

МЕТОД ФУНКЦИОНАЛЬНОГО СЧС-СИНТЕЗА ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПАРАЛЛЕЛЬНО-КОНВЕЙЕРНЫХ ЦИФРОВЫХ УСТРОЙСТВ

Г.А. ПОЛЯКОВ
В.В. ЛЫСЫХ

*Белгородский
государственный
национальный
исследовательский
университет*

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

В статье рассматривается СЧС - метод формального функционального синтеза параллельно - конвейерных цифровых устройств с жесткой логикой функционирования. Используемые методы параллелизма - метод совмещения независимых операций и конвейерный метод. Основной формализма процессов синтеза является применение нового математического аппарата - Алгебры Структур Семантико - Числовой Спецификации, СЧС (вместо языков HDL, Verylog, VHDL, System C.

Ключевые слова: глобализация, вызовы развитию, риски и ограничения социально-экономического развития, региональная политика. 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.

Центральной проблемой современной вычислительной техники является повышение эффективности параллельных вычислительных систем (ВС). Перспективным путем решения проблемы является проектирование проблемно - ориентированных мультипараллельных цифровых устройств и их аппаратная реализация на основе заказных СБИС (ASIC) и/или СБИС программируемой логики (FPGA).

Анализ известных систем EDA показывает, что концепцией их построения является выполнение человеком наиболее сложных, неформализованных, творческих этапов проектирования. Это определяет качество аппаратных средств, сложность, сроки и стоимость проектирования. Однако, возникли проблемы EDA, не имеющие до настоящего времени удовлетворительного решения: разрыва (GAP) между сложностью СБИС, которые может производить электронная индустрия, и предельной сложностью проектов, поддерживаемых известными EDA (проблема System-on-Chip, SOC); неспособность EDA существенно сократить сроки проектирования аппаратных средств (проблема Time-to-Market, T2M) [3, 4, 5, 6].

В отличие от традиционных систем EDA СЧС - метод обеспечивает поддержку следующих новых свойств:

1) интеллектуальность - формальное решение сложных задач проектирования, которые считаются в настоящее время прерогативой специалистов - разработчиков из-за их творческого характера;

2) формальный и автоматический характер проектирования - автоматическое выполнение всех этапов функционального проектирования;

3) адаптивность - автоматическое поддержание высокой эффективности функционального синтеза и результатов проектирования при изменении в широких пределах решаемых задач, областей применения и требований заказчиков к проектируемым объектам;

4) универсальность - возможность функционального проектирования проблемно-ориентированных параллельно-конвейерных цифровых устройств различных классов для различных прикладных областей при различных требованиях и ограничениях.

Целью статьи является описание обобщенного алгоритма формального СЧС синтеза проблемно-ориентированных параллельно-конвейерных цифровых устройств на функциональном уровне проектирования. В отличие от традиционных САПР метод формального функционального синтеза использует для спецификации всех этапов функцио-

нального проектирования новый математический аппарат - Алгебру Структур Семантико-Числовой Спецификации (СЧС).

Исходными данными метода функционального синтеза являются:

- задачи/алгоритмы, представленные исходными текстами Си/Си++ - программ;
- состав используемых методов параллельной обработки данных - совмещение независимых операций, конвейерная обработка данных;
- состав и характеристики элементной базы/библиотеки цифровых функциональных модулей;
- система требований и ограничений (время решения задачи, величина такта/тактовая частота обработки данных, производительность, сложность/стоимость) [1, 2].

Выходные данные метода функционального синтеза:

- структуры семантико-числовой спецификации функциональной схемы устройства;
- графическая спецификация функциональной схемы устройства;
- времяпараметризованная параллельно-конвейерная модель решения задачи устройством;
- значения показателей эффективности параллельно-конвейерного устройства (время решения задачи, величина тактового интервала/тактовая частота, количество функциональных модулей различных типов/суммарная вентиляционная сложность/стоимость)

Обобщенный алгоритм формального функционального СЧС-синтеза (*SSN, Structure Semantic Numericall Synthesis*) представлен на рис. 1.



Рис. 1. Обобщенный алгоритм формального функционального синтеза.

