G.A. Polyakov[1], V.V. Lysykh[2]

# A formal method of functional SNS- synthesis of problem-oriented parallel-pipelined devices

ABSTRACT. THE MAIN QUESTIONS OF FORMAL METHOD OF FUNCTIONAL SYNTHESIS OF PROBLEM-ORIENTED PARALLEL-PIPELINED ON THE BASIS OF STRUCTURE OF SEMANTIC - NUMERIC SPECIFICATION (SNS) ARE CONSIDERED. FORMULATION OF THE PROBLEM IS FORMULATED, DEFINES THE SEMANTICS OF THE MAIN STAGES. THE WORK OF THE METHOD IS ILLUSTRATED USING AN EXAMPLE.

*Keywords:* Electronic Design Automation, EDA, functional design , task C – Graph, concurrent computing, pipelining computing, structure of semantic - numeric specification (SNS), time parameterized model / process, time parallel graph – scheme (TPGS), verification, performance indicators, visualization.

## Introduction

The central problem of modern computer technologies is the increase of the efficiency of parallel computer systems (CS). Promising the way of solving the problem is the designing a problem-oriented multi – parallel digital devices and their hardware implementation based on the custom ASIC (ASIC), and /or VLSI programmable logic (FPGA).

Analysis of existing systems EDA shows that the conception of their design is to perform the most complex, formalized, creative design stages by person. It determines the quality of the hardware complexity, time and the cost of designing. However, there were problems of EDA, without satisfactory of solution: the gap between the complexity of VLSI circuits, which can produce the electronics industry, and the limiting complexity of the projects supported by the well-known EDA (problem of System-on-Chip, SOC); inability of EDA to short time of hardware design (problem Time-to-Market, T2M).

This paper describes a SNS - method of formal functional synthesis in parallel - pipelined digital devices with rigid logic operation. Used methods of concurrency - a method of combining independent operations and pipelining method. The basis of the formalism of the synthesis is the application of a new mathematical apparatus - Algebra of Structures Semantic - Numeric Specifications, SNS (instead of language HDL,

[1] NRU “Belgorod State University”, Russia, e-mail: tda_ua@pochtamt.ru
[2] NRU “Belgorod State University”, Russia, e-mail: lysykh@bsu.edu.ru

Verylog, VHDL, System C). Unlike traditional systems EDA, SNS - method provides the support for the following new features:

• intelligent - a formal resolution of complex design problems that are considered now the prerogative of specialists - the developers due to their creative nature;

• formal and automatic nature of the design - to perform all phases of functional design;

• adaptability - automatic maintenance of high performance functional synthesis and design results in a change of tasks, applications and customer requirements to the projected objects;

• flexibility - the ability to design a functional problem - oriented parallel - pipelined digital devices of different classes for different application domains with different requirements and restrictions[1-3].

## 1. Statement of the problem and steps of formal SNS – method of synthesis of functional parallel – pipelined digital devices

### 1.1. Initial data:

• tasks / algorithms presented by the source code in C / C + + - programs;

• the set of the methods of parallel processing - the combination of independent operations, pipelining data;

• the set and characteristics of the element database / reference library of functional modules;

• the system of requirements and constraints (time of solving the problem, time of clock, performance, complexity / cost).

### 1.2. Output data:

• the structure of semantic - numerical specification of functional scheme of device;

• the graphical specification of functional scheme of device;

• time parameterized parallel - pipelined solutions model of the device;

• The values of performance in parallel - pipeline device (while solving the problem, the value of the interval clock / clock frequency, the number of functional units of different types / the total gate complexity / the cost)

### 1.3. The main stages of the method of synthesis of functional device

A. Structures synthesis and semantic numerical specifications (SNS) as well as the graphic specifications (C - graph) C - program.
B. Synthesis of time - parameterized model C - program that uses a combination of independent operations.
C. Synthesis of structures SNSM and graphical specification (TPGS) time parallel-pipelined model C - program.
D. Estimation of the efficiency of the parallel-pipeline model of C programs (the time clock).
E. Checking requirements: "no" - into F, "yes" - into G.
F. Change the settings of the resource of function modules (FM): the number of FM, FM parameters (delay time, clock frequency), into C.
G. Synthesis of structures SNSF and graphical specification of functional scheme of parallel -pipelined digital device.
H. Displaying the results of a formal functional SNS - synthesis: SNS device specification, functional scheme of the device, the time model of the device, the device efficiency.

