



УДК 680.3

ФУНКЦИОНАЛЬНЫЙ СИНТЕЗ ПАРАЛЛЕЛЬНЫХ НЕПЕРЕСТРАИВАЕМЫХ СПЕЦПРОЦЕССОРОВ С ИСПОЛЬЗОВАНИЕМ АППАРАТА СТРУКТУР СЕМАНТИКО-ЧИСЛОВОЙ СПЕЦИФИКАЦИИ

Г.А. ПОЛЯКОВ
В.В. ЛЫСЫХ
В.В. ТОЛСТОЛУЖСКАЯ

*Белгородский
государственный
национальный
исследовательский
университет*

*e-mail:
lysykh@bsu.edu.ru*

В статье представлен обобщенный алгоритм формального функционального/RTL синтеза вычислительных подсистем параллельных спецпроцессоров с жесткой логикой функционирования. Определено содержание основных этапов. Семантика этапов синтеза и их результаты иллюстрируются с помощью конкретного примера.

Ключевые слова: система автоматизированного проектирования цифровых устройств (САПР), RTL (Register Transfer Level), Си – граф задачи, структуры семантико-числовой спецификации (СЧС), времяпараметризованная (временная) параллельная модель Си – программы, временная параллельная граф – схема (ВПГС), спецпроцессор с неперестраиваемой архитектурой, формальный функциональный синтез параллельных аппаратных средств.

Прогресс в развитии общества неразрывно связан с необходимостью решения во всё более короткие сроки всё более сложных задач науки, техники, телекоммуникаций, управления, бизнеса, энергетики, медицины, проектирования и т.д.

Поэтому, для решения таких задач широко применяются различные классы параллельных вычислительных систем. Несмотря на то, что эти системы успешно применяются, в настоящее время наметился большой разрыв между возросшими запросами практики, с одной стороны, и отсутствием возможности удовлетворения решения этих запросов с помощью параллельных вычислительных систем (включая суперкомпьютеры, грид-системы и кластеры, а также существующие автоматизированные системы проектирования параллельного аппаратно-программного обеспечения (САПР)), с другой.

Одной из центральных проблем современной вычислительной техники является повышение эффективности многопроцессорных ЭВМ и суперЭВМ [1,2,3]. Основными путями решения этой проблемы являются:

а) повышение эффективности параллельного программного обеспечения за счет совершенствования систем автоматизированного параллельного программирования (САПП);

б) повышение эффективности параллельных аппаратных средств на основе совершенствования систем автоматизированного проектирования цифровых устройств САПР/EDA (Elektronik Design Automation).

Анализ известных систем автоматизированного проектирования электронных цифровых устройств позволяет сделать следующие выводы [1,17,18,21]:

- основой концепции построения известных САПР/EDA является субъективное выполнение человеком вручную наиболее сложных, неформализованных, творческих этапов проектирования, определяющих качество цифровых устройств, допустимую сложность проектируемых объектов, а также сроки и стоимость проектирования – разработки функциональной (RTL) схемы устройства/системы и создание ее спецификации с помощью языков описания аппаратных средств (VHDL, Verilog, SystemC);

- известные САПР не обеспечивают возможности использования всех известных методов параллельной обработки данных с автоматической оптимизацией их состава на основе учета особенностей различных задач и конкретных требований/ограничений заказчиков;

- известные САПР не обеспечивают возможность проектирования перестраиваемых параллельных спецпроцессоров и систем, способных динамически адаптироваться к изменению состава решаемых задач, их особенностям и изменениям системы требований и ограничений;

- известные САПР не имеют средств, способных обеспечить автоматическое проектирование аппаратных средств, гарантированно обеспечивающих выполнение заданных показателей эффективности: времени выполнения, тактовой частоты, надежности, сложности/стоимости;

- использование при проектировании текстового и графического форматов спецификации объектов проектирования существенно ограничивает возможности повышения быстродействия и эффективности САПР.



• значительная роль ручного труда специалиста-проектировщика является причиной неспособности САПР существенно сократить сроки проектирования аппаратных средств (проблема *Time-to-Market*), в первую очередь наиболее сложных объектов проектирования – процессоров (CPU) и заказных специализированных устройств с «жесткой» структурой – ASIC при одновременном повышении качества результатов проектирования.

Целью статьи является описание обобщенного алгоритма формального Структурно – Семантико – Числового (ССЧ) синтеза вычислительных подсистем параллельных неперестраиваемых спецпроцессоров (НСП) на функциональном/RTL уровне проектирования цифровых аппаратных средств. В отличие от традиционных САПР метод формального функционального синтеза использует для спецификации всех этапов функционального проектирования новый математический аппарат – алгебру структур семантико-числовой спецификации (СЧС).

