

ФУНКЦИОНАЛЬНО ОРИЕНТИРОВАННЫЙ ПЕРЕСТРАИВАЕМЫЙ ПРОЦЕССОРНЫЙ МОДУЛЬ

Н.В. Белова, В.Г. Лобода, Д.А. Петросов

Предлагается общая концептуальная модель процессорного модуля. Анализирующего будущие компоненты с целью получения их различных моделей. Осуществлена композиция компонентов в единую модель, которая исследуется.

процессор, модуль, интегратор, функциональная и техническая структуры

Введение. Все функции вычислительных систем (ВС) можно разделить на основные (базовые) и дополнительные (сервисные). В процессе развития ВС наиболее интенсивным изменениям подвергаются дополнительные функции. Реализуются они отдельными подсистемами, находящимися на различных уровнях системной иерархии. Введение новых функций изменяет соотношения между числом многофункциональных и ориентированных (специализированных) модулей. Таким образом на определенном этапе развития ВС разрешаются противоречия между функциями и структурой, между содержанием и формой [1].

Отображение сервисных функциональных задач (ФЗ) может быть осуществлено на различных уровнях технической структуры ВС. Это – процессоры и интерфейсные компоненты на БИС общего и специального назначения, ориентированные (заказные полузаказные) БИС технических модулей ВС и т.п. Функциональное назначение таких модулей самое разнообразное.

Анализ источников [2] позволяет сделать вывод о том, что проблема реализации ориентированных (сервисных, дополнительных, специализированных) ФЗ в составе ВС и автономным образом (для децентрализованных ВС) является необходимой и достаточно сложной. Одним из путей реализации таких ФЗ является построение (синтез) функционально-ориентированных процессоров.

В статье предлагается новый функционально-ориентированный процессор с перестраиваемой структурой и функциями. Материал статьи – продолжение работ, выполненных авторами в рамках единого научно-технического направления [2].

1. Общая концепция построения процессорного модуля (ПМ). В составе подсистем обработки данных (ПОД) находят применение, в ос-

новном, однокристалльные микропроцессоры (микроконтроллеры) типов CISC и RISC (ОКМП). Предлагаются также MISC-архитектуры, адаптирующиеся к различным участкам глобального микроалгоритма с целью увеличения производительности МП. MISC-процессоры строятся с использованием принципа организации разрядно-модульных или многокристалльных МП (МКМП) [2, 3]. Такие ОКМП и МКМП, как правило, вычисляют значения функций для отдельных конкретных значений аргументов или для достаточно редкой последовательности значений аргументов, при которой функция – между соседними значениями или наборами аргументов изменяется настолько существенно, что это изменение имеет один порядок с величинами функций в рассматриваемых точках [4]:

$$|f(x_i) - f(x_{i-1})| \geq 10^{-1} \max|f(x)|. \quad (1)$$

Однако в практике применяются ПОД имеют место вычисления функций и в столь близко расположенных точках, что изменения функций при переходе от одной точки к соседней достаточно малы, причем для функций одной переменной [4]:

$$|f(x_i) - f(x_{i-1})| \leq \varepsilon \max|f(x)|. \quad (2)$$

В случае (1) информация представляется и перерабатывается полно-разрядно, в случае (2) ПОД должна производить операции над приращениями переменных. Такое инкрементное (унитарное) представление входной и выходной информации необходимо для систем управления многими типами объектов (связь, роботы, станки и т.п.). Для реализации инкрементных вычислений наиболее часто используется метод систем дифференциальных уравнений Шеннона и разностные системы уравнений [4].

Таким образом, имеет место два способа функционального преобразования – преобразования в стационарном (1) и динамическом (2) режимах. Тогда, очевидно, эффективным способом использования двух режимов преобразования информации будет создание единого перестраиваемого ПМ, общая структура которого представлена на рис. 1 в виде «черного ящика», где введены следующие обозначения: $\Delta x(dx)$ и $\Delta y(dy)$ – конечные (бесконечно малые) приращения входных и выходных переменных соответственно; X и Y – полноразрядные векторы входных (m разрядов) и выходных (n разрядов) двоичных слов соответственно; Z – q -разрядный вектор управления.

