УДК 621.3

ПРЕДПОСЫЛКИ СОЗДАНИЯ МЕТОДА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ, ОСНОВАННОГО НА ПРОСТРАНСТВЕННО - ВРЕМЕННОЙ МОДЕЛИ

к.т.н. О.И. Богатов, А.В. Низовцев, Е.В. Кондратьева,
В.Н. Кондратьев, И.И. Прокопенко
(представил д.т.н., проф. Г.А. Поляков)

Обосновывается выбор математической модели элемента вычислительного процесса в мультипроцессорной вычислительной системе (МВС), в которой в качестве процессорного элемента (ПЭ) используются программируемые логические интегральные схемы (ПЛИС), а также выбор адекватного алгоритмического языка описания структуры системы и происходящих в ней процессов на всех этапах проектирования. Предлагается общая методика проектирования (конфигурирования) МВС на основе ПЛИС.

Как известно [1], эффективность автоматизированных систем проектирования МВС, как и любых сложных цифровых систем, непосредственно в значительной степени зависит от выбора соответствующих математических моделей, применяемых на каждом из этапов проектирования, а также алгоритмических языков описания этих моделей. Применение ПЛИС в качестве ПЭ в вычислительной системе создает реальные предпосылки дальнейшего роста производительности МВС. В соответствии с [2] методология проектирования МВС на основе ПЛИС предполагает три последовательных этапа:

1) представление алгоритма задачи, подлежащей реализации на МВС, в терминах соответствующих фрагментов алгоритма и операторов управления;

2) перевод алгоритма, представленного в терминах фрагментов в абстрактную пространственно-временную модель вычислительного процесса решения задачи, учитывающую как организацию связей между ПЭ, реализующих каждый фрагмент, так и схему реализации временной последовательности их работы;

3) перевод абстрактной пространственно-временной модели в конкретную схему на ПЛИС с назначением выводов ПЛИС, всеми связями между ними и схемной организацией последовательности управляющих сигналов.

Следовательно, представляется разумным выбрать пространственно-временную модель, как некоторое промежуточное звено между алгоритмом и структурой устройства, исходя из следующих требований. Во-первых, в модели обязательно должны быть отражены элементы алгоритма, по крайней мере, с точностью до фрагмента. Во-вторых, модель должна быть такой, чтобы существовал тривиальный переход от нее к соответствующему устройству, обеспечивающему пространственно-временные характеристики, заложенные в модели. И, наконец, должен существовать математический аппарат, удовлетворяющий двум требованиям: 1) он должен обеспечить возможность получать сложные модели из более простых; 2) для него должны существовать тождественные преобразования, позволяющие формальным способом варьировать пространственными и временными параметрами модели.

В [1] в качестве базовой системы автоматизированного проектирования любых компьютерных систем предлагается система ПРОЕКТ, для которой были разработаны языки описания алгоритмов и структур соответственно. Нам представляется целесообразным, чтобы на всех указанных выше этапах проектирования использовался единый язык, тем более, что языки описания и моделирования проектируемых цифровых структур реально существуют. Из них наиболее подходящим является язык VHDL [3], разработанный в США по заказу Министерства обороны и с 1987 г. ставший стандартом IEEE Std1076, а фактически мировым стандартом. Язык VHDL по сравнению с другими алгоритмическими языками дополнен средствами учета временных задержек, что дает возможность использовать его и на втором этапе. Он позволяет описывать с необходимой степенью детализации цифровые схемы любой сложности, начиная с тривиальных вентилей и кончая самыми мощными современными вычислительными комплексами. Поэтому он полностью применим и на третьем этапе проектирования.