Основные этапы метода функционального синтеза

1. Синтез структур семантико-числовой спецификации (СЧС) и графической спецификации (Си-графа) Си-программы.
2. Синтез времяпараметризованной модели Си-программы, использующей совмещение независимых операций .
3. Синтез структур СЧСМ и графической спецификации (ВПГС) временной параллельно-конвейерной модели Си - программы.
4. Оценка показателей эффективности параллельно-конвейерной модели Си-программы (время, тактовая частота).
5. Проверка выполнения требований: «нет» - выполнение п.6 , «да» выполнение п.7.
6. Изменение параметров ресурса функциональных модулей (ФМ): количества ФМ, параметров ФМ (временной задержки, тактовой частоты), выполнение п.С.
7. Синтез структур СЧСФ и графической спецификации функциональной схемы параллельно - конвейерного цифрового устройства.
8. Вывод результатов формального функционального СЧС - синтеза: СЧС спецификации устройства, функциональной схемы устройства, временной модели работы устройства, показателей эффективности устройства.

Рассмотрим задачу, Си - программу которой представляет рис. 2. Примем, что длительности выполнения операций заданы табл. 1. Требование к длительности такта конвейера $T_T - T_Td = 44$ нс, ограничения на сложность устройства отсутствуют.

```
#include <stdio.h>
void main(void)
{
    int a,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);
    }
}
```

Рис. 2. Си-программа задачи

Таблица 1

Длительности $t^o(t_{opj})$ времени запаздывания в функциональных модулях базы данных (нс)

typ	vx	var	=	==	upl	dmx	/	*	clk	l.o	vix	stop
P(нс)	1.0	1.0	1.05	1.5	0.88	0.68	11.0	41.29	1.0	1.00	1.0	1.0

Таблица 2

**Структура SE семантико-числовой спецификации состава операторов
Си-программы**

N	MET	TYP	NSJ	SJD	BJ	NWJ	WJD	MP1	MP2	VH	VIH	REZ
0	0	58		0	0	0	1	e	0	0	1	a_in
1	0	58		0	0	1	1	0	0	0	1	b_in
2	0	47		0	0	2	1	0	0	0	2	a
3	0	47		0	0	3	1	0	0	0	2	b
4	0	47		0	0	4	1	0	0	0	2	k
5	0	47		0	0	5	1	0	0	0	2	z
6	0	47		0	0	6	1	0	0	0	2	P
7	0	47		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	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		0	0	0	1	0	stop
25	0	48	34	1	3		0	0	0	1	0	k_out
26	0	48	35	1	3		0	0	0	1	0	z_out
27	0	48	36	1	3		0	0	0	1	0	p_out
28	0	48	37	1	3	-1	0	0	0	1	0	s out

Таблица 3

**Структура BE семантико-числовой спецификации
состава операторов Си-программы**

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

Продолжение табл. 3

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

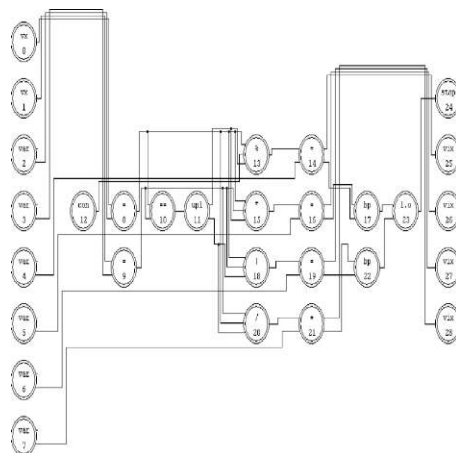


Рис. 3. Графическая спецификация Си-программы

Визуализация временной параллельно-конвейерной модели исходной Си-программы (в виде Временной Параллельной Граф-Схемы, ВПГС) представлена рис. 4. Она содержит перенумерованные ($n_j = 0, 1, 2, \dots, 11$) временные ярусы, множество значений ярусного времени $t(n_j) = 0.00, 1.00, 2.05, \dots, 88.41$ (нс), моменты t_{jB} одновременного начала выполнения которых определяются значением ярусного времени $t(n_j)$, а также связи операторов по данным и по управлению.