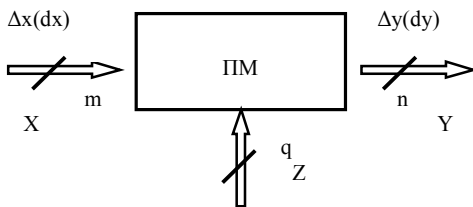


Рис. 1. Процессорный модуль

Топология растекания вычислительного процесса по внутренней структуре «черного ящика» ПМ назначается управляющим вектором Z .

Раскрыть содержимое «черного ящика» можно с помощью процедур получения «белого ящика», для чего используется системный подход к анализу и синтезу ВС [5].

2. Структурно-функциональный анализ и синтез ПМ. Рассмотрим вначале модель ПМ на базе любого типа МП, реализующую алгоритмы стационарных функциональных макроопераций над полноразрядными переменными.

Простейшая аналитическая модель отображает $\Phi Z_{МП}$ в виде векторных пространств вида

$$Y = F(X), \quad (3)$$

т.е. имеет место функциональная зависимость. $\Phi Z_{МП}$ – оператор преобразования в МП некоторого входного потока данных в выходной поток.

Алгоритмическая модель описывает $\Phi Z_{МП}$ в форме системы управлений. В частности – это алгоритмы вычисления функций одной переменной, которые базируются на аппроксимации функций $y(x)$ степенными рядами, бесконечными произведениями, суммами простейших дробей, степенными многочленами, цепными дробями, рациональными дробями, сплайн-функциями [4].

С другой стороны, $\Phi Z_{МП}$ есть подмножество ΦZ на следующем нижнем уровне функциональной иерархии МП. Элементами такого подмножества являются $\Phi Z_{вв}$ – ввода информации, $\Phi Z_{хоб}$ – хранения и обработки информации, $\Phi Z_{выв}$ – вывода информации во внешнюю среду, т.е.

$$\Phi Z_{МП} = \{\Phi Z_{вв}, \Phi Z_{хоб}, \Phi Z_{выв}\}.$$

Тогда древовидную функциональную модель $\{MF\}$ МП любого типа

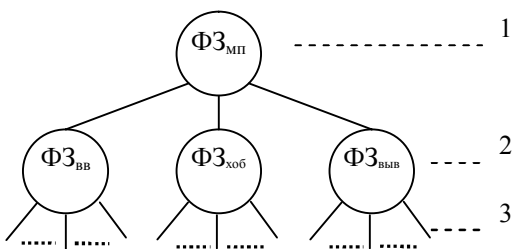


Рис. 2. Древовидная модель МП

можно представить в виде, показанном на рис. 2 [2, 3].

Вообще говоря, проблема моделирования МП усложняется наличием большого множества МП, отличающихся организацией, номенклатурой команд, способами адресации, емкостью памяти и

другими техническими параметрами (характеристиками). Однако если применить метод обобщения, основанный на генетических алгоритмах развития организации МП БИС, то можно будет получить следующую (пусть и примитивную на данном уровне) программистскую модель МП

[6 – 8]. Такая модель МП изображена на рис. 3, где $R_1 \dots R_7$ – программно доступные регистры МП. Конкретно: R_1 – аккумулятор, R_2 – регистр (регистры) общего назначения, R_3 – сверх-оперативный регистр, R_4 – указатель верхушки стека, организуемого в основной (главной) памяти ПОД, R_5 – адресный буферный регистр для запоминания адресов операндов, R_6 – программный счетчик (счетчик команд), R_7 – регистр подпрограмм для сохранения адреса возврата (принимаем здесь условия существования лишь одного уровня вложения подпрограмм).

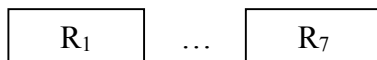


Рис. 3. Программистская модель МП

Для дальнейших процедур анализа необходимо раскрыть конкретное содержание $\Phi Z_{\text{вв}}$, $\Phi Z_{\text{хоб}}$, $\Phi Z_{\text{выв}}$, размещенные на втором уровне древовидной модели МП. $\Phi Z_{\text{вв}}$ и $\Phi Z_{\text{выв}}$ – задачи обмена с внешней средой ($\Phi Z_{\text{обм}}$). Это процедуры «IN», «OUT», реализуемые с помощью очередей и диспетчеризации в последних разработках МП известных ведущих фирм [2].

В данной работе не будем подвергать анализу процедуры связи «IN – OUT», но детализируем $\Phi Z_{\text{хоб}}$ на 3-м уровне дерева функций {MF}.