Исходя из изложенных выше принципов, введём понятие пространственно-временной модели (**ST**-модели*)*. ***ST****-моделью элемента вычислительного процесса* будем называть множество $\left(\mathbf{A},\mathbf{F},\lambda,\eta,\sigma,\tau\right)$, где $\mathbf{A}$ – алфавит (обычно $\mathbf{A}=\{\mathbf{0},\mathbf{1}\}$); $\mathbf{F}:\mathbf{A}^{\mathbf{m}}\to\mathbf{A}^{\mathbf{n}}$ – вычисляемая функция для целых $\mathbf{m}\geq\mathbf{1}$, $\mathbf{n}\geq\mathbf{1}$; $\lambda:\{\mathbf{0},\mathbf{1},\ldots,\mathbf{m}-\mathbf{1}\}\to\{\mathbf{s},\mathbf{s}+\mathbf{1},\ldots,\mathbf{s}+\sigma_{\mathbf{0}}-\mathbf{1}\}\times\{\mathbf{t},\mathbf{t}+\mathbf{1},\ldots,\mathbf{t}+\tau_{\mathbf{0}}-\mathbf{1}\}$ – входное биективное отображение, $\mathbf{s}$, $\mathbf{t}$, $\sigma_{\mathbf{0}}$, $\tau_{\mathbf{0}}$ - некоторые натуральные числа; $\eta:\{\mathbf{0},\mathbf{1},\ldots,\mathbf{n}-\mathbf{1}\}\to\{\mathbf{p},\mathbf{p}+\mathbf{1},\ldots,\mathbf{s}+\sigma_{\mathbf{1}}-\mathbf{1}\}\times\{\mathbf{q},\mathbf{q}+\mathbf{1},\ldots,\mathbf{q}+\tau_{\mathbf{1}}-\mathbf{1}\}$ – выходное биективное отображение, $\mathbf{p}$, $\mathbf{q}$, $\sigma_{\mathbf{1}}$, $\tau_{\mathbf{1}}$ - некоторые натуральные числа, $\sigma\geq\mathbf{max}\left(\sigma_{\mathbf{0}},\sigma_{\mathbf{1}}\right)$ – константа относительного пространственного сдвига данных; $\tau\geq\mathbf{max}\left(\tau_{\mathbf{0}},\tau_{\mathbf{1}}\right)$ – константа относительного временного сдвига данных. Символы λ и η для определенности иногда будем обозначать $\lambda_{\mathbf{s},\mathbf{t}}$ и $\eta_{\mathbf{p},\mathbf{q}}$.

Отображение $\lambda_{\mathbf{s},\mathbf{t}}$ определяет способ ввода в **ST**-модель исходных данных, соответственно отображение $\eta_{\mathbf{p},\mathbf{q}}$ – способ вывода данных, полученных в результате вычисления. Из биективности $\lambda_{\mathbf{s},\mathbf{t}}$ и $\eta_{\mathbf{p},\mathbf{q}}$ следует:

$$\sigma_{\mathbf{0}}\cdot\tau_{\mathbf{0}}\geq\mathbf{m},\ \sigma_{\mathbf{1}}\cdot\tau_{\mathbf{1}}\geq\mathbf{n}. \qquad (1)$$

С содержательной точки зрения это определение обозначает, что **ST**-модель переводит вычисляемую элементом любую функцию $\mathbf{Y}=\mathbf{F}\left(\mathbf{X}\right)$ для $\mathbf{X}\in\mathbf{A}^{\mathbf{m}}$, $\mathbf{Y}\in\mathbf{A}^{\mathbf{n}}$ из одномерной плоскости в последовательной записи не-которого алгоритма во взаимно однозначно соответствующую ей функцию

$$\mathbf{W} = \mathbf{F}_{\mathbf{s,t}}(\mathbf{V}), \qquad (2)$$

определенную и принимающую значения в двухмерной пространственно-временной плоскости, где $\mathbf{V} \in \mathbf{A}^{\sigma_0} \times \mathbf{A}^{\tau_0}$, $\mathbf{W} \in \mathbf{A}^{\sigma_1} \times \mathbf{A}^{\tau_1}$. Функциональный символ $\mathbf{F}$ постараемся сохранить и для этого случая. Таким образом, **ST**-модель соответствует преобразованию данных в пространственно-временной области, что соответствует реальному вычислительному процессу.