Исходными данными метода формального синтеза являются:

- Си-программа задачи, для аппаратного решения которой проектируется устройство/спецпроцессор;
- используемый метод параллельной обработки – совмещение независимых операций;
- база данных функциональных модулей и время выполнения ими операций/функций;
- требования/ограничения (заданное время решения задачи или ограничение на количество оборудования/стоимость);
- используемый математический аппарат – алгебра структур семантико-числовой спецификации [19, 20, 21].

Выходные данные должны быть представлены:

- структурами семантико-числовой спецификации СЧС результатов выполнения основных этапов формального синтеза вычислительной подсистемы параллельного спецпроцессора.
- параллельной временной моделью Си – программы исходной задачи;
- функциональной/RTL схемой вычислительной подсистемы параллельного спецпроцессора.
- Основные этапы обобщенного алгоритма формального функционального ССЧ – синтеза (*SSN, Structure Semantic Numerical Synthesis*) параллельных спецпроцессоров с совмещением независимых операций/ функций задач и жесткой логикой функционирования представляет рис. 1.

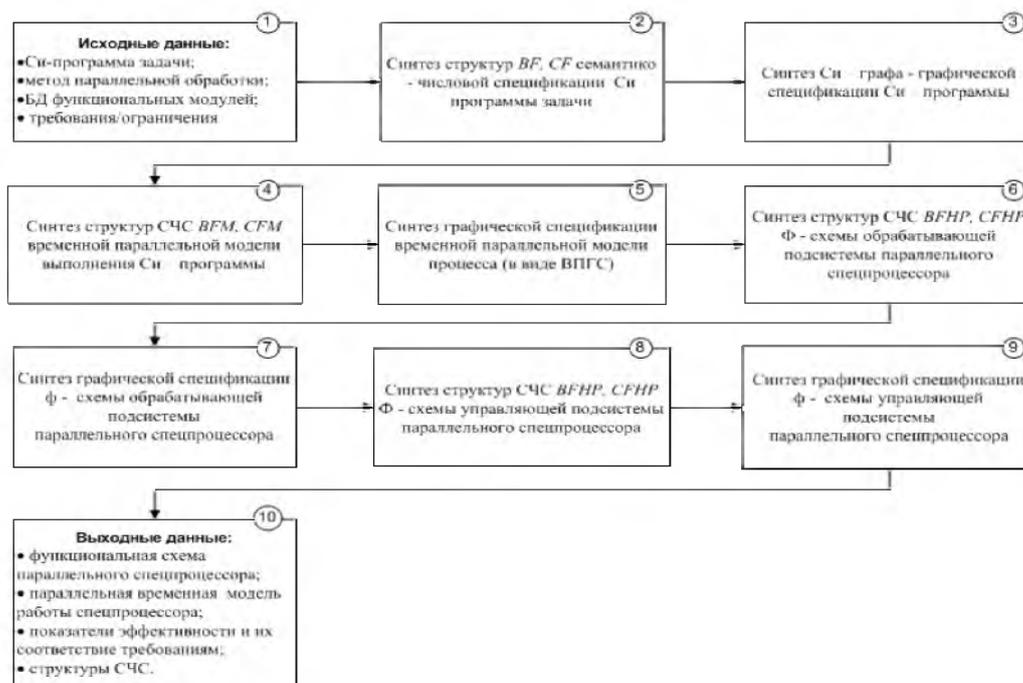


Рис. 1. Обобщенный алгоритм функционального/RTL синтеза вычислительных подсистем параллельных неперестраиваемых спецпроцессоров на основе аппарата структур семантико – числовой спецификации

При описании содержания шагов алгоритма синтеза будем использовать конкретную задачу, Си-программа которой представлена на рис. 2.



```

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

```

Рис. 2. Си-программа задачи

На первом этапе выполняется синтез структур семантико - числовой спецификации (СЧС), представляющих исходную Си программу задачи. При выполнении этапа формируются базовые структуры СЧС: структура BF СЧС состава операторов и структура CF СЧС связей операторов [22.].

Базовая структура BF операторов Си – программ имеет следующую организацию:

$$BF = (N, MET, TYP, NSJ, SJD, BJ, NWJ, WJD, MP1, MP2, VH, VIH, RES).$$

Базовая структура CF связей операторов Си – программы имеет следующую организацию:

$$CF = (N, JSD, SPJD, SWIH, SWHO, JWD, WPJD, WWHO, WWIH).$$

Структура RAZR_ID, содержащая дополнительную информацию о переменных и константах Си – программы, имеет следующую организацию:

$$RAZR_ID = (NN\ ID\ RAZR\ QRG\ INIT\ NO).$$

Структуры семантико-числовой спецификации содержат операторы-входы данных, операторы-имена переменных и констант, операторы-инструкции/функции, операторы вывода результатов, а также различные типы связей операторов: связи задания имен, связи по данным, управляющие связи, а также осведомительные связи признаков окончания естественных частей Си программы.