Введем нижеследующие обозначения с необходимыми комментариями: $\{I\}_{21}$ – множество операций (команд, инструкций) $\Phi Z_{\text{хоб}}$; I_1 – загрузить регистр R_1 из основной памяти, используя прямую адресацию (Т); I_2 – загрузить регистр R_2 из основной памяти, используя прямую адресацию (Т); I_3 – переслать содержимое регистра R_1 в регистр R_2 (Т); I_4 – сложить содержимое регистров R_1 и R_2 и запомнить результаты в регистре R_1 (М); I_5 – переслать содержимое регистра R_1 в регистр R_3 (Т); I_6 – переслать содержимое регистра R_3 в регистр R_1 (Т); I_7 – запомнить данные в регистре R_1 в основной памяти, используя косвенную адресацию (Т); I_8 – запомнить данные в регистре R_2 в основной памяти, используя метод неявной адресации (Т); I_9 – отдать команду перехода (В); I_{10} – пропустить команду, если содержимое регистра R_1 равно нулю (В); I_{11} – осуществить поразрядный сдвиг содержимого регистра R_1 влево на один бит (М); I_{12} – дополнить (с помощью побитовой операции) содержимое регистра R_1 (М); I_{13} – осуществить логическое умножение содержимого регистров R_1 и R_2 и запомнить результаты в регистре R_1 (М); I_{14} – неисполняемая команда (В); I_{15} – загрузить указатель стека (R_4) из основной памяти, используя прямую адресацию (Т); I_{16} – продвинуть регистр R_1 в стеке, организованном в основной памяти по дисциплине LIFO (Т); I_{17} – запомнить данные регистра R_2 в основной памяти, используя прямую адресацию (Т); I_{18} – сдвинуть начало стека и запомнить его в R_1 (Т); I_{19} – загрузить регистр R_2 из основной памяти, используя прямую адресацию (Т); I_{20} – переход к подпрограмме (адрес возврата сохраняется в регистре подпрограмм R_7) (В); I_{21} – возврат из подпрограммы (В).

Учитывая вышесказанное и введенные обозначения можно построить графовую модель (граф функционирования) МП ($\PhiЗ_{МП}$) [6, 7]. Графовая модель МП представлена на рис. 4.

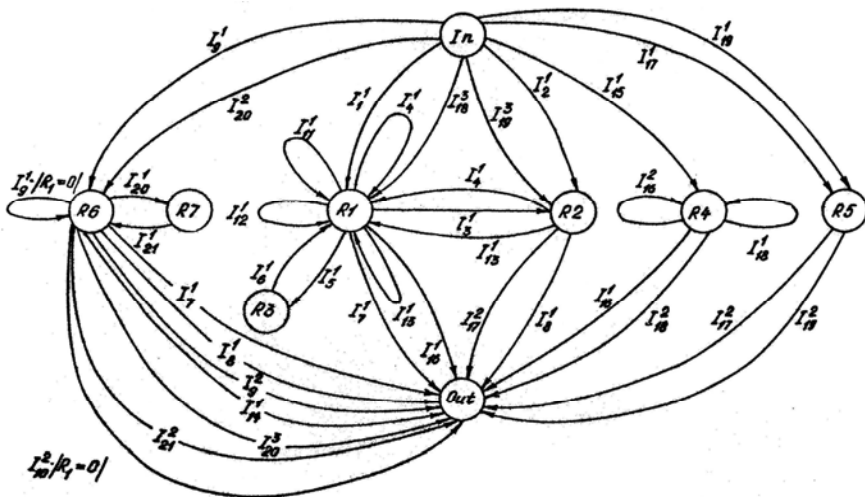


Рис. 4. Графовая модель МП

Вершины графа отображают основные регистры включая входной и выходной, дуги отображают обмен информацией, между ними при выполнении различных команд, а петли у соответствующих вершин – внутрирегистровые манипуляции. В системе операций $\{I\}_{21}$ отличается три вида инструкций: передачи (Т – transfer), манипуляции с данными (М – manipulation) и ветвления (В – branch). Любые информационные потоки во время выполнения любой инструкции представляются в графо-теоретической модели следующим образом:

- существует направленная дуга из вершин R_p к вершине R_q , если во время выполнения I_j , передаются данные из регистра R_p в регистр R_q (с манипуляциями или без них);
- существует направленная дуга из вершин IN к вершине R_j , если во время выполнения I_j передаются данные из основной памяти или от устройств ввода-вывода в регистр R_j (с манипуляциями или без них);
- существует направленная дуга из вершин R_j к вершине OUT , если во время выполнения операции передаются данные из регистра R_j в основную память (или ее адресный регистр) или к устройствам ввода-вывода (с манипуляциями или без них).