### 1.4. Visualization of some stages of SNS - method of synthesis of parallel - pipelined digital devices

Consider the problem of a C - program which is Figure 2.

The requirement to measure of the duration of the pipeline TT $\leq$ TTd = 44 ns, the restrictions on the complexity of the device are absent.

```
#include <stdio.h>
void main(void)
{
    inta,b,  k,z,p,s;
    scanf("%d %d %d %d",&a,&b);
if(a == b)
    {
     k = a % 2;
     z = a * b;
printf("%4d\n",k);
    printf("%4d\n",z);
    }
else
    {
     p = a | b;
     s = b / a ;
printf("%4d\n",p);
    printf("%4d\n",s);
    }
}
```

Figure 1. Initial C – program

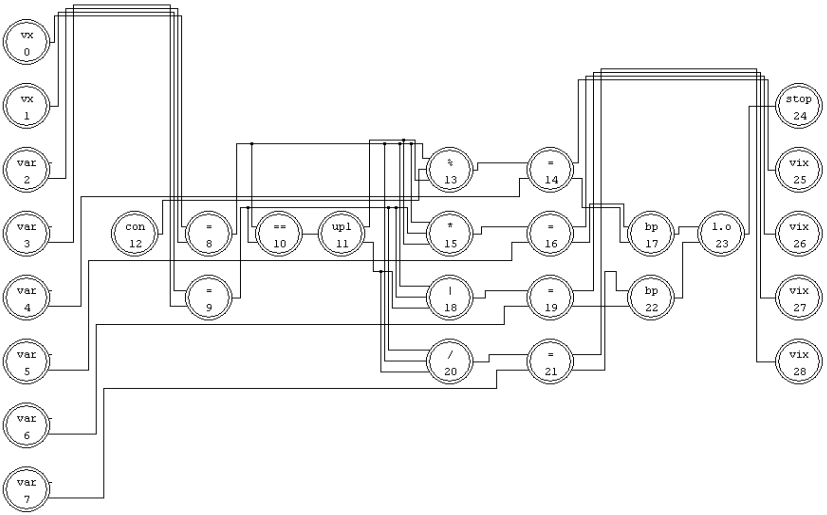


Figure 2. Graphical specification (C - Graph) of C - program (phase A)

Table 1. BF structure of semantic - numeric specification of operators in C – programs

| N | MET | TYP | NSJ | SJD | BJ | NWJ | WJD | MP1 | MP2 | VH | VIH | REZ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 58 | -1 | 0 | 0 | 0 | 1 | e | 0 | 0 | 1 | a_in |
| 1 | 0 | 58 | -1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | b_in |
| 2 | 0 | 47 | -1 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 2 | a |
| 3 | 0 | 47 | -1 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 2 | b |
| 4 | 0 | 47 | -1 | 0 | 0 | 4 | 1 | 0 | 0 | 0 | 2 | k |
| 5 | 0 | 47 | -1 | 0 | 0 | 5 | 1 | 0 | 0 | 0 | 2 | z |
| 6 | 0 | 47 | -1 | 0 | 0 | 6 | 1 | 0 | 0 | 0 | 2 | P |
| 7 | 0 | 47 | -1 | 0 | 0 | 7 | 1 | 0 | 0 | 0 | 2 | s |
| 8 | 0 | 12 | 0 | 2 | 0 | 8 | 5 | 0 | 0 | 2 | 1 | = |
| 9 | 0 | 12 | 2 | 2 | 0 | 3 | 4 | 0 | 0 | 2 | 1 | = |
| 10 | 0 | 23 | 4 | 2 | 0 | 17 | 1 | 0 | 0 | 2 | 1 | == |
| 11 | 0 | 51 | 6 | 1 | 0 | 18 | 4 | 1 | 2 | 1 | 2 | upl |
| 12 | 0 | 57 | -1 | 0 | 1 | 22 | 1 | 0 | 0 | 0 | 1 | C2 |
| 13 | 1 | 5 | 7 | 3 | 1 | 23 | 1 | 0 | 0 | 3 | 1 | % |
| 14 | 0 | 12 | 10 | 2 | 1 | 24 | 2 | 0 | 0 | 2 | 2 | = |
| 15 | 0 | 3 | 12 | 3 | 1 | 26 | 1 | 0 | 0 | 3 | 1 | * |
| 16 | 0 | 12 | 15 | 2 | 1 | 27 | 2 | 0 | 0 | 2 | 2 | = |
| 17 | 0 | 50 | 17 | 2 | 1 | 29 | 1 | 3 | 0 | 2 | 1 | bp |
| 18 | 2 | 34 | 19 | 3 | 2 | 30 | 1 | 0 | 0 | 3 | 1 | \| |
| 19 | 0 | 12 | 22 | 2 | 2 | 31 | 2 | 0 | 0 | 2 | 2 | = |
| 20 | 0 | 4 | 24 | 3 | 2 | 33 | 1 | 0 | 0 | 3 | 1 | / |
| 21 | 0 | 12 | 27 | 2 | 2 | 34 | 2 | 0 | 0 | 2 | 2 | = |
| 22 | 0 | 50 | 29 | 2 | 2 | 36 | 1 | 3 | 0 | 2 | 1 | bp |
| 23 | 3 | 54 | 31 | 2 | 3 | 37 | 1 | 0 | 0 | 2 | 1 | 1.0 |
| 24 | 0 | 49 | 33 | 1 | 3 | -1 | 0 | 0 | 0 | 1 | 0 | stop |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 0 | 48 | 34 | 1 | 3 | -1 | 0 | 0 | 0 | 1 | 0 | k_out |
| 26 | 0 | 48 | 35 | 1 | 3 | -1 | 0 | 0 | 0 | 1 | 0 | z_out |
| 27 | 0 | 48 | 36 | 1 | 3 | -1 | 0 | 0 | 0 | 1 | 0 | p_out |
| 28 | 0 | 48 | 37 | 1 | 3 | -1 | 0 | 0 | 0 | 1 | 0 | s out |