Переход от Си – программ к структурам BF и CF включает выполнение следующих действий: сквозную нумерацию операторов P_j (от $j=0$ до $j=n-1$, n – количество операторов); сквозную нумерацию входов - операндов конкретного оператора (от 0 до $VH_j - 1$, где VH_j – количество входов) для каждого оператора P_j и сквозную нумерацию выходов (от 0 до $VIH_j - 1$, где VIH_j – количество выходов оператора P_j); задание числового кодирования типов TYP_j операторов P_j ($j = 0, \dots, n - 1$); формирование для каждого оператора P_j ($j = 0, \dots, n - 1$) синтезируемой числовой спецификации множества $SPJD_j$ номеров «i» его операндов P_i (сопряженное множество для P_j) и задание его мощности SJD_j ; формирование для каждого оператора P_j множества $WPJD_j$ его внешних операторов (использующих результаты выполнения оператора P_j) и задание его мощности WJD_j ; формирование для каждого оператора P_j соответствующих меток $MP_j, MP1_j, MP2_j$

Массивы структуры RAZR_ID имеют следующую семантику: ID – массив идентификаторов переменных и констант; RAZR- массив значений разрядности переменных и констант; QRG – массив количеств ячеек памяти, необходимых для хранения переменных и констант; INIT- массив признаков необходимости их начальной инициализации; NO – номер оператора базовой структуры BF, представляющего переменную константу Си программы.

Результат формирования структуры BF состава операторов и структуры CF связей операторов СЧС исходной Си – программы (рис. 2) представляют рис. 3 и рис. 4.



N	MET	TPP	NSJ	SJD	BJ	NWJ	WJD	MP1	MP2	UH	UIH	RES
0	0	58	-1	0	0	0	1	0	0	0	1	a_in
1	0	58	-1	0	0	1	1	0	0	0	1	b_in
2	0	58	-1	0	0	2	1	0	0	0	1	c_in
3	0	58	-1	0	0	3	1	0	0	0	1	d_in
4	0	47	-1	0	0	4	1	0	0	0	2	a
5	0	47	-1	0	0	5	1	0	0	0	2	b
6	0	47	-1	0	0	6	1	0	0	0	2	c
7	0	47	-1	0	0	7	1	0	0	0	2	d
8	0	47	-1	0	0	8	1	0	0	0	2	r
9	0	47	-1	0	0	9	1	0	0	0	2	k
10	0	47	-1	0	0	10	1	0	0	0	2	z
11	0	47	-1	0	0	11	1	0	0	0	2	p
12	0	47	-1	0	0	12	1	0	0	0	2	s
13	0	12	0	2	0	13	4	0	0	2	1	=
14	0	12	2	2	0	17	3	0	0	2	1	=
15	0	12	4	2	0	20	2	0	0	2	1	=
16	0	12	6	2	0	22	1	0	0	2	1	=
17	0	23	8	2	0	23	1	0	0	2	1	==
18	0	51	10	1	0	24	5	1	2	1	2	up1
19	0	57	-1	0	1	29	2	0	0	0	1	C2_
20	1	5	11	3	1	31	1	0	0	3	1	%_
21	0	12	14	2	1	32	2	0	0	2	2	=
22	0	3	16	3	1	34	1	0	0	3	1	*
23	0	12	19	2	1	35	2	0	0	2	2	=
24	0	4	21	3	1	37	1	0	0	3	1	/
25	0	12	24	2	1	38	2	0	0	2	2	=
26	0	50	26	3	1	40	1	3	0	3	1	bp
27	2	3	29	3	2	41	1	0	0	3	1	*
28	0	12	32	2	2	42	2	0	0	2	2	=
29	0	4	34	3	2	44	1	0	0	3	1	/
30	0	12	37	2	2	45	2	0	0	2	2	=
31	0	50	39	2	2	47	1	3	0	2	1	bp
32	3	54	41	2	3	48	1	0	0	2	1	l.o
33	0	49	43	1	3	-1	0	0	0	1	0	stop
34	0	48	44	1	3	-1	0	0	0	1	0	k_out
35	0	48	45	1	3	-1	0	0	0	1	0	z_out
36	0	48	46	1	3	-1	0	0	0	1	0	r_out
37	0	48	47	1	3	-1	0	0	0	1	0	p_out
38	0	48	48	1	3	-1	0	0	0	1	0	f_out