Покажем, что **ST**-модель $(\mathbf{A}, \mathbf{F}, \lambda, \eta, \sigma, \tau)$, заданная (2), действительно реализует любую функцию $\mathbf{Y} = \mathbf{F}(\mathbf{X})$ для любых значений $\mathbf{X} \in \mathbf{A}^{\mathbf{m}}$, $\mathbf{Y} \in \mathbf{A}^{\mathbf{n}}$. Из (2) и определения **ST**-модели следует:

$$\mathbf{V} = \lambda_{\mathbf{s,t}}(\mathbf{X});\ \ \mathbf{W} = \mathbf{F}_{\mathbf{s,t}}(\lambda_{\mathbf{s,t}}(\mathbf{X}));\ \ \mathbf{Y} = \eta^{-1}(\mathbf{W}) = \eta^{-1}(\mathbf{F}_{\mathbf{s,t}}(\lambda_{\mathbf{s,t}}(\mathbf{X}))),$$

т.е. функция $\mathbf{F}$ в $\mathbf{Y} = \mathbf{F}(\mathbf{X})$ представляет собой композицию отображений

$$\mathbf{F} = \eta^{-1} \circ \mathbf{F}_{\mathbf{s,t}} \circ \lambda_{\mathbf{s,t}}.$$

Функция $\mathbf{F} : \mathbf{A}^{\mathbf{m}} \to \mathbf{A}^{\mathbf{n}}$ однозначно соответствует некоторому фрагменту алгоритма. Поскольку структура алгоритма предполагается регулярной, то соответствующий фрагмент в общем случае должен встречаться в нём неоднократно. Следовательно, идентификатором любого такого фрагмента может служить символ **F**. С другой стороны, каждому фрагменту взаимно однозначно соответствует некоторая **ST**-модель. Естественно, возникает вопрос, в каком плане может быть использована эта модель для реализации всего алгоритма, по крайней мере, в части реализации всех его фрагментов, имеющих идентификатор **F**. Согласно определению фрагмента ему соответствует некоторое устройство, которое может реально вычислять соответствующую функцию, определённые черты которой отражены в соответствующей ему **ST**-модели. Пусть это устройство уже использовалось ранее для реализации такого фрагмента алгоритма с некоторыми известными для него исходными данными. Пусть также далее по ходу выполнения алгоритма встречается еще один такой фрагмент с отличными от прежних исходными данными. Тогда согласно определению **ST**-модели эти исходные данные должны быть поданы на те же входные каналы устройства спустя не менее, чем через τ дискретных интервалов времени. Естественно, и результаты реализации фрагмента с новыми исходными данными должны появиться на выходных каналах этого устройства также не ранее, чем через τ дискретных интервалов времени. Во втором случае предполагается введение еще одного полностью аналогичного устройства, также реализующего фрагмент с идентификатором **F**. Действуя согласно предложенным принципам, как в первом, так и во втором случае, мы получим некоторый вычислительный процесс, позволяющий реализовать выполнение двух фрагментов алгоритма, хотя и различными техническими способами. В первом случае выполнение фрагмента осуществляется последовательным способом одним устройством; во втором для этой цели используются два идентичных устройства, работающие параллельно. В любом случае мы имеем право сохранить символ **F** в качестве идентификатора **ST**-модели процесса преобразования данных во всех устройствах, которые физически реализуют функции $\mathbf{F : A^m \to A^n}$ с любыми исходными данными в едином дискретном пространстве и едином дискретном времени. Таким образом, **ST**-модель сочетает в себе как элементы алгоритмического описания, так и черты устройств, реализующих эти элементы. Покажем, что различие способов реализации двух фрагментов с одним и тем же идентификатором, рассмотренное выше, может быть математически отражено парой индексов при символе **F**, взятом в качестве идентификатора соответствующих фрагментов.

Пусть первому вхождению фрагмента **F** в алгоритм соответствует пара $(\mathbf{s,t})$ индексов, а второму – $(\boldsymbol{l},\mathbf{r})$. Тогда в первом случае (при последовательном выполнении двух фрагментов одним устройством) должно иметь место

$$\boldsymbol{l} = \mathbf{s} \text{ и } \left| \mathbf{r - t \geq \tau} \right|, \tag{3}$$

во втором (при параллельном выполнении фрагментов двумя устройствами)

$$\left| \boldsymbol{l} - \mathbf{s} \geq \boldsymbol{\sigma} \right| \quad \text{и } \mathbf{r = t}. \tag{4}$$

Введем далее формальный математический аппарат, позволяющий оперировать с алгоритмами и устройствами на уровне **ST**-моделей, в основу которого положены обычные теоретико-множественные операции. Чтобы не усложнять дальнейшее изложение преобразованиями с индексами, введем условное обозначение

$$\mathbf{F_{s,t}(X) = F[\,\{s\} \times \{t\}\,](X)}.$$

Тогда при выполнении (3) будем иметь

$$\mathbf{F[\,\{s\} \times \{t\}](X) \cup F[\,\{s\} \times \{r\}](X) = F[\,\{s\} \times \{t,r\}](X)}, \tag{5}$$

