}(t)}$.

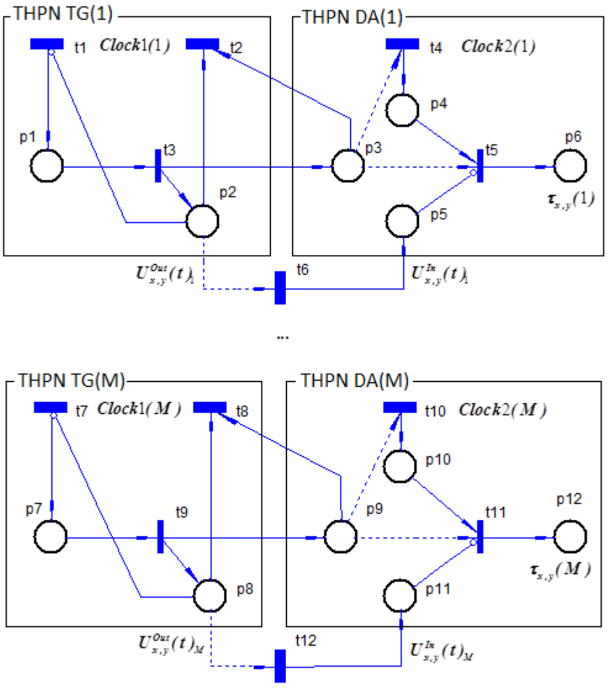


Fig. 5. The Timed Synchronous Petri Net model.

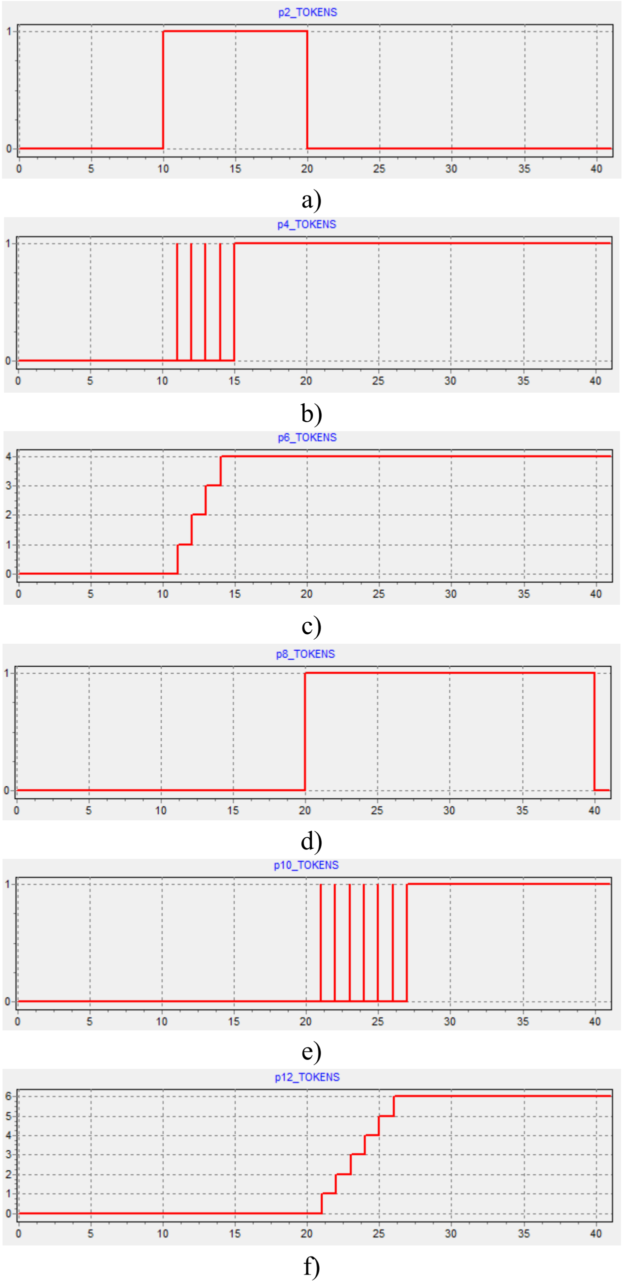
The Timed Synchronous Petri Net model comprises two basic components:

$\boldsymbol{THPN\ TG(i), \forall i = \overline{1,M}}$ - the test signal generator that consists of: $\boldsymbol{t}1$, $\boldsymbol{t}2$ ($\boldsymbol{t}7$, $\boldsymbol{t}8$) - timed transitions that determine the logic value 0 or 1 of the test signal (together form the clock generator $\boldsymbol{Clock}\ 1(1),...,\boldsymbol{Clock}1(\boldsymbol{M})$); $\boldsymbol{p}2$ ($\boldsymbol{p}8$) - discreet places that represent the output signals $\boldsymbol{U^{Out}_{x,y}(t)_i, \forall i = \overline{1,M}}$; $(\boldsymbol{t}3,\boldsymbol{p}3)((\boldsymbol{t}9,\boldsymbol{p}9))$ - synchronization arcs that validate the beginning of the delay time evaluation.

$\boldsymbol{THPN\ DA(i), \forall i = \overline{1,M}}$ - the time delay analyzer that consists of: $\boldsymbol{t}4$ ($\boldsymbol{t}10$) - timed transitions (clock generators $\boldsymbol{Clock}\ 2(1),...,\boldsymbol{Clock}2(\boldsymbol{M})$) that generate pulses to evaluate the delay time; $\boldsymbol{t}5$ ($\boldsymbol{t}11$) - non-timed transitions that synchronize the time interval for the calculation process; $\boldsymbol{p}5$ ($\boldsymbol{p}11$) - discreet places that represent the input signal $\boldsymbol{U^{In}_{x,y}(t)_i, \forall i = \overline{1,M}}$; $\boldsymbol{p}6$ ($\boldsymbol{p}12$) - discreet places that accumulate the impulses generated by $\boldsymbol{t}4$ ($\boldsymbol{t}10$), which then allow the evaluation of the delay time $\boldsymbol{\tau_{x,y}(i), \forall i = \overline{1,M}}$.

Timed transitions $\boldsymbol{t}6$ ($\boldsymbol{t}12$) are used in order to model the delay time of the signals in the printed circuit board.

Fig. 6 shows the time diagrams for channel $\boldsymbol{1}$ obtained as a result of Timed Synchronous Petri Net modeling in $\boldsymbol{VPNP}$ environment [6].



a)

b)

c)

d)

e)

f)

Fig. 6. Time diagrams obtained as a result of Timed Synchronous Petri Net modeling in VPNP environment.

For functional testing of the Timed Synchronous Petri Net model for delay time evaluation in printed circuit boards, it was considered that the timed transition

$\boldsymbol{t}6$ remains enabled for $\tau_{x,y}(1) = 4\boldsymbol{\mu s}$ before firing. Correspondently, timed transition $\boldsymbol{t}12$ remains enabled for $\tau_{x,y}(\boldsymbol{M}) = 6\boldsymbol{\mu s}$.

The state of the place $\boldsymbol{p}2$, which represents the output signal $\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_1$ is described by the time diagram in Fig. 6, a. The waveform is determined by the delay time generated by the transitions $\boldsymbol{t}1\left(\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_1 = 0 \middle| \Delta = 10\boldsymbol{\mu s}\right)$ and $\boldsymbol{t}2\left(\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_1 = 1 \middle| \Delta = 10\boldsymbol{\mu s}\right)$.

The time diagram for place $\boldsymbol{p}4$ that represents the clock signal for delay time incrementation is shown in Fig. 6, b. The duration of one clock period is $1\boldsymbol{\mu s}$.

The evaluation of the delay time of the signal $\tau_{x,y}(1) = 4\boldsymbol{\mu s}$, induced by place $\boldsymbol{p}6$, is done by counting the pulses generated by $\boldsymbol{p}4$ (Fig. 6, c).

Fig. 6, d shows the time diagram of the signal generated by place $\boldsymbol{p}8$, which represents the output signal $\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_{\boldsymbol{M}}$. The waveform of this signal depends on delay time of transitions $\boldsymbol{t}7\left(\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_{\boldsymbol{M}} = 0 \middle| \Delta = 20\boldsymbol{\mu s}\right)$ and $\boldsymbol{t}8\left(\boldsymbol{U}^{Out}_{x,y}(\boldsymbol{t})_{\boldsymbol{M}} = 1 \middle| \Delta = 20\boldsymbol{\mu s}\right)$.

The time diagram for place $\boldsymbol{p}10$ that represents the clock signal for delay time incrementation is shown in Fig. 6, e. The duration of one clock period is $1\boldsymbol{\mu s}$.