Рис. 3. Структура BF СЧС состава операторов Си-программы

N	JSD	SPJD	SNWIH	SNUHO	JWD	MPJD	WNWHO	WNWIH
0	1	0	0	0	-1	13	0	0
1	-1	4	1	1	-1	14	0	0
2	3	1	0	0	-1	15	0	0
3	-1	5	1	1	-1	16	0	0
4	5	2	0	0	-1	13	1	1
5	-1	6	1	1	-1	14	1	1
6	7	3	0	0	-1	15	1	1
7	-1	7	1	1	-1	16	1	1
8	9	13	0	0	-1	25	1	1
9	-1	14	0	1	-1	21	1	1
10	-1	17	0	0	-1	23	1	1
11	12	13	0	0	-1	28	1	1
12	13	19	0	1	-1	30	1	1
13	-1	18	0	2	14	17	0	0
14	15	9	1	1	15	20	0	0
15	-1	20	0	0	16	22	0	0
16	17	13	0	0	-1	29	1	0
17	18	14	0	1	18	17	1	0
18	-1	18	0	2	19	22	1	0
19	20	10	1	1	-1	29	0	0
20	-1	22	0	0	21	24	0	0
21	22	15	0	0	-1	27	0	0
22	23	19	0	1	-1	27	1	0
23	-1	18	0	2	-1	18	0	0
24	25	8	1	1	25	20	2	0
25	-1	24	0	0	26	22	2	0
26	27	25	1	0	27	24	2	0
27	28	23	1	1	28	27	2	1
28	-1	21	1	2	-1	29	2	1
29	30	15	0	0	30	20	1	0
30	31	16	0	1	-1	24	1	0
31	-1	18	1	2	-1	21	0	0
32	33	11	1	1	33	26	2	1
33	-1	27	0	0	-1	34	0	0
34	35	14	0	0	-1	23	0	0
35	36	13	0	1	36	26	1	1
36	-1	18	1	2	-1	35	0	0
37	38	12	1	1	-1	25	0	0
38	-1	29	0	0	39	26	0	1
39	40	30	1	0	-1	36	0	0
40	-1	28	1	1	-1	32	0	0
41	42	31	0	1	-1	28	0	0
42	-1	26	0	0	43	31	1	1
43	-1	32	0	0	-1	37	0	0
44	-1	21	0	0	-1	30	0	0
45	-1	23	0	0	46	31	0	1
46	-1	25	0	0	-1	38	0	0
47	-1	28	0	0	-1	32	1	0
48	-1	30	0	0	-1	33	0	0

Рис. 4. Структура CF СЧС связей операторов Си-программы



Результат формирования структуры *RAZR_ID* СЧС исходной Си-программы (рисунок 2) представляет табл. 1.

Таблица 1

Структура СЧС *RAZR_ID* исходной Си – программы

NN	ID	RAZR	QRG	INIT	NO
0	a	16	1	2	4
1	b	16	1	2	5
2	c	16	1	2	6
3	d	16	1	2	7
4	r	16	1	1	8
5	k	16	1	1	9
6	z	16	1	1	10
7	p	16	1	1	11
8	s	16	1	1	12
9	c2	1	1	1	19

На втором этапе выполняется построение графической визуализации (в виде Си-графа) полученных структур *BF* и *CF* семантико-числовой спецификации исходной Си-программы. Результат выполнения этапа представляет рис. 5. Синтез Си графа является вспомогательным этапом, выполняемым в интересах представления визуальной информации, позволяющей проектировщику оценить корректность формального компьютерного выполнения первого этапа.

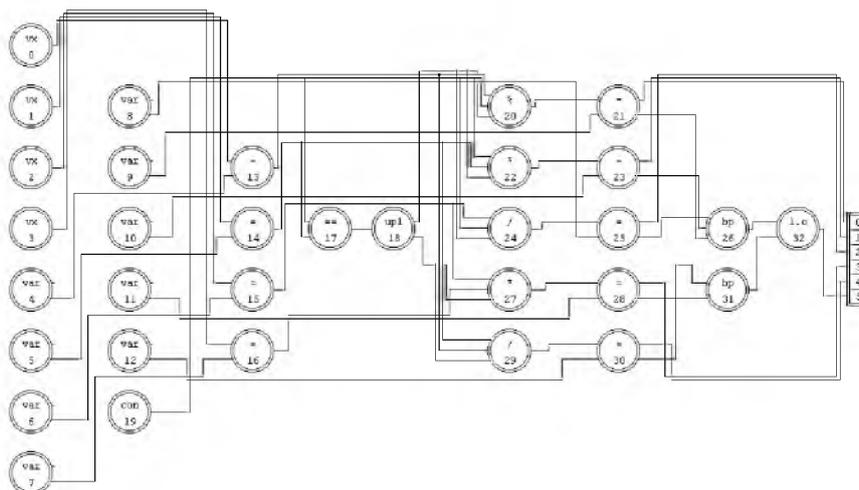


Рис.5. Графическая спецификация (Си-граф) исходной Си-программы

На третьем этапе выполняется синтез структур *BFM* и *CFM* СЧС, задающих состав и связи операторов временной параллельной модели и определяющих для каждого оператора временной модели СЧС Си - программы момент времени t_j^n начала выполнения, реализующий его функциональный модуль и, следовательно, определяющих псевдо временную модель процесса параллельного решения задачи с помощью спецпроцессора. Принятые при синтезе временной параллельной модели значения длительностей t_j^o (нс) выполнения модулями элементной базы операторов P_j различных типов (*typ*) показаны в табл. 2.