а при выполнении (4) получим

$$\mathbf{F[\,\{s\} \times \{t\}](X) \cup F[\,\{l\} \times \{t\}](X) = F[\,\{s,l\} \times \{t\}](X)}. \tag{6}$$

Таким образом, выражение (5) описывает процесс выполнения двух фрагментов алгоритма одним и тем же устройством последовательно во времени, выражение (6) – двумя идентичными устройствами параллельно. Из (5) и (6) следует, что функция **F** в качестве оператора, описывающего **ST**-модель, с теоретико-множественной операцией объединения множеств подчиняется дистрибутивному закону.

Пусть далее задана некоторая последовательность

$$(\mathbf{s_0,t_0}),(\mathbf{s_1,t_1}),(\mathbf{s_2,t_2})\cdots,(\mathbf{s_q,t_q}) \tag{7}$$

пар $(\mathbf{s_j,t_j})$ для $\mathbf{0 \leq j \leq q}$, для которых для всех $\mathbf{0 \leq j \leq q}$ выполняется хотя бы одно из двух условий:

$$\mathbf{s_{j+1} \geq s_j + \sigma} \text{ и } \mathbf{t_{j+1} = t_j} \quad \text{или} \quad \mathbf{s_{j+1} = s_j} \text{ и } \mathbf{t_{j+1} \geq t_j + \tau},$$

где $\mathbf{q}$ - некоторое натуральное число. Из (1) следует, что последовательность

(7) существует. В дальнейшем будем предполагать, что каждый бит любой переменной определен на множестве пар типа (7) для любых $\mathbf{q} \in \mathbf{N}$. Далее будем часто использовать обозначение $\mathbf{C}[\mathbf{s}_j \mathbf{x} \ \mathbf{t}_j] = (\mathbf{s}_j, \mathbf{t}_j)$. Из (6) следует, что в соответствии с дистрибутивным законом задана новая функция

$$\mathbf{Y} = \mathbf{F} \bigcup_{\mathbf{j}=0}^{\mathbf{q}} \left[\{ \mathbf{s}_j \} \times \{ \mathbf{t}_j \} \right] (\mathbf{X}) = \bigcup_{\mathbf{j}=0}^{\mathbf{q}} \left[\{ \mathbf{s}_j \} \times \{ \mathbf{t}_j \} \right] \mathbf{F}(\mathbf{X}), \tag{8}$$

которой может быть сопоставлена новая **ST**-модель, соответствующая некоторому новому устройству, реализующая функцию **F**. Следовательно, существует метод синтеза **ST**-модели, соответствующей устройству, реализующему любой алгоритм, состоящий из фрагментов с идентификатором **F**.

В целях упрощения записи условимся обозначать $[\mathbf{a}]\mathbf{F}(\mathbf{X})$ вместо $[\{\mathbf{a}\}]\mathbf{F}(\mathbf{X})$. Тогда (8) будет представлено более компактным выражением:

$$\mathbf{Y} = \mathbf{F} \bigcup_{\mathbf{j}}^{\mathbf{q}} [\mathbf{s}_j \times \mathbf{t}_j] (\mathbf{X}) = \bigcup_{\mathbf{j}=0}^{\mathbf{q}} [\mathbf{s}_j \times \mathbf{t}_j] \mathbf{F}(\mathbf{X}). \tag{9}$$

В качестве стандартных операторов, задающих **ST** - модели, участвующие в формальных преобразованиях в процессе синтеза **ST** - моделей будут использоваться следующие.

1. $\mathbf{W} = \mathbf{L}[\mathbf{G} \times \mathbf{H}](\mathbf{X})$ – оператор выделения, который определяется как

$$\mathbf{w} \in \mathbf{W} \equiv (\mathbf{w} = \mathbf{X}[\mathbf{a} \times \mathbf{b}]) \& (\mathbf{a} \in \mathbf{G}) \& (\mathbf{b} \in \mathbf{H}),$$

где $\mathbf{G} \subset \mathbf{N}, \mathbf{H} \subset \mathbf{N}$.