The time diagram for counting the pulses generated by place $\boldsymbol{p}10$ is shown in Fig. 6, f. This process evaluates the delay time of the signal $\tau_{x,y}(\boldsymbol{M}) = 6\boldsymbol{\mu s}$, induced by place $\boldsymbol{p}12$.

When the system will be implemented in reconfigurable architecture, the delay time induced by the places $\boldsymbol{p}6$ and $\boldsymbol{p}12$ will depend on the waveform and physical parameters of the printed circuit board conductor.

V. Timed Hard Petri Net Model

Timed Hard Petri Nets (THPN) was proposed to perform the conversion of the Timed Synchronous Petri Nets model to logic circuit. THPN model consists of functional elements and logical connections between them. THPN represents the hardware model of TSPN and completely repeats the topology and connections between positions P and transitions T. THPN model is generated by the ***HPNC*** compiler (Fig. 3) and it is used to produce the Altera Hardware Description Language (AHDL) code of the designed system. The direct implementation of the THPN model into FPGA circuit ensures that the behavioral properties and time constraints, under which the system works, will be respected.

Timed Hard Petri Nets is a 14-tuple:

$$\boldsymbol{THPN} = <\boldsymbol{T}, \boldsymbol{T}^{T}, \boldsymbol{P}, \boldsymbol{A}^{+}, \boldsymbol{A}^{-}, \boldsymbol{A}^{S}, \boldsymbol{A}^{T}, \\ \boldsymbol{A}^{I}, \boldsymbol{P}^{In}, \boldsymbol{P}^{Out}, \boldsymbol{M}_0, \boldsymbol{M}_{max}, \boldsymbol{C}, \Delta >,$$

where:

$\boldsymbol{T} = \left\{\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_L\right\}$, - is a set of processing elements that correspond to non-timed transition nodes;

$\boldsymbol{T}^{T} = \left\{\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_{L^T}\right\}$, $\boldsymbol{T} \cup \boldsymbol{T}^{T} \neq \varnothing$ - is a set of processing elements that correspond to timed transition nodes;

$\boldsymbol{P} = \left\{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\right\}, \boldsymbol{P} \neq \varnothing$ - is a set of processing elements that correspond to place nodes;

$\boldsymbol{A}^{+}$ - is a set of increment connections of the number of tokens in place processing element $\boldsymbol{p_i}$;

$\boldsymbol{A}^{-}$ - is a set of decrement connections of the number of tokens in place processing element $\boldsymbol{p_i}$;

$\boldsymbol{A}^{S}$ - is a set of state connections that determine the enable firing condition of the transition $\boldsymbol{t_j}$ related to the marking in the input place $\boldsymbol{p_i}$;

$\boldsymbol{A}^{T}$ - is a set of test connections, which have the same function as the set of state connections, but the transition firing does not change the marking in the input place;

$\boldsymbol{A}^{I}$ - is a set of inhibitor connections, which provide an enabling function of the transition $\boldsymbol{t_j}$, when the place $\boldsymbol{p_i}$ stores no tokens. Inhibitor connection has the ability to test whether a place is empty. The firing does not change the marking in the input place;

$\boldsymbol{P}^{In} = \left\{\boldsymbol{P}^{In}_{i}, \ \boldsymbol{i} = \overline{1, \boldsymbol{N}^{In}}\right\}$, $\boldsymbol{P}^{In} \in \boldsymbol{P}$ - is a set of processing elements that correspond to place nodes $\boldsymbol{p_i}$ that specify the input signals of the system;

$\boldsymbol{P}^{Out} = \left\{\boldsymbol{P}^{Out}_{i}, \ \boldsymbol{i} = \overline{1, \boldsymbol{N}^{Out}}\right\}$, $\boldsymbol{P}^{Out} \in \boldsymbol{P}$ - is a set of processing elements that correspond to place nodes $\boldsymbol{p_i}$ that specify the output signals from the system;

$\boldsymbol{M}_0 = \{ \boldsymbol{M}^{P_1}_0, \boldsymbol{M}^{P_2}_0, \ldots, \boldsymbol{M}^{P_N}_0 \}$ - is the initial marking of the ***THPN***, defined as the initial number of tokens in each place;

$\boldsymbol{M}_{max} = \{ \boldsymbol{M}^{P_1}_{max}, \boldsymbol{M}^{P_2}_{max}, \ldots, \boldsymbol{M}^{P_N}_{max} \}$ - is the maximal marking of the ***THPN***, defined by the maximal number of tokens in each place;

$\boldsymbol{C}$ - is the synchronization variable;

Δ - is a set of time variables, which specify delays between transitions enabling and firing.