Таблица 2

Длительность выполнения модулями элементной базы различных типов (*typ*) операторов P_j параллельной модели задачи (нс)

<i>typ</i>	<i>vx</i>	<i>var</i>	=	==	<i>upl</i>	<i>dmx</i>	%,/	*	<i>bp</i>	<i>l.o</i>	&,	<i>vix</i>	<i>stop</i>
t^o (нс)	1.0	1.0	1.05	1.5	0.88	0.68	11.0	41.29	0.68	1.00	0.68	1.0	1.0

Выполнение этапа включает:

- расширение структур *BF* и *CF* СЧС путем введения коммутационных операторов (для рассматриваемого примера – операторов типа «*dmx*»), управляемых оператором типа «*upl*», и введения соответствующих связей по данным и по управлению;



• постановка в соответствие каждому оператору P_j структур СЧС (и Си-графа) значения времени выполнения t_j^0 путем выбора типа $typ(P_j)$ оператора P_j и длительности $t^0(typ)$ выполнения оператора данного типа, определяемой из табл.2;

• формирование множества $G_t(nj)$ операторов P_j^s с началом выполнения на nj -м ярусе в момент времени t , включающее:

а) формирование множества E_{nj}^c операторов P_j^s – кандидатов на начало выполнения в момент t в соответствии с соотношениями (1 - 3)

$$E_{nj}^c = R_{nj-1}^c \cup K_{nj}^c, \tag{1}$$

где R_{nj-1}^c – множество операторов P_j , реализация которых не была начата на $nj-1$ -м ярусе в связи с отсутствием необходимых свободных ф-модулей; K_{nj}^c – множество операторов, выполнение которых может быть начато на nj -м ярусе в связи с наличием всех необходимых для них данных (с учетом ранее выполненных операторов и информационно-управляющих связей между операторами $P_j \in P$);

$$R_{nj-1}^c = \begin{cases} 0 & \text{при } nj = 0, \\ E_{nj-1}^c \setminus G_{nj-1}^c & \text{при } nj > 0; \end{cases} \tag{2}$$

$$K_{nj}^c = \begin{cases} \bigcup_j P_j & \text{для } S(P_j)=0 \text{ (при } nj = 0), \\ \bigcup_j P_j & \text{при выполнении для } P_j \text{ следующих условий} \end{cases} \tag{3}$$

а) $P_j \in \bigcup_{P_\xi \in G_{nj-1}^1} W(P_\xi)$ при $S(P_j) \subseteq G_{nj-1}^1$;

б) $t_\xi^n + t_\xi^0 \leq t(nj)$ для всех $P_\xi \in S(P_j)$;

здесь

G_{nj-1}^1 – множество операторов P_j , реализация которых была начата на ярусах $nj = 1 \dots (nj - 1)$;

- б) оценку свободного в момент времени $t(nj)$ состава ф-модулей различных типов;
- с) выбор из множества операторов-кандидатов E_{nj}^c операторов P_j (в порядке уменьшения приоритетов);
- д) проверку наличия необходимого для выполнения оператора P_j свободного ресурса и закрепление оператора P_j за соответствующим функциональным модулем;
- е) расчет момента освобождения ресурса по завершении выполнения оператора P_j .

Отметим следующие особенности структур СЧС ВФМ и СФМ по сравнению со структурами ВФ, СФ:

- наличие операторов типа «dmx», управляемых оператором типа «upl», и их связей по данным и по управлению;
- наличие массива NT, элементы которого задают моменты t_j^n начала выполнения операторов P_j , рассчитанные по соотношениям (1 - 3);
- наличие массива NY, элементы которого задают принадлежность операторов P_j конкретному временному ярусу с номером nj параллельной модели.

На четвёртом этапе выполняется синтез временной параллельной граф-схемы (ВПГС), являющейся визуализацией временной максимально параллельной модели решения задачи. Графическую спецификацию временной параллельной модели представляет рис. 6.