Точная последовательность передачи данных может быть неизвестна, поскольку неизвестны детали функционирования оборудования МП.

Однако можно вывести порядок выполнения передач на основе анализа логической зависимости данных.

Выполним теперь системный анализ ядра ПОД, предназначенного для реализации $\Phi Z_{ин}$ динамического принципа функционирования (2), основанного на методах уравнений Шеннона (d) и разностных систем уравнений (Δ). В общем виде система дифференциальных уравнений Шеннона записывается следующим образом [4]:

$$dy_{pk} = A_{pk1}dx + \sum_{j=2}^N A_{pkj}y_{pj}dy_{qj}; \quad dy_{qk} = A_{qk1}dx + \sum_{j=2}^N A_{qkj}y_{pj}dy_{qj}; \quad (4)$$

$$y_{pk}(x_0) = y_{pko}, \quad k = \overline{2, N}.$$

По своей сути система (4) является алгоритмической моделью $\Phi Z_{ин}$. Здесь $A_{pkj} \in \{0, 1\}$ и $A_{qkj} \in \{0, 1\}$, $j = 1, 2, \dots, N$ – константы. Наборы таких констант полностью задают ту или иную конкретную систему (4) при моделировании, например, динамических объектов, описываемых линейными или нелинейными системами обыкновенных дифференциальных уравнений. Так как $\Phi Z_{ин} = \langle \Phi Z_{хоб}, \Phi Z_{обм} \rangle$, то вновь оставляя в стороне анализ $\Phi Z_{обм}$ заметим, что $\Phi Z_{хоб}$ сводится к простейшим операциям умножения переменных на дифференциалы $y_{pj}dy_{qj}$ и операциям суммирования этих членов.

Это обстоятельство позволяет осуществить интегрирование уравнений Шеннона с помощью только двух типов макроопераций – интегрирования и суммирования. Поэтому ПМ, работающий в динамическом режиме (2), должен иметь в своем составе два устройства – интегратор (ИН) и сумматор (СМ). А так как современные вычисления строятся на базе цифровой техники, то интеграторы и сумматоры должны быть цифровыми. Тогда система (4) необходимо будет свести к форме разностных уравнений путем замены оператора d Δ . Такую замену осуществить часто не просто. Таким образом, имея конечное число ИН и СМ, любую ФЗ можно решить структурно-аппаратным методом, если только ФЗ удовлетворяет условиям Шеннона.

Детализируем $\Phi Z_{хоб}$ на третьем уровне дерева функций {MF} для общей $\Phi Z_{ин}$ ПМ типа ИН, которые в литературе носят название цифровые дифференциальные анализаторы, инкрементные машины, цифровые интегрирующие машины. Основным устройством в таких машинах является цифровой интегратор [4, 9]. Если принять простейший метод численного интегрирования по Эйлеру, то приближенное значение интеграла на отрезке оси $x_0 - x_n$ можно определить следующим образом:

$$\int_{x_0}^{x_n} ydx \approx \sum_{i=0}^{n-1} y_i \Delta x_i, \quad \Delta x_i = x_{i+1} - x_i. \quad (5)$$

Переход от дифференциальных выражений к разностным вносит погрешность ε , задаваясь которой можно найти такую величину Δx , что

$$\left| \int_{x_0}^{x_n} y dx - \sum_{i=0}^{n-1} y_i \Delta x_i \right| < \varepsilon. \quad (6)$$

Если все $\Delta x_i = \text{const}$ и одинаковы, то приближенное значение интеграла

$$S_n = \Delta x \sum_{i=0}^{n-1} y_i. \quad (7)$$

Аналитическая модель (7) ЦИН позволяет получить регистровую (программистскую) модель ЦИН. Обобщенная такая модель показана на рис. 5 (ΔS – переполнение регистра). На входах (Δx , Δy) и выходе ΔS ПМ типа ЦИН имеют место только конечные приращения переменных.