For each processing element analytical models and AHDL codes were developed.

a) The processing element place $(\boldsymbol{P})$

The processing element $\boldsymbol{P}$ stores the state value and performs the increment and decrement operations of the number of tokens. In an ordinary Petri Net all arcs are unity-weighted and the place is marked with a positive integer number of tokens. The increment operation occurs when one of the input transitions to the processing element $\boldsymbol{P}$ fires. The decrement operation occurs when one of the output transition fires. The number of tokens in $\boldsymbol{p_i}$ at the step ***k+1*** of data processing, denoted by $\boldsymbol{m}^{k+1}_{\boldsymbol{i}}$, is changed according to the following rules:

$$M_{k+1}^{p_i}=\begin{cases} M_k^{p_i}+1 \left| \sum\limits_{j=1}^{N(^*p_i)}(a^+{}_{ij})=1,\ \forall\ M_k^{p_i}<M_{\max}^{p_i}; \right. \\ M_k^{p_i}-1 \left| \sum\limits_{j=1}^{N(p_i^*)}(a^-{}_{ij})=1\ \forall\ M_k^{p_i}>0; \right. \\ M_k^{p_i} \left| \sum\limits_{j=1}^{N(^*p_i)}(a^+{}_{ij})=0\ \&\ \sum\limits_{j=1}^{N(p_i^*)}(a^-{}_{ij})=0; \right. \\ M_k^{p_i} \left| \sum\limits_{j=1}^{N(^*p_i)}(a^+{}_{ij})=1\ \&\ \sum\limits_{j=1}^{N(p_i^*)}(a^-{}_{ij})=1; \right. \end{cases} ,i=\overline{1,N} \quad (1)$$

where: $M_k^{p_i}$ is the number of tokens in p_i at the step k of data processing, $M_{k+1}^{p_i}$ is the number of tokens in p_i at the step $k+1$ of data processing, $N\left({}^*p_i\right)$ and $N\left(p_i^*\right)$ are the total number of increment and decrement arcs for the place p_i, $(M_{max}^{P_i}\ \forall\ i=\overline{1,N}) \in M_{max}$ represent the maximal number of tokens that can be stored in p_i.

The example of the AHDL code for the processing element P is shown below. The code describes the places $P=\{p_1,p_2,...,p_N\}$ with 4 inputs for incrementation of the number of tokens ***Pinc***, 4 inputs for decrementation of the number of tokens ***Pdec***, the maximal number of tokens in the place $cnt=8$ and output signal ***Pout***.

```
--include "p_obj_o";
include "lpm_counter";
   parameters(
       cnt=8, init=1
            );
subdesign p_obj_o(
       Pinc0, Pinc1, Pinc2, Pinc3 : INPUT = GND;
       Pdec0, Pdec1, Pdec2, Pdec3 : INPUT = GND;
       clk  : INPUT;
       Reset : INPUT;
       Pout : OUTPUT;
            )
variable
       inc, dec, res, full, empty :NODE;
       c : lpm_counter with (LPM_WIDTH=ceil(log2(cnt+1)),
                          LPM_SVALUE=init);
      begin
      inc = Pinc0 # Pinc1 # Pinc2 # Pinc3;
     dec = Pdec0 # Pdec1 # Pdec2 # Pdec3;
     res = Reset;
     c.clock=clk;
        if c.q[]==0 then empty=vcc;
          else    empty=gnd;
        end if;
        if c.q[]==cnt then    full=vcc;
        else        full=gnd;
        end if;
   c.cnt_en=(inc & !full & !dec) # (dec & !empty & !inc);
   c.sset=res;
   c.updown=inc;
   -- count[]=c.q[];
  Pout = !empty;
 end;
```

b) *The processing element non-timed transition* (***T***)

T prepares the data processing operation. After analyzing the global state $S^k=\left\{(M_i^k,P_i),\forall\, i=\overline{1,N}\right\}$ at the step k of data processing, the condition for passing the model from state S^k to state S^{k+1} is formed.