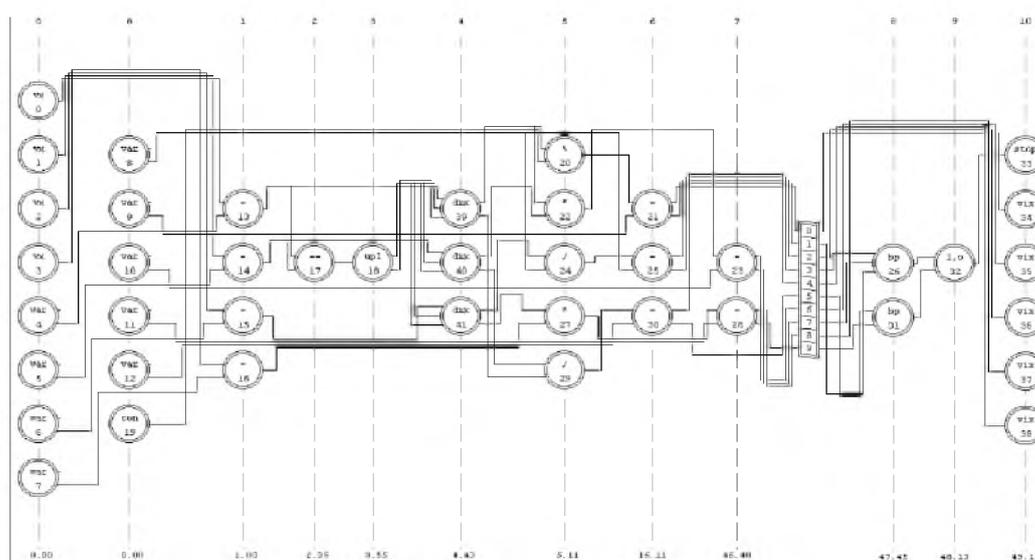


Рис. 6. Времяпараметризованная (временная) параллельная модель Си – программы задачи

На пятом этапе выполняется синтез структур СЧС ВФНР и СФНР (НР, Hard Processing) семантико-числовой спецификации функциональной схемы вычислительной подсистемы (НР) параллельного спецпроцессора, исходя из представляющих временную параллельную модель структур СЧС ВФМ и СФМ.

Выполнение этапа включает:

- формирование, исходя из структуры СЧС ВФМ, «базового» состава типов функциональных модулей, необходимых для реализации временной параллельной модели Си – программы (рис. 4);
- определение, исходя из структуры СЧС ВФМ, количества функциональных модулей каждого типа из «базового» состава модулей, необходимых для реализации временной параллельной модели Си – программы (рис. 6), и сопоставление каждому «базовому» модулю (для выполнения) «закрепленных» за ним операторов P_j ;
- синтез исходных структур ВФН и СФН СЧС функциональной схемы вычислительной подсистемы спецпроцессора, исходя из структур СЧС ВФМ и СФМ временной параллельной модели, путем интерпретации операторов P_j , реализуемых «индивидуальными» Φ – модулями, и их связей как соответствующих функциональных модулей вычислительной подсистемы и связей между ними;
- выделение подмножества «базовых» функциональных модулей, реализующих при выполнении временной параллельной модели Си – программы несколько операторов P_j соответствующего типа;
- введение для подмножества базовых Φ – модулей коммуникационных узлов типов «&» и «|»(образующих модули типа «mux»), обеспечивающих возможность ввода в выделенные Φ – модули различных операндов, соответствующих каждому из «закрепленных» за модулем операторов P_j ;
- расширение структур ВФН и СФН путем введения в синтезированные исходные структуры ВФН и СФН СЧС подмножества «базовых» функциональных модулей, подмножества коммуникационных узлов и связей между ними;
- завершение синтеза структур СЧС ВФНР и СФНР путем введения в структуры ВФН и СФН множества $T_c = \{tc_{j_in}\}$ входов tc_{j_in} временной синхронизации модулей памяти (тип «=>») и коммуникационных узлов (тип «&») и их связей с управляемыми Φ – модулями/узлами вычислительной подсистемы.

Основой формализации и автоматизации выполнения этапа 5 является использование библиотеки операций алгебры структур СЧС [21].

На шестом этапе выполняется визуализация, исходя из структур СЧС ВФНР и СФНР, функциональной/RTL схемы обрабатываемой подсистемы (НР) параллельного спецпроцессора, которую представляет рис. 7.