Регистровая модель ЦИН является конечным цифровым автоматом (ЦА) второго рода. Поэтому функционирование такого ЦА можно описать [10] с помощью графов, представленных на рис. 6, 7.

В результате структурно-функционального анализа ПМ типа МП и ПМ типа ЦИН имеет две графовые модели (рис. 4, 7) поведения ЦА Неймана (ЦАН) и ЦА Шеннона (ЦАШ). Для получения (синтеза) общей модели функционально-ориентированного перестраиваемого ПМ (ФОППМ) необходима композиция ЦАН и ЦАШ. Сигнал Δx – вид команды (инструкции) сложения с переполнением сумматора ΔS без останова МП.

В результате проведенного структурно-функционального анализа и синтеза можно сделать следующий вывод: внутренние ресурсы МП любого типа позволяют модифицировать функциональную (регистровую) структуру МП так, что он может работать либо в режиме полных слов, либо в режиме их приращений. Модификацию можно производить с помощью специализированного внутреннего коммутатора (сигнал Z) [4].

3. Отображение функциональной модели ФОППМ на технические модули. Для получения конечного продукта в виде технической структуры ФОППМ вновь будем следовать в соответствии с алгоритмом вида [2, 11]:

$$\{MF\} \Rightarrow \{O\} \Rightarrow \{\Phi\} \Rightarrow \{T\}, \quad (8)$$

где $\{O\}$, $\{\Phi\}$, $\{T\}$ – операторная, функциональная и техническая модели.

Осуществим отображение по закону α :

$$\{\Phi\} \xrightarrow{\alpha} \{T\}. \quad (9)$$

Эта задача сводится к задаче получения минимального замкнутого покрытия двух классов ЦА – ЦАН и ЦАШ [10 – 12]. Если подмножество операторов состояния ЦАН покрывает подмножество операторов состояния ЦАШ, то следует переходить к следующему алгоритмическому уровню декомпозиции ЦАН и ЦАШ, до тех пор, пока не будет получено

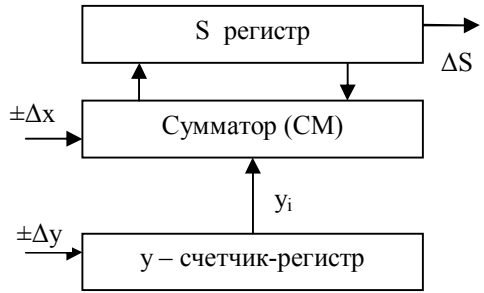
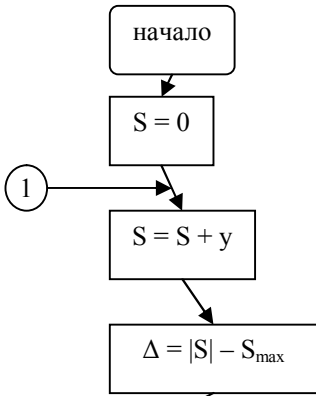


Рис. 5. Регистровая модель ЦИН

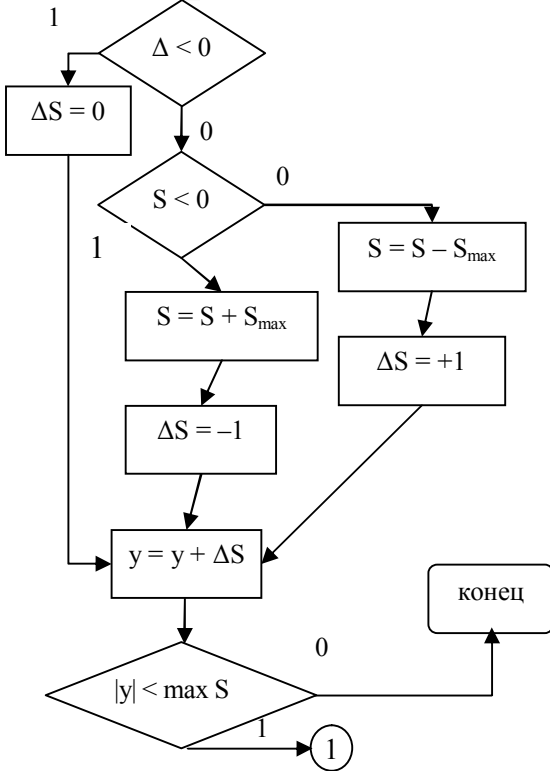


Рис. 6. Алгоритм работы ЦИН в режиме единичной обратной связи $\Delta S - \Delta y$

