

Теоретико-множественное представление сложно-функциональных блоков в системе автоматизированного проектирования интегральных схем на основе реконфигурируемых систем на кристалле

В. М. Хватов

Институт проблем проектирования в микроэлектронике РАН, Москва

Аннотация: В статье представлена программно-ориентированная теоретико-множественная модель сложно-функциональных блоков (СФ-блоков) реконфигурируемой системы на кристалле (РСнК), позволяющая ввести их в систему автоматизированного проектирования (САПР) на разных этапах проектирования интегральной схемы на основе РСнК и выполнить отображение архитектурно-независимых СФ-блоков проектируемой схемы в СФ-блоки базового кристалла. Модель объединяет в себе формализованные представления компонентов базового кристалла, архитектурно-независимых и архитектурно-зависимых параметризованных СФ-блоков для этапа логического синтеза, а также СФ-блоков для этапа топологического синтеза. Также модель имеет несколько отличительных особенностей. Первой особенностью является наличие относительных координат элементов гибких СФ-блоков, что позволяет учесть архитектурные особенности РСнК на этапе размещения. Вторая особенность - информация о специализированных трассировочных ресурсах и межсоединениях проектируемой и базовой схем, что дает возможность установить соответствие между соединениями СФ-блока и РСнК независимо от типов трассировочных ресурсов.

Ключевые слова: реконфигурируемая система на кристалле, сложно-функциональный блок, программируемая логическая интегральная схема, система автоматизированного проектирования, теоретико-множественная модель, формализация.

Введение

В настоящее время широкое применение получают отечественные реконфигурируемые системы на кристалле (РСнК). Они, как правило, состоят из набора заказных жестких сложно-функциональных блоков (СФ-блоков) и программируемой логической интегральной схемы (ПЛИС) [1].

Жёсткими СФ-блоками из базиса РСнК являются блоки, имеющие заранее определённые позиции на кристалле, и выполняющие ряд логических функций, определяющих режимы работы блока. Программирование требуемого режима работы жестких СФ-блоков, в зависимости от архитектуры, может выполняться статически, за счет подачи сигналов из

конфигурационной памяти на управляющие контакты, или динамически, за счет передачи данных на информационные контакты [2].

Другая составная часть РСнК – ПЛИС – представляет собой регулярную матрицу конфигурируемых логических блоков (ЛБ) и множество коммутационных элементов, обеспечивающих соединения между ними [3,4]. Стандартные ячейки, являющиеся запрограммированными ЛБ ПЛИС, в совокупности с жёсткими СФ-блоками представляют библиотечный базис РСнК, который является основой для проектируемой схемы. Базис может быть расширен за счет добавления гибких СФ-блоков, которые, в отличие от жестких, состоят из ЛБ, не имеют заранее известной позиции на кристалле и сконфигурированных межсоединений [5]. Для более эффективного результата логического и физического синтеза данного типа СФ-блоков в архитектуре ПЛИС содержатся специализированные ресурсы, позволяющие повысить частоту их работы, а также сократить количество используемых ЛБ и коммутационных элементов [6].

Все особенности жестких и гибких СФ-блоков необходимо учесть при их интеграции в систему автоматизированного проектирования (САПР) интегральных схем (ИС) на основе базовой РСнК. Для этого необходим программно-ориентированный математический аппарат

Существующие модели, используемые в САПР зарубежных производителей [7,8] и в методах для выполнения определенных этапов проектирования, либо разработаны под конкретную технологию или архитектуру [9,10] и не применимы к новым РСнК, либо не содержат требуемой информации, которая позволила бы управлять архитектурными особенностями и специализированными ресурсами базовой схемы [11,12].

В связи с этим, была разработана обобщенная теоретико-множественная модель параметризованных программируемых СФ-блоков, ориентированная на практическое применение в САПР. Она объединяет в

себе представление СФ-блоков из базовой схемы, разрабатываемой схемы и из библиотек САПР для различных этапов маршрута проектирования.