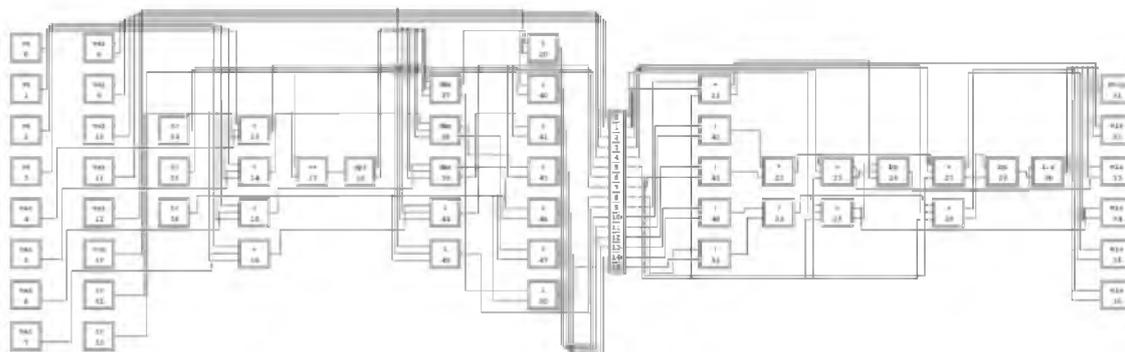


Рис. 7. Графическая спецификация синтезированной функциональной схемы обрабатывающей подсистемы параллельного спецпроцессора

На седьмом этапе выполняется синтез временной диаграммы работы вычислительной подсистемы (HP) и расчет моментов ввода в вычислительную подсистему сигналов временной синхронизации.

Выполнение этапа включает:

- а) формирование для подмножества узлов временной синхронизации (имеющих тип «&»), принадлежащих структуре ВФНР, подмножества РС операторов $P_j \in PC$ (типа «&»);
- б) «расширение» структур ВФМ и CFM СЧС временной параллельной модели (рис. 6) за счет:

- введения операторов $P_j \in PC$ в структуру ВФМ;
- формирования для каждого оператора $P_j \in PC$ сопряженного $S(P_j)$ и $W(P_j)$ внешнего множеств связей с использованием взаимно однозначных соответствий: «узел синхронизации \leftrightarrow оператор синхронизации» и «множества $S(P_j)$ и $W(P_j)$ оператора синхронизации P_j – это множества операторов, реализуемых сопряженными и внешними модулями/узлами конкретного узла временной синхронизации»;

с) введение в структуру CFM сопряженного $S(P_j)$ и $W(P_j)$ внешнего множеств связей оператора P_j ; получение «расширенных» структур структур ВФМН и CFМН СЧС;

д) синтез для структур ВФМН и CFМН в соответствии с соотношениями (1, 2, 3), «расширенной» структуры ТФМН СЧС, задающей временную диаграмму работы вычислительной подсистемы и определяющей (в качестве исходных данных для последующего синтеза устройства управления, Control Machine) множество моментов t_c ввода в вычислительную подсистему сигналов синхронизации t_{c_in} .

Синтезированные структуры СЧС ВФНР, CFНР, ТФМН и BAZ_COMP полностью определяют функциональную/RTL схему вычислительной подсистемы спецпроцессора, задают время параллельного решения задачи спецпроцессором, позволяют получить оценку сложности обрабатываемой подсистемы (по известной вентиляционной сложности ф-модулей/узлов используемой базы АК) и предоставляют исходные данные, необходимые для последующего формального синтеза управляющей подсистемы (Control Machine) спецпроцессора.

Разработанный обобщенный алгоритм структурно-семантико-числового метода синтеза и рассмотрение содержания его основных этапов показывает, что использование аппарата алгебры структур семантико-числовой спецификации обеспечивает, в отличие от известных САПР, полную формализацию функционального/RTL проектирования параллельных цифровых аппаратных средств. Метод формального ССЧ синтеза является основой создания инструментальных программных средств технологии автоматического синтеза мультипараллельных спецпроцессоров с жесткой и аппаратно-реконфигурируемой логикой работы.

Список литературы

1. Поляков Г.А. Основы построения и автоматического проектирования самоорганизующихся систем параллельной цифровой обработки информации и повышение эффективности комплексов радиолокационного вооружения ПВО / Г. А. Поляков ; [под общ. ред. проф. В. К. Стрельникова]. – Х.: ВИРТА ПВО, 1986. – 572 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
3. Поляков Г.А. Адаптивные самоорганизующиеся системы с мультипараллельной обработкой данных – стратегия развития цифровой вычислительной техники в XXI-м веке / Г. А. Поляков // Прикладная радиоэлектроника. – Х. : АН ПРЭ, 2002. – № 1. – С. 57–69.
4. Гаврилов М.А. Логическое проектирование дискретных автоматов (языки, методы, алгоритмы) /