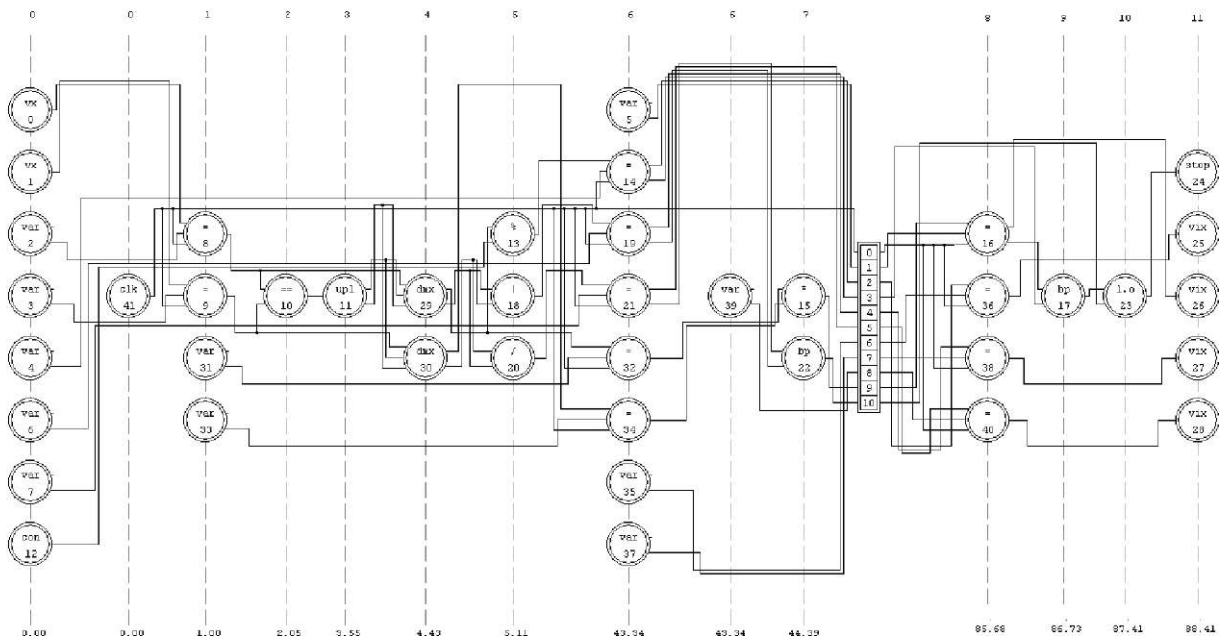


Рис. 4. Графическая спецификация временной параллельно-конвейерной модели исходной Си-программы

Графическая спецификация формально синтезированной функциональной схемы параллельно-конвейерного устройства представлена на рис.5. Устройство обеспечивает значение тактового интервала $TT = 42.34 \text{ нс} < 44 \text{ нс}$ и содержит две функциональные части:

- **исполнительное устройство** (функциональные модули с номерами 8 ...16, 18...21, 29,30, входной интерфейс схемы представляют вход «vx0» (a_in) и вход «vx1» (b_in), выходной интерфейс представляется выходами «vix25» (k_out), «vix26» (z_out), «vix27» (p_out) и «vix28» (s_out), выход «stop»24 обеспечивает выдачу признака завершения решения задачи);
- **устройство управления**, в состав которого входит оператор P41 ввода тактового сигнала «clk», задающего моменты передачи значений данных между конвейерными фрагментами путем синхронизации срабатывания памяти (фиксаторов), и необходимые синхронизирующие связи.

Модули устройства имеют следующую функциональность:

- блоки памяти - регистры (тип выполняемой операции «=») исходных данных RG8, RG9, CON12 нулевой ступени конвейера, регистры - фиксаторы RG14, RG19, RG21, RG32, RG34 первой ступени конвейера FR1, регистры - фиксаторы RG16, RG36, RG38, RG40 второй ступени конвейера FR2[7,8];
- компаратор 10 (блок проверки равенства «==»), обеспечивающий совместно с блоком UPL11 управление разветвлением вычислительного процесса;
- функциональные блоки коммутации (демультиплексоры) DMX29, DMX30, управляемые модулем UPL11 и обеспечивающие активизацию вычислений по одному из двух возможных дальнейших маршрутов: или выполнение операций «%» и «*», либо операций «|» и «/»[9];
- модули BP22, BP17 типа «bp» обеспечивают передачу управления на выходную часть после завершения реализации фрагментов задачи.