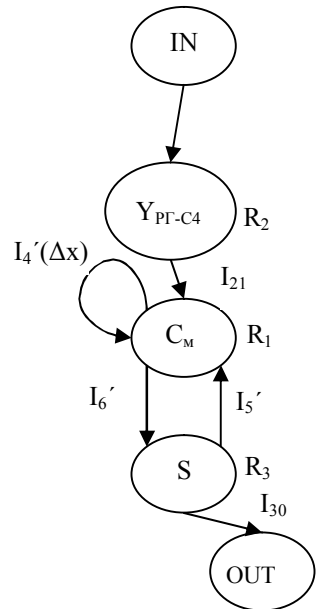


Рис. 7. Графовая модель ЦИН

покрытие одного ЦА другим (либо наоборот) на всех алгоритмических уровнях. Результатом таких процедур является единая минимальная алгоритмическая модель функционально-ориентированного перестраиваемого ПМ.

Число операторных вершин на графах рис. 4, 7 относительно не велико, поэтому здесь нет необходимости в формализации процедур перебора совместимых (несовместимых) пар состояний по М. Поллу и С. Ангеру [10]. Очевидно, что графовая модель МП покрывает графовую модель ЦИН с введением в первую инструкций передач типа I_{21} и I_{30} . Набор конкретных технических модулей для реализации Т должен находиться в полном соответствии с Ф. Техническая структура функционально-ориентированного перестраиваемого ПМ должна иметь организацию, позволяющую осуществлять возможные изменения функционально-структурных и программно-аппаратурных соотношений.

Иерархия уровней Т позволяет выбрать любые технические модули – от БИС (СБИС), ПЛИС до ИС. Но при этом необходимо соблюдать закон отображения α (9), который может быть и нечетким [5]. Воспользуемся здесь критерием соответствия заданным нормативным показателям [13].

Примем в качестве исходного базового технического модуля БИС МП. Такой выбор не противоречит всем предыдущим рассуждениям. Условия применимости БИС МП [13] перечислены ниже.

$$1. \quad x_i \geq 2; \quad y_j \geq 2; \quad Q_1 \geq 2, \quad (10)$$

где x_i – операнд, принимаемый по i -му каналу ввода; y_j – результат, отмеченный на j -м канале вывода; Q_1 – множество состояний регистров модели МП.

2. Полный граф коммутации.

3. Выполняются микроинструкции (микрокоманды) инкремента, алгебраического сложения, булевых операций, сдвига (все виды), сложения по модулю два.

4. Перечисленные микрооперации выполняются над содержимым любых источников информации в соответствии с полным графом коммутации:

$$\begin{aligned} Q_{(l+1)} &= \delta_1(Q_1, x_i, x_0); \\ Q_{(l+1)} &= \delta_2(Q_1, Q_2, x_0); \\ Q_{(l+1)} &= \delta_3(x_1, x_2, x_0); \\ y_i &= \lambda_1(Q_1, x_i, x_0); \\ y_j &= \lambda_2(Q_1, Q_2, x_0); \\ y_j &= \lambda_3(x_1, x_2, x_0). \end{aligned} \quad (11)$$

Если выполняются все условия, то $K_1 = K_2 = K_3 = K_4 = 1$. Тогда структурная полнота определяется как

$$K = \sum_{i=1}^4 K_i / \sum_{i=1}^4 K_{i\max} = 1. \quad (12)$$

При невыполнении одного из условий $K < 1$.

Условию (12) удовлетворяют все типы и ОКМП, и МКМП. Однако МКМП более гибкий в применении, так как не имеет собственной системы команд.

В данной работе для схемной эмуляции ФОППМ выбрана процессорная секция (чип) комплекта К 1804 для дальнейших экспериментов.

4. Моделирование работы технической структуры ФОППМ. Для моделирования подобных устройств широко используются методы программной эмуляции, К-значного моделирования, сети Петри и VHDL-среда, как международный инструментарий [2].