- М. А. Гаврилов, В.В. Девятков, Е. И. Пурпырев. – М. Наука, 1977. – 352 с.
5. Горбатов В.А. Автоматизация проектирования сложных логических структур / [В.А. Горбатов, В.Ф. Демьянов, Г.Б. Кулиев и др.]; под ред. проф. В.А. Горбатова. – М.: Энергия, 1978. – 352 с.
 6. Корячко В.П. Теоретические основы САПР : учеб. для вузов / В.П. Корячко, В.М. Курейчик, И. П. Норенков. – М.: Энергоатомиздат, 1987. – 400 с.
 7. Угрюмов Е.П. Цифровая схемотехника. – СПб.: БХВ-Петербург, 2001. – 528 с.: ил.
 8. Норенков И.П. Основы автоматизированного проектирования : учеб. для вузов. – 2-е изд., перераб. и доп. / И.П. Норенков. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. – 336 с.
 9. Грушевицкий Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушевицкий, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ – Петербург, 2002. – 608 с.
 10. Кривуля Г.Ф. Новые информационные технологии проектирования цифровых систем / Г.Ф. Кривуля, В.И. Хаханов // Прикладная радиоэлектроника. Состояние и перспективы развития : первый Междунар. радиоэлектр. форум МРФ-2002 : тезисы док. Часть 2. – Х.: АН ПРЭ, ХНУРЭ. 2002. – С. 233–236.
 11. Максфилд К. Курс молодого бойца /К. Максфилд.–М. Издательский дом «Додэка – XXI», 2007. – 408 с.
 12. Разевиг В. Д. От OrCAD для DOS ACCEL EDA Windows / В.Д. Разевиг // САПР и Графика. – 1997. – № 6. – С. 7–14.
 13. Разевиг В.Д. Система P-CAD 8. 5 – 8. 7. Руководство пользователя / В.Д. Разевиг. – М.: СОЛОН-Р, 1999. – 329 с.
 14. Лобанов В.И. Алгоритмы разработчика цифровых устройств / В.И. Лобанов. – М.: Горячая линия – Телеком, 2001. – 192 с.
 15. Бибило П.Н. Синтез логических схем с использованием языка VHDL / П.Н. Бибило. – М.: СОЛОН-Р, 2002. – 384 с.
 16. Разевиг В.Д. SystemView-средство системного проектирования радиоэлектронных устройств / В.Д. Разевиг, Г.В. Лаврентьев, И.Л. Златин; под ред. В.Д. Разевига. – М. Горячая линия – Телеком, 2002. – 352 с.
 17. Умрихин Ю.Д. Автоматизация проектирования сложных цифровых систем обработки информации и управления реального времени / Ю.Д. Умрихин, Г.А. Поляков. – М.: Минрадиопром, 1985. – 228 с.
 18. Поляков Г.А. Автоматизация проектирования сложных цифровых систем коммутации и управления / Г. А. Поляков, Ю.Д. Умрихин. – М. Радио и связь, 1988. – 304 с.
 19. Поляков Г.А. Аппарат структур временной семантико-числовой спецификации как основа синтеза параллельных аппаратно-программных средств / Г.А. Поляков, Е.Г. Толстолужская // Параллельная компьютерная алгебра : Всероссийская научная конференция с элементами научной школы для молодежи, 11–15 октября 2010 г. Сборник научных трудов. – С. 31–39.
 20. Поляков Г.А. Гарантоспособные адаптивные системы управления и технологии автоматического проектирования их параллельного аппаратно-программного обеспечения // Радиоэлектронные и компьютерные системы. – Х., 2010. – Вып. 2(43). – С. 32–54.
 21. Поляков Г.А. Синтез и анализ параллельных процессов в адаптивных времяпараметризованных вычислительных системах / Г.А. Поляков, С.И. Шматков, Е.Г. Толстолужская, Д.А. Толстолужский : монография – Х.: ХНУ имени В.Н. Каразина, 2012. – 672 с.
 22. Поляков Г.А. Программа для ЭВМ № 2004610165 «Синтезатор графов Си-программ». РОСПАТЕНТ. Москва, 2004.
 23. Поляков Г.А., Поляков О.Г. Программа для ЭВМ № 2008610573 «Синтезатор параллельных аппаратно-реализуемых моделей Си-программ». РОСПАТЕНТ. Москва, 2008.

FUNCTIONAL SYNTHESIS OF PARALLEL NON REARRANGED SPECIAL PROCESSOR USING THE APPARATUS OF STRUCTURES SEMANTIC NUMBER-SPECIFICATIONS

**G.A. POLYAKOV,
V.V. LYSYKH,
V.V. TOLSTOLUZHSKAYA**

*Belgorod National Research
University*

lyszykh@bsu.edu.ru

In the article presents a generalized algorithm for formal functional / RTL synthesis of parallel computing subsystems, special processor with rigid logic operation. The content of the main stages. The semantics of the stages of synthesis and the results are illustrated with specific examples.

Keywords: computer-aided design of digital devices (CAD), RTL (Register Transfer Level), C-graph problems, Semantic Structures – Number Specifications (SNS), parameterized by the time (time) parallel model of C-programs, temporary parallel graph - the scheme (TPGS), special processor with nonrearranged architecture, a formal functional synthesis of parallel hardware.