Основные составляющие теоретико-множественной модели СФ-блоков

В состав разработанной модели входит ряд основных множеств СФ-блоков:

- 1) Y – множество архитектурно-независимых логических элементов и параметризованных СФ-блоков из библиотеки для логического синтеза;
- 2) T – вспомогательное множество параметризованных СФ-блоков для технологического отображения архитектурно-независимых СФ-блоков проектируемой схемы в архитектурно-зависимые;
- 3) V – множество параметризованных СФ-блоков для топологического синтеза и отображения архитектурно-зависимых блоков в подсхемы целевого кристалла.

На основе представленных множеств СФ-блоков введены следующие представления проектируемой схемы:

- 1) c_y – схема после архитектурно-независимого логического синтеза,
- 2) c_t – схема после логического синтеза и до технологического отображения архитектурно-независимых СФ-блоков в архитектурно-зависимые,
- 3) c_b – схема после отображения архитектурно-независимых СФ-блоков схемы в архитектурно-зависимые.

Формализованное представление базовой РСнК

Все подсхемы, составляющие базовую РСнК в схемотехническом описании представлены множеством S . Из данного множества выделены подсхемы, участвующие в процессе проектирования разрабатываемой схемы: S_{LE} – логические блоки, S_{IO} – ячейки ввода/вывода, S_M – жесткие СФ-блоки,

C_{RT} – трассировочные элементы и c_f – подсхема верхнего уровня: $C = \{C_{LE} \cup C_{IO} \cup C_M \cup C_{RT} \cup c_f\}$.

Подсхема верхнего уровня c_f характеризуется следующими структурными компонентами: $c_f = (\mu(c_f), E(c_f), P(c_f), N(c_f))$, где $\mu(c_f)$ – имя подсхемы, $E(c_f) = \{e_f\}$ – элементы подсхемы; $P(c_f) = \{p_f\}$ – внешние контакты; $N(c_f) = \{n_f\}$ – межсоединения. На основе того, что внутреннее содержимое подсхем $C_{LE} \cup C_{IO} \cup C_M \cup C_{RT}$ скрыто от внешнего пользователя, они представлены исключительно именем и внешними контактами.

Элементы c_f характеризуются следующими компонентами: $\forall e_f \in E(c_f)$: $e_f = (\mu(e_f), m_c(e_f), P(e_f), r(e_f))$, где $\mu(e_f)$ – имя элемента; $m_c(e_f) \in C / c_f$ – модель элемента базовой РСНК; $P(e_f) = \{p_x\}$ – множество контактов элемента; $r(e_f)$ – координаты ЛБ. Если $m_c(e_f) \in C_{LE}$, то $r(e_f) \neq \emptyset$.

Контакты элемента e_f представлены следующими компонентами: $\forall p_x \in P(e_f)$: $p_x = (\mu(p_x), d(p_x), f(p_x))$, где $\mu(p_x)$ – имя контакта, $d(p_x) \in \{d_{inp}, d_{out}, d_{inout}\}$ – направление контакта: входной, выходной или двунаправленный, $f(p_x) \in \{f_{rstd}, f_{rsp}, f_m\}$ – функциональное назначение контакта: стандартный коммутационный, специальный коммутационный или конфигурационный. В соответствии с назначением можно выделить следующие типы контактов: конфигурационные $P_m(e_f)$, необходимые для выбора режима подсхемы; коммутационные стандартные $P_{rstd}(e_f)$, подключенные к трассировочным ресурсам общего назначения и предназначенные для конфигурирования подсхем под стандартный элемент библиотеки логических элементов; коммутационные специализированные $P_{rsp}(e_f)$, подключенные к трассировочным ресурсам специального назначения, позволяющим снизить количество ЛЭ в схеме, увеличить её быстродействие и снизить объем занятых трассировочных ресурсов. Последний тип контактов предназначен для конфигурирования подсхем под библиотечные элементы, требуемые для

проектирования гибких СФ-блоков. Первый и второй типы контактов входят в состав подсхем C / c_f , третий тип входит только в состав ЛБ.

Началом отсчета координат $r(e_f)$ элементов регулярной матрицы ПЛИС в РСнК выбран левый нижний ЛБ. Координаты состоят из номера строки и столбца ЛБ в РСнК: если $e_f \in E(c_f)$, $m_c(e_f) \in C_{LE}$, то $r(e_f) = (r_l, r_c)$, где r_l – номер строки, r_c – номер столбца матрицы ЛБ РСнК, $r_l = 1, \dots, M_l$, $r_c = 1, \dots, M_c$, где M_l – число строк, M_c – число столбцов ЛБ в РСнК. Информация о координатах не задана в схемотехническом описании, но присутствует в графическом представлении РСнК или её топологии.

