

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 устрой-

ству, 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 вентилях 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 $(A, F, \lambda, \eta, \sigma, \tau)$, где A – алфавит (обычно $A = \{0, 1\}$); $F: A^m \rightarrow A^n$ – вычисляемая функция для целых $m \geq 1$, $n \geq 1$; $\lambda: \{0, 1, \dots, m-1\} \rightarrow \{s, s+1, \dots, s+\sigma_0-1\} \times \{t, t+1, \dots, t+\tau_0-1\}$ – входное биективное отображение, s, t, σ_0, τ_0 – некоторые натуральные числа; $\eta: \{0, 1, \dots, n-1\} \rightarrow \{p, p+1, \dots, p+\sigma_1-1\} \times \{q, q+1, \dots, q+\tau_1-1\}$ – выходное биективное отображение, p, q, σ_1, τ_1 – некоторые натуральные числа, $\sigma \geq \max(\sigma_0, \sigma_1)$ – константа относительного пространственного сдвига данных; $\tau \geq \max(\tau_0, \tau_1)$ – константа относительного временного сдвига данных. Символы λ и η для определенности иногда будем обозначать $\lambda_{s,t}$ и $\eta_{p,q}$.

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

$$\sigma_0 \cdot \tau_0 \geq m, \quad \sigma_1 \cdot \tau_1 \geq n. \quad (1)$$

С содержательной точки зрения это определение обозначает, что ST-модель переводит вычисляемую элементом любую функцию $Y = F(X)$ для $X \in A^m, Y \in A^n$ из одномерной плоскости в последовательной записи не-

которого алгоритма во взаимно однозначно соответствующую ей функцию

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

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, \mathbf{ST} -модель соответствует преобразованию данных в пространственно-временной области, что соответствует реальному вычислительному процессу.

Покажем, что \mathbf{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 $\mathbf{X} \in \mathbf{A}^m$, $\mathbf{Y} \in \mathbf{A}^n$. Из (2) и определения \mathbf{ST} -модели следует:

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

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

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

Функция $\mathbf{F} : \mathbf{A}^m \rightarrow \mathbf{A}^n$ 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 \mathbf{F} . С другой стороны, каждому фрагменту взаимно однозначно соответствует некоторая \mathbf{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 \mathbf{F} . Согласно определению фрагмента ему соответствует некоторое устройство, которое может реально вычислять соответствующую функцию, определённые черты которой отражены в соответствующей ему \mathbf{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 \mathbf{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 \mathbf{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 \mathbf{F} в качестве идентификатора \mathbf{ST} -модели процесса преобразования данных во всех устройствах, которые физически реализуют функции $\mathbf{F}: \mathbf{A}^m \rightarrow \mathbf{A}^n$ с любыми исходными данными в едином дискретном пространстве и едином дискретном времени. Таким образом, \mathbf{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 \mathbf{F} , взятом в качестве идентификатора соответствующих фрагментов.

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

$$l = s \text{ и } |r - t| \geq \tau, \quad (3)$$

во втором (при параллельном выполнении фрагментов двумя устройствами)

$$|l - s| \geq \sigma \text{ и } r = t. \quad (4)$$

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

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

Тогда при выполнении (3) будем иметь

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

а при выполнении (4) получим

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

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 \mathbf{F} в качестве оператора, описывающего \mathbf{ST} -модель, с теоретико-множественной операцией объединения множеств подчиняется дистрибутивному закону.

Пусть далее задана некоторая последовательность

$$(\mathbf{s}_0, \mathbf{t}_0), (\mathbf{s}_1, \mathbf{t}_1), (\mathbf{s}_2, \mathbf{t}_2) \dots, (\mathbf{s}_q, \mathbf{t}_q) \quad (7)$$

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

$$\mathbf{s}_{j+1} \geq \mathbf{s}_j + \sigma \text{ и } \mathbf{t}_{j+1} = \mathbf{t}_j \quad \text{или} \quad \mathbf{s}_{j+1} = \mathbf{s}_j \text{ и } \mathbf{t}_{j+1} \geq \mathbf{t}_j + \tau,$$

где 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 $\mathbf{C}[s_j \times t_j] = (s_j, t_j)$. Из (6) следует, что в соответствии с дистрибутивным законом задана новая функция

$$\mathbf{Y} = \mathbf{F} \bigcup_{j=0}^{\mathbf{q}} [\{s_j\} \times \{t_j\}] (\mathbf{X}) = \bigcup_{j=0}^{\mathbf{q}} [\{s_j\} \times \{t_j\}] \mathbf{F} (\mathbf{X}), \quad (8)$$

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

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

$$\mathbf{Y} = \mathbf{F} \bigcup_j^{\mathbf{q}} [s_j \times t_j] (\mathbf{X}) = \bigcup_{j=0}^{\mathbf{q}} [s_j \times t_j] \mathbf{F} (\mathbf{X}). \quad (9)$$

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

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

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 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

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

для любых целых чисел \mathbf{a} .

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

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

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 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 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.

$$P^a D^c F [H](X) = F [P^a D^c H] (P^a D^c X). \quad (11)$$

Отсюда следует

$$P^a D^c L [H](X) = L [P^a D^c H] (P^a D^c X). \quad (12)$$

Естественно, что всегда будет иметь место:

$$P^a D^c (P^b D^d (X)) = P^{a+b} D^{c+d} (X) = P^b D^d (P^a D^c X); \quad (13)$$

$$L [G](L [H](X)) = L [G \cap H] (X). \quad (14)$$

Из (9) следует

$$F [s_j \times t_j] (X) = L [s_j \times t_j] (Y) \quad (15)$$

Пусть в алгоритме встречается 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 l переменных x_j , где $0 \geq j \geq l-1$, представляющих собой исходные данные алгоритма и r промежуточных переменных y_k для всех $0 \leq k \leq r-1$, являющихся результатами вычисления определенных функций в любых фрагментах алгоритма. В соответствии с введенными выше определениями операторов представим все исходные данные алгоритма единственной переменной X :

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

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

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

$$L[i](x_k) = L[\gamma(k) + i](X). \quad (16)$$

Пусть переменная y_k определена в алгоритме фрагментом с идентификатором F_j , т.е.

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

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

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

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

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

соответствует ST-модель, заданная выражением

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

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

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

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

Тогда из (12) следует, что для каждого $1 \leq j \leq q$ и каждого $k \in M_j$

$$y_k = L [a(k, j) \times b(k, j)] (Y_j). \quad (18)$$

Описанным выше способом весь алгоритм, подлежащий схемной реализации, в соответствии с общим числом различных фрагментов в алгоритме может быть формальными преобразованиями сведен к 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 научно-го 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 ВВКШУ 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.