В качестве частного случая такого оператора будет применяться

$$\mathbf{W} = \mathbf{L}[\mathbf{G}](\mathbf{X}) \equiv \mathbf{L}[\mathbf{G} \times \mathbf{N}](\mathbf{X}), \tag{10}$$

где $\mathbf{N}$ – множество натуральных чисел.

Стандартный оператор выделения в общем виде реализуется набором обычных двухвходовых схем "И", один из входов каждой из которых соединяется с $\mathbf{j}$-й компонентой вектора $\mathbf{X}$ для всех $\mathbf{j} \in \mathbf{G}$, а второй соответствует сигналу из последовательности сигналов временной диаграммы, по времени определяющим каждую временную проекцию $\mathbf{t} \in \mathbf{H}$. Если оператор выделения представлен выражением (10), то он не требует для реализации специальных схем и реализуется набором обычных связей, соответствующих первым входам описанных выше схем "И".

2. $\mathbf{W} = \mathbf{P}^{\mathbf{a}}(\mathbf{X})$ – оператор пространственного сдвига, определяемый как

$$\mathbf{w} \in \mathbf{W} \equiv \exists \mathbf{X}(\mathbf{b} \times \mathbf{d}) \to \mathbf{w} = \mathbf{W}[(\mathbf{b} + \mathbf{a}) \times \mathbf{d}]$$

для любых целых чисел $\mathbf{a}$.

3. $\mathbf{W} = \mathbf{D}^{\mathbf{c}}(\mathbf{X})$ – оператор временного сдвига, определяемый выражением

$$\mathbf{w} \in \mathbf{W} \equiv \exists \mathbf{X}(\mathbf{b} \times \mathbf{d}) \to \mathbf{w} = \mathbf{W}[\mathbf{b} \times (\mathbf{d} + \mathbf{c})]$$

для любых целых неотрицательных чисел $\mathbf{c}$.

Оператор пространственного сдвига, как и оператор (10), также не требует для реализации специальных схем и служит для задания конкретных связей между входными и выходными каналами схем, соответствующих компонентам векторов в реализуемом алгоритме. В отличие от оператора пространственного сдвига оператор временного сдвига осуществляет временную задержку сигналов на время, указанное параметром $\mathbf{c}$ и может быть схемно реализован различными способами. В частном случае это может быть обычный **D**-триггер, либо любой другой элемент памяти, для которого сигнал появляется на его выходе спустя $\mathbf{c}$ дискретных интервалов относительно момента, когда он появился на его входе.

В дополнение к (9) введем еще ряд тождественных преобразований, встречающихся в процессе синтеза **ST**-моделей.

$$\mathbf{P^a\ D^c\ F[H](X) = F\left[\ P^aD^cH\ \right]\left(P^aD^cX\right)}. \tag{11}$$

Отсюда следует

$$\mathbf{P^a\ D^c\ L[H](X) = L\left[\ P^aD^cH\ \right]\left(P^aD^cX\right)}. \tag{12}$$

Естественно, что всегда будет иметь место:

$$\mathbf{P^aD^c\left(P^bD^d\ (X)\right) = P^{a+b}\ D^{c+d}\ (X) = P^b\ D^d\left(P^aD^cX\right)}; \tag{13}$$

$$\mathbf{L[G]\big(\ L[H](X)\ \big) = L\left[\ G \cap H\ \right](X)}. \tag{14}$$

Из (9) следует

$$\mathbf{F\left[\ s_j \times t_j\ \right](X) = L\left[\ s_j \times t_j\ \right](Y)} \tag{15}$$

Пусть в алгоритме встречается $\mathbf{q}$ различных фрагментов с идентификатором **F**, нумерация которых начинается с нуля (что, вообще говоря, не является существенным). Тогда всегда существует множество последовательностей (7), каждая из которых определяет **ST**-модель, соответствующую устройству, реализующую все фрагменты алгоритма с идентификатором **F**. Исходя из полученных результатов, нетрудно сформулировать общую методику формального нахождения структуры устройства, реализующего заданный алгоритм.