Для программной эмуляции была выбрана схема типа КМ 1804ВС1. При этом использовались формальные параметры, описывающие состояния сигналов на всех выводах БИС. Программная модель состоит из девяти подпрограмм, реализующих алгоритм работы, КМ1804ВС1. Тексты подпрограмм в данной работе не приводятся из-за ограничения на ее объем. Имитация работы ФОППМ проводилась [14] при вычислении функций вида $y = e^x$ и $y = \sin x$ в режиме (1) и режиме (2). Причем для реализации (1) функции раскладывались в ряд Тейлора, а для реализации (2) – использовались приемы получения дифференциальных уравнений, приведенные в [4, 9].

Моделирующая среда VHDL была применена как альтернативный вариант для схем типа К1804ВМ1. Листинг программы, вычисляющей функции $y = e^x$ и $y = \sin x$ в режимах (1) и (2), содержит объемную информацию и здесь не приводится.

Результат вычислений принятых функций различными по типу схемами в различных режимах работы моделей ФОППМ показал полное (в пределах $\pm 8\%$) совпадение при изменении аргумента x в пределах $0 \dots \pi/2$. Длина разрядной сетки – один байт.

Заключение. В работе рассмотрены вопросы построения ПМ реализующего режимы полноразрядного функционального преобразования входных слов и унитарного (инкрементного) преобразования. Проведен структурно-функциональный анализ составляющих компонентов и синтез общей функциональной структуры ПМ. Осуществлен выбор технических модулей для построения моделирования работы процессорных модулей на программном уровне и в среде VHDL.

Материал статьи может быть полезным для разработчиков заказных БИС.

ЛИТЕРАТУРА

1. Балаиов Е.П. Эволюционный синтез систем. – М.: Радио и связь, 1895. – 328 с.
2. Лобода В.Г., Петросов Д.А. Концепция построения структур функционально ориентированных вычислительных устройств // АСУ и приборы автоматики. – 2003. – Вып. 122. – С. 61 – 71.
3. Белова Н.В., Долженкова Т.Г. Подсистема обработки данных с программируемой структурой // 8-я Международная конференция «Теория и техника передачи, приема и обработки информации» («ИИСТ-2002»): Сб. научных трудов. – Х.: ХНУРЭ. – 2002. – С. 355 – 356.
4. Каляев А.В. Микропроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.
5. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. – М.: Сов. радио, 1973. – 440 с.
6. Thatte S.M., Abraham J.A. Test generation for microprocessors // IEEE Tras. Comput. – 1980. – С – 29. – № 6. – Р. 429 – 441.
7. Белова Н.В., Долженкова Т.Г. Петросов Д.А. Модель перепрограммируемого процессора с гибкой архитектурой // Сб. материалов 8-го Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Х.: ХНУРЭ. – 2004. – Ч. 2. – С. 277.
8. Петросов Д.А., Лобода В.Г., Ельчанинов Д.Б. Представление генетических алгоритмов сетями Петри в задачах проектирования компьютерной техники // Материалы научно-практической конференции «Информационные технологии – в науку и образование». – Х.: ХНУРЭ. – 2005. – С. 48 – 51.
9. Неслуховский К.С. Цифровые дифференциальные анализаторы. – М.: Машиностроение, 1968. – 260 с.
10. Баранов С.И. Синтез микропрограммных автоматов. – Л.: Энергия, 1979. – 232 с.
11. Белова Н.В., Механа Сами Саади, Петросов Д.А. Концепция буферизации в вычислительных устройствах и системах. // Сб. научн. трудов 6-го Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Х.: ХНУРЭ. – 2002. – Ч. 2. – С. 36 – 37.
12. Белова Н.В., Долженкова Т.Г., Петросов Д.А. Минимальное замкнутое покрытие двух классов цифровых автоматов // Сб. материалов 7-го Международного молодежного форума «Радиоэлектроника и молодежь XXI веке». – Х.: ХНУРЭ. – 2003.– С. 468.
13. Лобода В.Г., Цуканов В.Ю. Базовый специализированный процессор с открытой архитектурой // Сб. научных трудов «Автоматика и приборостроение»: Вестник национального технического университета «ХПИ». – Х.: НТУ «ХПИ». – 2001. – Вып. 114. – С. 95 – 100.
14. Петросов Д.А., Цуканов В.Ю. MISC-компилятор // Радиоэлектроника и информатика. – 2000. – № 2. – С. 58 – 62.

Поступила 16.03.2005

Рецензент: доктор физико-математических наук профессор С.В. Смеляков,
Харьковский университет Воздушных Сил.