BF structure of operators (Table 1) contains in its structure the following data sets

BF = (N MET TYP NSJ SJD BJ NWJ WJD MP1 MP2 VH VIH RES), where N - array of the "operators" C programs (the original name / output data of the login data input and output, user IDs / functions), or "tops" - when the design graph C programs); MET - array of labels operators ; TYP - an array of types of operators C programs; NSJ(Pj) - an array of pointers to the beginning of a chain of numbers i operators Pi, is an operand for Pj (conjugate set); SJD(Pj) - an array of amounts sjd(Pj) "paired" to Pj operators Pi C programs; BJ - array of the natural parts of the C - program (non-branching fragments operators); NWJ(Pj)- an array of pointers to the beginning of the chain nwj numbers i "external" operators Pi/vertex C - graph, using the result of the operator Pj; WJD(Pj) - an array of amounts wjd "external" to the Pj operators Pi C programs; MP1, MP2-label operators in C - programs; VH and VIH - the number of inputs (operands) and the number of outputs (output) of the various operators in C - programs ; RES - an array of data names and instructions / functions.

Table 2. CF structure of semantic - numeric specification of relation operators in C – programs (phase A)

| **NN** | **JSD** | **SPJD** | **SNWIH** | **SNWHO** | **TSS** | **JWD** | **WPJD** | **WNWHO** | **WNWIH** | **TVS** |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | -1 | 8 | 0 | 0 | 0 |
| 1 | -1 | 2 | 1 | 1 | 2 | -1 | 9 | 0 | 0 | 0 |
| 2 | 3 | 1 | 0 | 0 | 0 | -1 | 8 | 1 | 1 | 2 |
| 3 | -1 | 3 | 1 | 1 | 2 | -1 | 9 | 1 | 1 | 2 |
| 4 | 5 | 8 | 0 | 0 | 0 | -1 | 14 | 1 | 1 | 2 |
| 5 | -1 | 9 | 0 | 1 | 0 | -1 | 16 | 1 | 1 | 2 |
| 6 | -1 | 10 | 0 | 0 | 0 | -1 | 19 | 1 | 1 | 2 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 8 | 8 | 0 | 0 | 0 | -1 | 21 | 1 | 1 | 2 |
| 8 | 9 | 12 | 0 | 1 | 0 | 9 | 10 | 0 | 0 | 0 |
| 9 | -1 | 11 | 0 | 2 | 1 | 10 | 13 | 0 | 0 | 0 |
| 10 | 11 | 4 | 1 | 1 | 2 | 11 | 15 | 0 | 0 | 0 |
| 11 | -1 | 13 | 0 | 0 | 0 | 12 | 18 | 0 | 0 | 0 |
| 12 | 13 | 8 | 0 | 0 | 0 | -1 | 20 | 1 | 0 | 0 |
| 13 | 14 | 9 | 0 | 1 | 0 | 14 | 10 | 1 | 0 | 0 |
| 14 | -1 | 11 | 0 | 2 | 1 | 15 | 15 | 1 | 0 | 0 |