Межсоединения подсхемы c_f представлены следующими составляющими: $\forall n_f \in N(c_f): n_f = (\mu(n_f), p_x^s(n_f), \{p_x^d(n_f)\})$, где $\mu(n_f)$ – имя межсоединения, $p_x^s(n_f)$ – контакт типа «источник», являющийся выходом элемента подсхемы, $\{p_x^d(n_f)\}$ – множество контактов типа «приемник», являющихся входами подсхемы. Если $p_x^s(n_f)$ и $n_f \in N(c_f)$, то $d(p_x^s) = d_{out} \mid d_{inout}$, если $p_x^d(n_f)$ и $n_f \in N(c_f)$, то $d(p_x^d) = d_{in} \mid d_{inout}$.

Формализованное представление проектируемой схемы

Множество логических элементов и СФ-блоков внутренней библиотеке программы для логического синтеза, не зависящих от архитектуры целевого кристалла, обозначено через Y . Элементы данного множества характеризуются следующими компонентами: $\forall y \in Y: y = (\mu(y), \tau(y), \hat{\tau}(y), P(y), O(y))$, где $\mu(y)$ – имя библиотечного элемента y , $\tau(y)$ – тип элемента, $\hat{\tau}(y)$ – подтип элемента, $\tau(y) \in \{\tau_{le}, \tau_{macro}\}$, где τ_{le} – стандартный комбинационный элемент или триггер, τ_{macro} – СФ-блок; $O(y)$ – параметры элемента, $P(y)$ – внешние контакты. На основе типа элемента формируются следующие подмножества: $Y = Y_{le} \cup Y_{macro}$. При этом, если $\tau(y) = \tau_{macro}$, то $y \in Y_{macro}$. Подтип СФ-блока определяется в соответствии с его функциональным назначением. Например, в состав РСнК и ПЛИС могут входить СФ-блоки

следующих подтипов: умножители ($\hat{\tau}_{mul}$), сумматоры ($\hat{\tau}_{add}$), многоразрядные счётчики ($\hat{\tau}_{cnt}$), блоки памяти ($\hat{\tau}_{ram}$), процессоры обработки сигналов ($\hat{\tau}_{dsp}$). В соответствии с этим, если элемент имеет тип $\tau(y) = \tau_{macro}$, то $\hat{\tau}(y) \in \{\hat{\tau}_{mul}, \hat{\tau}_{add}, \hat{\tau}_{cnt}, \hat{\tau}_{ram}, \hat{\tau}_{dsp}\}$. Если тип элемента $\tau(y) = \tau_{le}$, то элемент может быть комбинационным элементом ($\hat{\tau}_{comb}$) или триггером ($\hat{\tau}_{dff}$): $\hat{\tau}(y) \in \{\hat{\tau}_{comb}, \hat{\tau}_{dff}\}$.

Параметры библиотечного элемента $O(y)$ необходимы для технологического отображения элемента внутренней архитектурно-независимой библиотеки в требуемый элемент, учитывающий архитектуру РСнК. Параметры $O(y)$ представлены следующими компонентами: $\forall o \in O(y): o = (\mu(o), Y(o))$, где $\mu(o)$ – имя параметра, $Y(o) = \{v\}$ – его значения.

Проектируемая схема после логического синтеза и до технологического отображения обозначена через c_y и является списком соединений без иерархии, который можно описать компонентами, аналогичными c_f : $c_y = (\mu(c_y), E(c_y), P(c_y), N(c_y))$. Отличие заключается в представлении элементов схемы $E(c_y)$: $\forall e_y \in E(c_y): e_y = (\mu(e_y), m_y(e_y), O(e_y), P(e_y))$. Модель элементов $m_y(e_y) \in Y$. Элементы в c_y , в отличие от c_f , не имеют координат, но имеют параметры, заданные после логического синтеза: $O(e_y) = \{o\}$.