Будем считать, что алгоритм, подлежащий схемной реализации, содержит $\boldsymbol{l}$ переменных $\mathbf{x_j}$, где $\mathbf{0\ \geq\ j\ \geq\ \boldsymbol{l}\ -1}$, представляющих собой исходные данные алгоритма и $\mathbf{r}$ промежуточных переменных $\mathbf{y_k}$ для всех $\mathbf{0 \leq k \leq r-1}$, являющихся результатами вычисления определенных функций в любых фрагментах алгоритма. В соответствии с введенными выше определениями операторов представим все исходные данные алгоритма единственной переменной $\mathbf{X}$:

$$\mathbf{X = \bigcup_{k=0}^{\boldsymbol{l}-1}\left\{P^{\gamma(k)}\left(x_k\right)\right\}},$$

где $\mathbf{h(k)}$ – разрядность каждой переменной $\mathbf{x_k}$ в битах; $\mathbf{\gamma(0) = 0}$;

$\gamma(\mathbf{k}) = \sum_{\mathbf{i}=0}^{\mathbf{k}-1} \mathbf{h}(\mathbf{i})$ для $\mathbf{k} \ge 1$. Следовательно, для $\mathbf{0} \le \mathbf{i} \le \mathbf{h}(\mathbf{k}) - 1$

$$\mathbf{L}[\,\mathbf{i}\,](\,\mathbf{x_k}\,) = \mathbf{L}[\,\gamma(\mathbf{k}) + \mathbf{i}\,](\mathbf{X}). \qquad (16)$$

Пусть переменная $\mathbf{y_k}$ определена в алгоритме фрагментом с идентификатором $\mathbf{F_j}$, т.е.

$$\mathbf{y_k} = \mathbf{F_j}\,(\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}).$$

где $\mathbf{F_j} : \mathbf{A}^{\mathbf{m}(\mathbf{j})} \to \mathbf{A}^{\mathbf{n}(\mathbf{j})}$, $\mathbf{m}(\mathbf{j})$ – общее количество бит, содержащихся во всех аргументах функции, обозначенной символом $\mathbf{F_j}$; $\mathbf{n}(\mathbf{j})$ - количество бит в любой переменной, являющейся результатом вычисления функции, обозначенной символом $\mathbf{F_j}$. Естественно, что данному фрагменту взаимно однозначно соответствует $\mathbf{ST}$-модель

$$\mathbf{y_k} = \mathbf{F_j}\,[\,\mathbf{0} \times \mathbf{0}\,](\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}).$$

Если эта функция существенно зависит от всех аргументов, указанных в скобках, то должно иметь место $\mathbf{l} + \mathbf{r} \le \mathbf{m}(\mathbf{j})$. В этом случае элементы множества $\{\,\mathbf{0}, \mathbf{1}, \ldots, \mathbf{m}(\mathbf{j}) - \mathbf{1}\,\}$ должны взаимно однозначно соответствовать элементам множества $\{\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}\}$. Как уже было отмечено ранее, в $\mathbf{ST}$-моделях каждая переменная определена в едином пространстве и в едином времени. Следовательно, каждой переменной $\mathbf{z} \in \{\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}\}$ должно быть соотнесено некоторое натуральное число, определяющее её положение в пространстве. Так, если переменной $\mathbf{y_1} \in \{\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}\,\}$ соответствует число $\mathbf{c}$, а в множестве $\{\,\mathbf{0}, \mathbf{1}, \ldots, \mathbf{m}(\mathbf{j}) - \mathbf{1}\,\}$ согласно определению функции с идентификатором $\mathbf{F_j}$ в определении $\mathbf{ST}$-модели ей соответствует число $\mathbf{d} \in \{\mathbf{0}, \mathbf{1}, \cdots \mathbf{m}(\mathbf{j}) - \mathbf{1}\}$ (это обозначает, что в эталонной записи функции с идентификатором $\mathbf{F_j}$ соответствующему аргументу сопоставлена позиция с номером $\mathbf{d}$), то на месте символа $\mathbf{y_1}$, обозначающего эту переменную, в записи $\mathbf{y_k} = \mathbf{F_j}\,(\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}})$ должно стоять выражение $\mathbf{P}^{-\mathbf{c}+\mathbf{d}}\mathbf{y_1}$. Нетрудно заметить, что каждому оператору $\mathbf{F_j}$ соответствует множество $\mathbf{ST}$-моделей. Чтобы избежать неоднозначности, будем считать, что в первом приближении оператору

$$\mathbf{y_k} = \mathbf{F_j}\,(\,\mathbf{x_0}, \ldots, \mathbf{x_{l-1}}, \mathbf{y_0}, \ldots, \mathbf{y_{r-1}}\,)$$