The behavior of the processing element T may be described as follows: if in each input place of a transition T there is a token, then the firing condition of T occurs. In this case tokens are removed from all input places and are placed into all output places. The behavior of non-timed transition is described by the following formula:

$$T_{out}=\left(\prod_{i=1}^{N(^*t_j)} a_{i,j}^S \wedge \prod_{i=1}^{N(^*t_j)} a_{i,j}^T \wedge \prod_{i=1}^{N(^*t_j)} \overline{a}_{i,j}^I\right) \quad (2)$$

where: $N\left({}^*t_j\right)$ is the number of input places for transition t_j.

The example of the AHDL code for the processing element non-timed transition T is shown below. The code describes the non-timed transitions $T=\{t_1,t_2,...,t_L\}$ with 4 state inputs ***Tin*** and output signal ***Tout***.

```
-- include "P_OBJ";
Subdesign T_OBJ
      (
      Tin0, Tin1, Tin2, Tin3 : INPUT = VCC;
      Clk  : INPUT;
      Set  : INPUT = GND;
      Reset: INPUT = GND;
      Tout : OUTPUT;
       )
variable
      T_DFF : JKFF;
      T : NODE;
    begin
      T_DFF.Clk = Clk;
      T_DFF.PRN = !Set;
      T_DFF.CLRN = !Reset;
      T = Tin0 & Tin1 & Tin2 & Tin3;
      T_DFF.J = T;
      T_DFF.K = !T;
      Tout = T_DFF.Q;
  end;
```

c) *The processing element timed transition* (T^T)

T^T prepares the data processing operation that will occur after a defined delay time Δ. After analyzing both the global state $S^k=\left\{(M_i^k,P_i),\forall\, i=\overline{1,N}\right\}$ and the delay time Δ, at the step k of data processing, the condition for passing the model from state S^k to state S^{k+1} is formed.

The behavior of timed transition is described by the following formula:

$$T_{out}^T=\left(\prod_{i=1}^{N(^*t_j)} a_{i,j}^S \wedge \prod_{i=1}^{N(^*t_j)} a_{i,j}^T \wedge \prod_{i=1}^{N(^*t_j)} \overline{a}_{i,j}^I\right)\wedge\Delta_j \quad (3)$$

where: $N\left({}^*t_j\right)$ is the number of input places for transition t_j and Δ_j the delay time variable for transition t_j.

The logic state of the delay time variable Δ_j is calculated according to expression (4).

$$\Delta_j = \begin{cases} 0 \;\middle|\; \boldsymbol{Delay}(\boldsymbol{t}_j) < \Delta(\boldsymbol{t}_j); \\ 1 \;\middle|\; \boldsymbol{Delay}(\boldsymbol{t}_j) = \Delta(\boldsymbol{t}_j); \end{cases} \quad \boldsymbol{j} = \overline{1, \boldsymbol{L}} \tag{4}$$

where $\boldsymbol{Delay}(\boldsymbol{t}_j)$ - is the current delay time processed for transition $\boldsymbol{t}_j$; $\Delta(\boldsymbol{t}_j)$ - is the defined delay time for transition $\boldsymbol{t}_j$.

According to expression (4), the enable logic signal for transition $\boldsymbol{t}_j$ will be equal to „0" if the processed delay time $\boldsymbol{Delay}(\boldsymbol{t}_j)$ is smaller than the defined delay time $\Delta(\boldsymbol{t}_j)$ and will be equal to „1" if it reaches the value $\Delta(\boldsymbol{t}_j)$.

The example of the AHDL code for the processing element timed transition $\boldsymbol{T}^T$ is shown below. The code describes the timed transitions $\boldsymbol{T}^T = \{\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_{L^T}\}$ with 4 state inputs ***Tin***, internal parameter $\boldsymbol{cnt} = 8$ (determines delay time for transition $\boldsymbol{T}^T$) and output signal ***Tout***.

```
--include "t_obj_t";
include "lpm_counter";
parameters (
        cnt=8, init=1
            );
subdesign T_OBJ_T
        (
        Tin0, Tin1, Tin2, Tin3 : INPUT = VCC;
        Clk  : INPUT;
        Set  : INPUT = GND;
        Reset: INPUT = GND;
        Tout : OUTPUT;
        )
   variable
        inc, dec, res :NODE;
        c : lpm_counter with (LPM_WIDTH=ceil(log2(cnt+1)),
                                LPM_SVALUE=init);
    begin
        inc = GND;
        dec = Tin0 # Tin1 # Tin2 # Tin3;
        res = Reset;
        c.clock=clk;
        if c.q[]==0   then      Tout=vcc;
        else                    Tout=gnd;
        end if;
           c.cnt_en=(dec & !inc);
    c.sset=res;
    c.updown=dec;
   Tout = vcc;
  end;
```