Параметры элемента $O(e_y)$ представлены парой компонентов: $\forall o \in O(e_y): o = (\mu(o), v(o))$, где $\mu(o)$ – имя параметра, $v(o)$ – заданное значение параметра. Контакты элемента $\forall p_z \in P(e_y)$ и контакты проектируемой схемы $\forall p_y \in P(c_y)$ представлены компонентами, аналогичными компонентам контактов $P(c_f)$. Межсоединения проектируемой схемы $\forall n_y \in N(c_y)$ представлены компонентами, аналогичными компонентам c_f .

Отображение СФ-блока из архитектурно-независимой библиотеки в СФ-блок РСнК имеет следующий вид: $\{m_y(e_y): m_y(e_y) \in Y, e_y \in E(c_y)\} \rightarrow$

$\{m_c(e_f): m_c(e_f) \in C, e_f \in E(c_f)\}$. Компоненты представленной модели позволяют выполнить отображение в три этапа:

1) отображение архитектурно-независимых моделей СФ-блоков проектируемой схемы из Y в модели из вспомогательной библиотеки для технологического отображения $T: \{m_y(e_y): m_y(e_y) \in Y, e_y \in E(c_y)\} \rightarrow \{m_t(e_t): m_t(e_t) \in T, e_t \in E(c_t)\}$;

2) отображение моделей полученных СФ-блоков в архитектурно-зависимые модели из библиотеки для физического синтеза $B: \{m_t(e_t): m_t(e_t) \in T, e_t \in E(c_t)\} \rightarrow \{m_b(e_b): m_b(e_b) \in B, e_b \in E(c_b)\}$;

3) отображение в модели СФ-блоков из РСнК: $\{m_b(e_b): m_b(e_b) \in B, e_b \in E(c_b)\} \rightarrow \{m_c(e_f): m_c(e_f) \in C, e_f \in E(c_f)\}$.

Схемы c_t и c_b характеризуется компонентами, которые аналогичны составляющим схемы c_y .

Формализованное представление СФ-блоков из библиотеки для технологического отображения

СФ-блок из множества T представлен в модели следующими компонентами: $\forall t \in T: t = (\mu(t), \tau^{\wedge}(t), O(t), P(t), U(t), N(t))$, где $\mu(t)$ – имя СФ-блока, совпадающее с именем СФ-блока $y \in Y$: если $t \in T, y \in Y_{macro}$ и $\tau^{\wedge}(t) = \tau^{\wedge}(y)$, то $\mu(t) = \mu(y)$; $\tau^{\wedge}(t)$ – подтип СФ-блока. $O(t) = \{o_t\}$ – параметры СФ-блока идентичные параметрам $O(y), y \in Y_{macro}$; $P(t) = \{p_t\}$ – внешние контакты; $U(t) = \{u_t\}$ – варианты отображения СФ-блока; $N(t) = \{n_t\}$ – межсоединения.

Контакты СФ-блока $P(t)$ характеризуются именем и направлением. При этом, если $t, y, t \in T, y \in Y_{macro}, \tau^{\wedge}(t) = \tau^{\wedge}(y)$, то $P(t) \Leftrightarrow P(y)$.

Варианты отображения СФ-блока t представлены следующими компонентами: $\forall u_t \in U(t): u_t = (m_b(u_t), O(u_t), P(u_t))$, где $m_b(u_t) \in B$ – модель элемента из множества СФ-блоков для физического синтеза, $O(u_t)$ – параметры для технологического отображения, $P(u_t)$ – контакты элемента.

Если параметризованная модель элемента СФ-блока $m_b(u_t) \in B$ не имеет фиксированных параметров, $O(m(u_t)) \Leftrightarrow O(t)$, $t \in T$, то множество $U(t)$ содержит один элемент: $|U(t)| = |E(t)| = 1$. Если модель имеет фиксированные параметры, то количество вариантов зависит от количества и значений параметров:

$$|U(t)| = \prod_{i=1}^{|O(t)|} |Y(o_i)|, o_i \in O(t), i = 1, \dots, |O(t)|$$

Межсоединения СФ-блока t характеризуются следующими компонентами: $\forall n_t \in N(t): n_t = (\mu(n_t), \{p_t^s(n_t)\}, \{p_t^d(n_t)\})$, где $\mu(n_t)$ – имя межсоединения, $\{p_t^s(n_t)\}$ – множество контактов типа «источник», $\{p_t^d(n_t)\}$ – множество контактов типа «приемник».