| 15 | 16 | 5 | 1 | 1 | 2 | 16 | 18 | 1 | 0 | 0 |
| 16 | -1 | 15 | 0 | 0 | 0 | -1 | 20 | 0 | 0 | 0 |
| 17 | 18 | 16 | 1 | 0 | 1 | -1 | 11 | 0 | 0 | 0 |
| 18 | -1 | 14 | 1 | 1 | 1 | 19 | 13 | 2 | 0 | 1 |
| 19 | 20 | 8 | 0 | 0 | 0 | 20 | 15 | 2 | 0 | 1 |
| 20 | 21 | 9 | 0 | 1 | 0 | 21 | 18 | 2 | 1 | 1 |
| 21 | -1 | 11 | 1 | 2 | 1 | -1 | 20 | 2 | 1 | 1 |
| 22 | 23 | 6 | 1 | 1 | 2 | -1 | 13 | 1 | 0 | 0 |
| 23 | -1 | 18 | 0 | 0 | 0 | -1 | 14 | 0 | 0 | 0 |
| 24 | 25 | 9 | 0 | 0 | 0 | 25 | 17 | 1 | 1 | 1 |
| 25 | 26 | 8 | 0 | 1 | 0 | -1 | 25 | 0 | 0 | 0 |
| 26 | -1 | 11 | 1 | 2 | 1 | -1 | 16 | 0 | 0 | 0 |
| 27 | 28 | 7 | 1 | 1 | 2 | 28 | 17 | 0 | 1 | 1 |
| 28 | -1 | 20 | 0 | 0 | 0 | -1 | 26 | 0 | 0 | 0 |
| 29 | 30 | 21 | 1 | 0 | 1 | -1 | 23 | 0 | 0 | 1 |
| 30 | -1 | 19 | 1 | 1 | 1 | -1 | 19 | 0 | 0 | 0 |
| 31 | 32 | 22 | 0 | 1 | 1 | 32 | 22 | 1 | 1 | 1 |
| 32 | -1 | 17 | 0 | 0 | 1 | -1 | 27 | 0 | 0 | 0 |
| 33 | -1 | 23 | 0 | 0 | 1 | -1 | 21 | 0 | 0 | 0 |
| 34 | -1 | 14 | 0 | 0 | 0 | 35 | 22 | 0 | 1 | 1 |
| 35 | -1 | 16 | 0 | 0 | 0 | -1 | 28 | 0 | 0 | 0 |
| 36 | -1 | 19 | 0 | 0 | 0 | -1 | 23 | 1 | 0 | 1 |
| 37 | -1 | 21 | 0 | 0 | 0 | -1 | 24 | 0 | 0 | 1 |

The structure of the CF relations operators (Table 2) contains in its structure the following data sets

CF = (NN JSD SPJD SNWIH SNWHO TSS JWD WPJD WNWHO WNWIH TVS), where NN - the line of number in the structure of CF of the relation of operators; SPJD(Pj) - the set of numbers of operators Pi, which are operands for Pj (conjugate set); JSD - a pointer to the continuation of the chain of numbers i operators Pi C - programs / vertices of C - Graph included a conjugate set SPJD(j); SNWIH(Pi) - output number operators

Pi, conjugate to Pj; SNWHO(Pj)- input number of "external" to the operators Pi / vertices Pj; WPJD(Pj) – “external" set of numbers i operators Pi / vertex C - graph, using the result of the operators Pj; JWD-pointer to the continuation of the chain of numbers i operators Pi, which are “external” to Pj; WNWHO(Pi) - number of the input operators Pi, “external” to Pj; WNWIH(Pj) - output number operators Pj; TSS, TVS - «types" associated and external relations operators in C - programs / vertices C - Graphs.