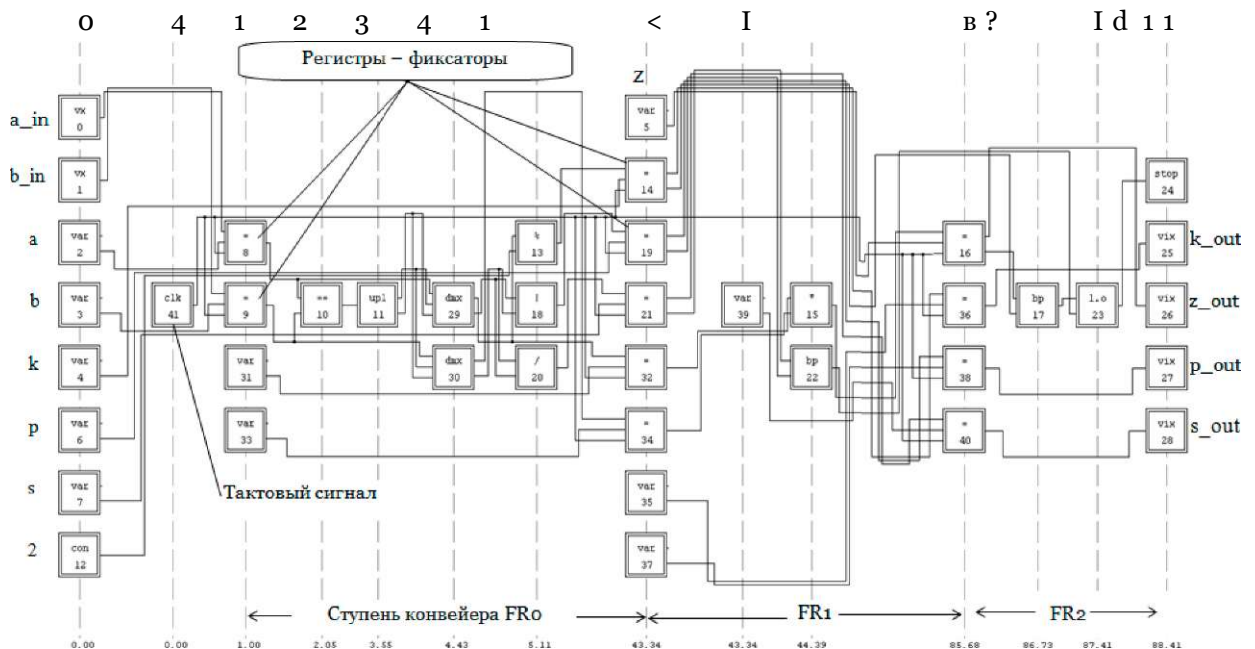


Рис. 5. Результат формального синтеза функциональной схемы параллельно-конвейерного устройства, совмещенный с временной диаграммой функционирования

Разработанный метод позволяет получить функциональную схему параллельно-конвейерного спецпроцессора из Си-программы, исключив при этом субъективные факторы проектирования, связанные со специалистом-проектировщиком. Это обеспечивает возможность существенного уменьшения времени выхода на рынок и многократного увеличения сложности объектов проектирования по сравнению с известными EDA.

Список литературы

1. Поляков, Г.А. Функциональный синтез параллельных неперестраиваемых спецпроцессоров с использованием аппарата структур семантико-числовой спецификации / Г.А. Поляков, В.В. Лысых, В.В. Толстолужская // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. - 2012. - № 13 (132). - Вып. 23/1. - С. 142-150.
2. Поляков, Г.А. «Разработка фрагментированной временной параллельной модели алгоритма гаусса на основе формальных полиномов и структур семантико-числовой спецификации» / Поляков Г.А., Лысых К.В. // Научные ведомости БелГУ: история, политология, экономика, информатика, №19(138) 2012, выпуск 24/1. - Белгород: ИД «Белгород», 2012. - с. 133-136.
3. Поляков, Г.А., Толстолужский, Д.А., Толстолужская, Е.Г. Метод синтеза Си-программ смеси алгоритмов. // Сборник научных трудов СОИ. - Х.: ХУ ВС им. И. Кожедуба. - 2008. - Вып. 1(68) - С. 96-100.
4. Поляков Г.А., Е.Г. Толстолужская. Формальный синтез параллельных программ для высокопроизводительных VLIW - процессоров. // Сборник научных трудов СОИ.- Х.: ХУ ВС им. И. Кожедуба. - 2007. - Вып. 8(66) - С. 72-80.
5. 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.
6. Гаврилов М. А. Логическое проектирование дискретных автоматов (языки, методы, алгоритмы) / М. А. Гаврилов, В. В. Девятков, Е. И. Пурпырев. - М. Наука, 1977. - 352 с.
7. 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.
8. Корячко В. П. Теоретические основы САПР : учеб. для вузов / В. П. Корячко, В. М. Курейчик, И. П. Норенков. - М.: Энергоатомиздат, 1987. - 400 с.
9. Угрюмов Е.П. Цифровая схемотехника. - СПб.:БХВ-Петербург, 2001. -528 с.: ил.

METHOD OF FUNCTIONAL SNS- SYNTHESIS OF PROBLEM-ORIENTED PARALLEL-PIPELINED DIGITAL DEVICES

G.A. POLYAKOV
V.V. LYSYKH

*Belgorod National
Research University*

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

This paper describes a SNS - method of formal functional synthesis of parallel - pipelined digital devices. 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 traditional language HDL, Verylog, VHDL, System C of Hardware Design

Keywords: 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.