На первом этапе архитектурно-зависимого отображения СФ-блоков выполняется отображение $m_y(e_y) \in Y$ из схемы c_y в $m_t(e_t) \in T$ схемы c_t . На втором этапе модель СФ-блока из множества $m_t(e_t) \in T$ схемы c_t отображается в модель из библиотеки для физического синтеза B .

Для отображения СФ-блоков из схемы c_b в СФ-блоки РСнК разработано формализованное представление СФ-блоков в библиотеке для топологического синтеза, включающее топологическую информацию о СФ-блоке и его архитектурных особенностях.

Формализованное представление СФ-блоков из библиотеки для топологического синтеза

Множество СФ-блоков для физического синтеза состоит из двух подмножеств: $B = S \cup H$, где $S = \{s\}$ – множество гибких СФ-блоков; $H = \{h\}$ – множество жестких СФ-блоков.

Множество жестких СФ-блоков представлено следующими компонентами: $\forall h \in H: h = (\mu(h), m_c(h), P(h))$, где $\mu(h)$ – имя СФ-блока, $m_c(h) \in C_M$ – модель жёсткого СФ-блока, из схемотехнического описания РСнК, $P(h) = \{p_h\}$ – контакты СФ-блока. Количество контактов жесткого СФ-блока,

их направление и функциональное назначение совпадает с контактами подсистемы $P(c)$, $c \in C_M$.

Гибкий СФ-блок представлен следующими компонентами: $\forall s \in S: s = (\mu(s), O(s), P(s), E(s), N(s))$, где $\mu(s)$ – имя СФ-блока; $O(s) = \{o_s\}$ – параметры СФ-блока; $P(s) = \{p_s\}$ – внешние контакты; $E(s) = \{e_s\}$ – элементы в составе гибкого СФ-блока; $N(s) = \{n_s\}$ – межсоединения.

Элементы множества $E(s)$ характеризуются компонентами, аналогичными компонентам ЛБ подсистемы c_i : $\forall e_s \in E(s): e_s = (\mu(e_s), m_l(e_s), P(e_s), r(e_s))$, где $\mu(e_s)$ – имя элемента; $m_l(e_s) \in L_{LE}$ – модель элемента, представленная в библиотеке ЛЭ; $P(e_s) = \{p_q\}$ – контакты элемента, которые частично или полностью повторяют контакты модели, а также характеризуются введённым для неё набором компонентов; $r(e_s)$ – относительные координаты элементов гибкого СФ-блока.

Элементы библиотеки ЛЭ $\forall l \in L_{LE}$ получены с помощью программирования ЛБ РСнК под определённую функцию. Они характеризуются именем, моделью – $m_c(l) \in C_{LE}$ и внешними контактами, имеющими направление и функциональное назначение контактов подсистемы $c \in C_{LE}$.

Относительные координаты элементов гибкого СФ-блока представляют собой пару компонентов: $r(e_s) = (r_l, r_c)$, где r_l – номер строки, r_c – номер столбца элемента в СФ-блоке: $r_l = 1, \dots, I_{max}$, $r_c = 1, \dots, J_{max}$, где I_{max} – максимальный номер строки элементов и J_{max} – максимальный номер столбца, не превышающие количество строк и столбцов РСнК. Они позволяют разместить элементы гибкого СФ-блока относительно друг друга. Началом отсчета координат является левый нижний угол СФ-блока.

Межсоединения гибкого СФ-блока представлены следующими компонентами: $\forall n_s \in N(s): n_s = (\mu(n_s), p_q^s(n_s), \{p_q^d(n_s)\}, v_s^s(n_s), \{v_s^d(n_s)\})$, где $\mu(n_s)$ – имя меж-соединения, $p_q^s(n_s)$ – контакт типа «источник», $\{p_q^d(n_s)\}$ –

множество контактов типа «приемник», $v_s^s(n_s)$ – узел РСнК, соответствующий межсоединению разрабатываемой схемы, типа «источник», $\{v_s^d(n_s)\}$ – множество узлов РСнК типа «приемник». При этом, если $\forall p_q \in p_q^s(n_s) \cup \{p_q^d(n_s)\}$ и $n_s \in N(s)$, то $p_q \in \cup_{i=1}^{|E(s)|} P(e_s^i) \cup P(s)$.