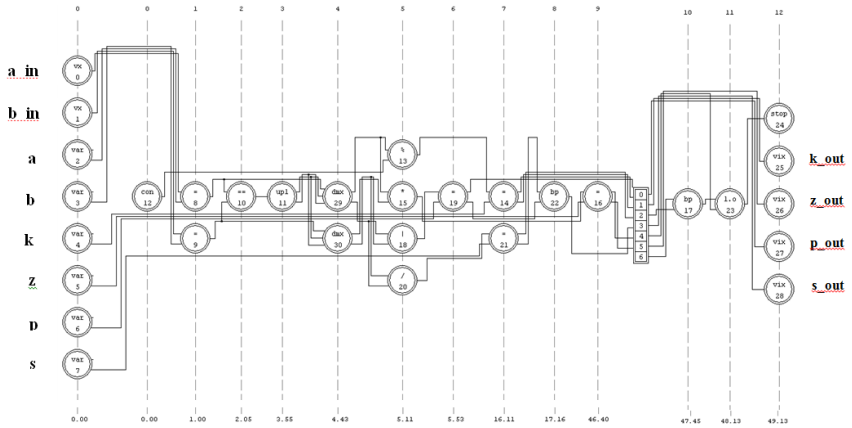


Figure3. Temporary parallel model C - programs with a combination of independent operations (Phase B)

Graphical specification of formal functional synthesis of parallel-pipeline device is presented in Figure 4. The device provides a clock value TT = 42.34 ns <44 ns and has two functional parts:

processing unit (function modules with the numbers 8 ... 16, 18 ... 21, 29.30, the input interface circuit are input «vx0» (a_in) and input «vx1» (b_in), the output interface is the output of «vix25» (k_out) , «vix26» (z_out), «vix27» (p_out) and «vix28» (s_out), exit «stop» 24;

control unit, which includes the operator P41 of input of clock signal "CLK" defining the moments of the transfer of data values between the pipeline fragments by synchronizing(memory) and the necessary synchronization of communication.

The report explains the semantics of stages A, ..., H and their formalization, the results of the formal synthesis of structures BFM, CFM SNS temporary parallel model C - programs with a combination of

independent operations (phase B), and structures BFPK, CFPK of the functional diagram of the synthesized device and temporal models of his work, as well as estimates values of performance indicators (running time, the value of the clock interval / clock speed, complexity in the number of functional units of each type.

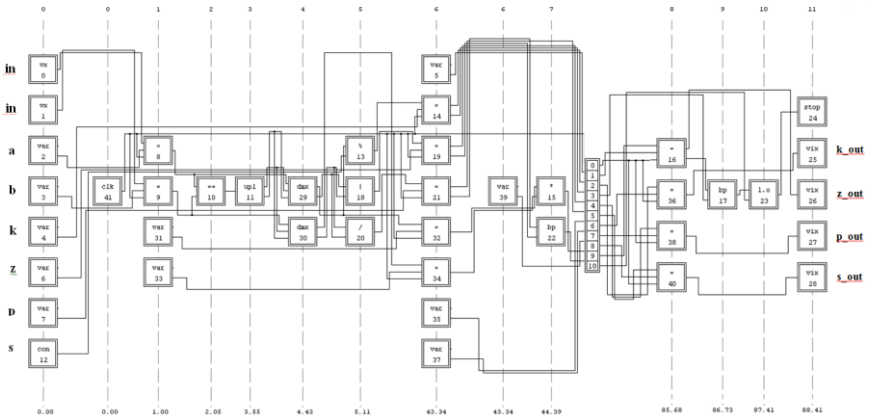


Figure 4. The results of formal SNS - synthesis of functional circuits in parallel - the pipeline device, combined with the timing chart of operation (from the beginning stages, G)

## References


[1] G.A. Polykov, The adaptive self-organizing systems with multiparallel data processing - the strategy of development of digital computer technology in the XXI century, *Applied electronics* ***1****(2002)*, 57-69.

[2] G.A. Polykov, The hard-and-soft Automatic Design of Self-Organizing Adaptive Systems, *Electronics and Computer Science* ***3****(2003)*, 149-155.

[3] G.A. Polykov, *Synthesis and analysis of parallel processes in the time-parametrized adaptive computing systems*, V.N. Karazin Kharkiv National University, Kharkiv, 2012.