VI. Conclisions

This paper presents the synthesis method for delay time evaluation systems in printed circuit boards based on Petri Nets. For behavioral specification and modeling of a control system Timed Synchronous Petri net is used. The application of TSPN allows capturing the causality relations, concurrency of actions and conflicting conditions in the digital system already at the modeling stage. Timed Hard Petri Nets were proposed to perform the conversion of the model into logic circuit. The direct mapping of the THPN model into FPGA circuit ensures that the behavioral properties and time constraints will be respected. Moreover, direct mapping avoids algorithmic complexity inherent in logic synthesis methods based on state encoding and substantially reduces the design time and cost.

In order to validate the proposed method an example of delay time evaluation system in printed circuit boards, which consists of M homogeneous channels running in parallel has been presented.

VII. Acknowledgment

Some results included in this paper are related with the research works associated with the Ph.D. thesis activity plan "Parallel Computing Architectures for the Real-Time Acquisition and Processing of Multidimensional Signals".

Contribution of authors:
First author – 40%
First coauthor – 40%
Second coauthor – 10%
Third coauthor – 10%

Received on November 1, 2017 Editorial
Approval on November 15, 2017

References

[1] S. Gebus, S. Lorillard & E. Juuso. Defect Localization on a PCB with Functional Testing. *Report A No 20*, May 2002. 44 p., ISBN 951-42-6731-1.

[2] Ch. Lotz, P. Collins, D. Wiatrowski. "Finctional Board Test – Coverage Analysis." *In European Board Test Workshop, Southampton (UK),* May 24 & 26, 2006. (Accessed: http://www.aster-technologies.com/pub/en-ebtw06%20paper_1.2.pdf).

[3] N. Paunovic, J. Kovacevic, I. Resetar. "A Methodology for Testing Complex Professional Electronic Systems." *Serbian Journal of Electrical Engineering.* Vol. 9, No. 1, February 2012, pp. 71-80. DOI: 10.2298/SJEE1201071P.

[4] V. Ababii, Viorica Sudacevschi, D. Calugari. „Synthesis of Parallel Data Acquisition System for Analyzing of Multidimensional Signals." *In Sciences of Europe (Praha, Chech Republic),* Vol 1, No 17(17), 2017, pp. 75-79, ISSN 3162-2364.

[5] J.L. Peterson, "Petri Net Theory and the Modeling of Systems", *Prentice-Hall*, 1981.

[6] E. Guțuleac, C. Boșneaga, A. Railean, "VPNP-Software tool for modeling and performance evaluation using generalized stochastic Petri nets", *In: Proceedings of the 6-th International Conference on DAS-2002,* 23-25 May 2002, Suceava, România, pp. 243-248, ISBN 973-98670-9-X.

[7] V. Ababii, Viorica Sudacevschi, M. Podubnii, Irina Cojuhari. "Real-time reconfiguration of distributed control system based on hard Petri nets." *International Conference on DEVELOPMENT AND APPLICATION SYSTEMS 12th Edition, May 15-17, 2014, Suceava, ROMANIA,* pp. 21-24, ISSN 1844-5039. DOI: 10.1109/DAAS.2014.6842421. In SCOPUS (http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6842421&sortType%3Dasc_p_Sequence%26filter%3DAND%28p_IS_Number%3A6842413%29).

[8] Viorica Sudacevschi, V. Ababii, M. Podubnii. "A Synthesis Method for Reconfigurable Embedded Processors." *In the 13th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS,* Suceava, Romania, May 19-21, 2016, IEEE Catalog Number: CFP1665Y-DVD, pp. 184-188, ISBN: 978-1-5090-1992-2.

[9] Viorica Sudacevschi, V. Ababii. „Modelling and Synthesis of Real-Time Control Systems Based on Hardware Timed Petri Nets." *In Buletinul Institutului Politehnic Din Iaşi, Publicat de Universitatea Tehnică „Gheorghe Asachi" din Iaşi, Secția „Electrotehnică. Energetică. Electronică"*, Tomul LIX (LXIII), Fasc. 4, 2013, pp. 161-172.