соответствует $\mathbf{ST}$-модель, заданная выражением

$$\mathbf{y_k = F_j\left[\,0\times 0\,\right]\left(x_0, \ldots, x_{l-1}, y_0, P^{-c+d} y_1, \ldots, y_{r-1}\right)}.$$

Обозначим далее $\mathbf{M_j}$ множество всех индексов $\mathbf{k}$ переменных $\mathbf{y_k}$, которые определяются в алгоритме фрагментами с идентификатором $\mathbf{F_j}$. Пусть в алгоритме встречается $\mathbf{q}$ идентификаторов $\mathbf{F_j}$, определяющих различные множества $\mathbf{M_j}$. Условимся, что для всех $\mathbf{1 \le j \le q}$ имеет место $\mathbf{0 \in M_j}$, $\lambda(\mathbf{0}) = (\mathbf{0,0})$ и $\eta(\mathbf{0}) = (\mathbf{0,0})$. Тогда в соответствии с (9) для каждого $\mathbf{j}$, для которого имеет место $\mathbf{1 \le j \le q}$, можно ввести новую **ST**-модель и соответствующую ей переменную

$$\mathbf{Y_j = \bigcup_{k \in M_j} P^{a(k,j)} D^{b(k,j)} F_j\left[0\times 0\right]\left(x_0, \ldots, x_{l-1}, y_0, \ldots, y_{r-1}\right)}, \qquad (17)$$

где $\mathbf{a(k,j) = s_k}$, $\mathbf{b(k,j) = t_k}$, а пара $(\mathbf{s_k, t_k})$ – соответствующий член одной из последовательностей (7).

Тогда из (12) следует, что для каждого $\mathbf{1 \le j \le q}$ и каждого $\mathbf{k \in M_j}$

$$\mathbf{y_k = L\left[\,a(k,j)\times b(k,j)\,\right]\left(Y_j\right)}. \qquad (18)$$

Описанным выше способом весь алгоритм, подлежащий схемной реализации, в соответствии с общим числом различных фрагментов в алгоритме может быть формальными преобразованиями сведен к $\mathbf{q}$ выражениям (17), в которых каждая переменная в правой части может быть представлена либо (16), либо (18). Таким образом, каждому алгоритму может быть поставлена в соответствие **ST**-модель, соответствующая определенному устройству, реализующему этот алгоритм. Поскольку в (17) фигурируют только операторы, техническая реализация которых была описана выше, то переход к функциональной схеме соответствующего устройства будет тривиален.

ЛИТЕРАТУРА

1. *Глушков В.М., Капитонова Ю.В., Летичевский А.А. Автоматизация проектирования вычислительных машин. – К.: Наук. думка, 1975. - 228 с.*
2. *Кондратьев В. Н., Богатов О.И., Кондратьева Е.В. О возможностях учёта алгоритмических особенностей задач // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. –1999. – Вип. 1(6). – С. 124 - 127.*
3. *Армстронг. Дж. Моделирование цифровых схем на языке VHDL. – М.: Мир, 1992. – 171 с.*

Поступила 22.07.2002

БОГАТОВ Олег Иванович*, канд. техн. наук, ст. научн. сотр., начальник НИО научного центра при ХВУ. В 1983 окончил Киевское ВИРТУ. Область научных интересов – параллельная обработка информации, САПР.*

НИЗОВЦЕВ Андрей Валерьевич, *нач. лаб. ИВЦ ХВУ. В 1986 году окончил Харьковское ВВКИУ РВ. Область научных интересов – ассоциативный поиск неявной информации.*

КОНДРАТЬЕВА Елена Владимировна, *инженер ИВЦ ХВУ. В 1998 году окончила*

ХГПУ. Область научных интересов – системы автоматизированного проектирования.

КОНДРАТЬЕВ Владимир Николаевич, *инженер ИВЦ ХВУ. В 1961 году окончил ХПИ. Область научных интересов – системы автоматизированного проектирования.*

ПРОКОПЕНКО Игорь Иванович*, нач. лаборатории ИВЦ ХВУ. В 1987 году окончил ПВУРЭ. Область научных интересов – аппаратно-программный контроль вычислительных комплексов.*