Подключение межсоединений к специализированным контактам ЛБ РСнК возможно только после корректного размещения элементов СФ-блока $s \in S$ друг относительно друга и трассировке его межсоединений через специализированные ресурсы. Корректное размещение элементов СФ-блока достигается за счет относительных координат, введенных в модели, трассировка через специализированные ресурсы – за счет компонентов межсоединений, позволяющих установить соответствие между соединениями СФ-блока и РСнК.

Пусть проектируемая схема c_b и составляющие её гибкие СФ-блоки представлены в плоском виде. Тогда размещение элементов схемы – это процесс, в ходе которого каждому её элементу ставится в соответствие элемент РСнК: $E(c_b) \rightarrow \{e_f: e_f \in E(c_f)\}$. При этом относительные координаты элементов СФ-блока учитываются следующим образом:

если $e_b^i, e_f^j, e_b^i \in E(c_b), e_f^j \in E(c_f), m_b(e_b^i) \in L_{LE}, m_c(e_f^j) \in C_{LE}$,

$i = 1, \dots, |E(s)|, j = 1, \dots, |E(c_f)|$, то

$(r_l(e_b^i), r_c(e_b^i)) \rightarrow (r_l(e_f^j) + r_l(e_b^i), r_c(e_f^j) + r_c(e_b^i))$.

Чтобы гарантировать использование специализированных трассировочных ресурсов РСнК в СФ-блоке, используется точечная трассировка. Для определения понятия точечной трассировки в САПР задан трассировочный граф: $G = (V, A)$. Вершины V являются узлами РСнК: $\forall v_g \in V: v_g = (n_v(v_g), d_v(v_g))$, $n_v(v_g) \in N(c_f)$, где $n_v(v_g)$ – межсоединение РСнК, соответствующее вершине, $d_v(v_g)$ – направление вершины (направленная

прямая, направленная инверсная, двунаправленная). Ребра графа $A = \{a_g\}$ являются упорядоченной парой вершин $A \subseteq \{v_g^s, v_g^d \mid v_g^s, v_g^d \in V \& v_g^s \neq v_g^d\}$.

С учетом вышесказанного, точечная трассировка – это установление соответствия между межсоединением СФ-блока и одним узлом или одним ребром графа трассировки. Для её реализации введены несколько подходов. Первый подход осуществляет точечную трассировку с помощью заданного межсоединения СФ-блока net , соответствующего ему ребра трассировочного графа $a_g = (node0, node1)$ и направления узла приемника $d(node1)$: если $n_s = net$, $net \in N(s)$, то $v_s^s(n_s) = v_g^s(n_s)$, $v_s^d(n_s) = v_g^d$, $v_g^s = node0$, $v_g^d = node1$, $d(v_g^d) = d(node1)$, $d(node1) \in \{d_{dir}, d_{inv}, d_{bidir}\}$.

Второй подход осуществляет точечную трассировку с помощью межсоединения проектируемой схемы net , соответствующего ему узла РСнК $node0$ и направления узла $d(node0)$: если $n_s = net$, $net \in N(s)$, то $v_s^d(n_s) = v_g^d$, $v_s^s(n_s) \in \{v_s^d(n_s)\}$, $v_g^d = node0$, $d(node0) \in \{d_{dir}, d_{inv}, d_{bidir}\}$.

Заключение

В данной работе представлена обобщенная модель жёстких и гибких сложно-функциональных блоков, позволяющая выполнить отображение архитектурно-независимых блоков из списка соединений проектируемой схемы в СФ-блоки, входящие в состав РСнК. Модель объединяет в себе ряд множеств СФ-блоков на различных этапах маршрута проектирования ИС на основе РСнК: параметризованные архитектурно-независимые СФ-блоки для логического синтеза, вспомогательные архитектурно-зависимые СФ-блоки для технологического отображения и архитектурно-зависимые СФ-блоки для топологического синтеза и отображения в компоненты РСнК. Формализованное представление СФ-блоков из последнего множества также включает в себя компоненты, позволяющие выполнить размещение и

трассировку СФ-блоков с учетом специализированных коммутационных ресурсов и нестандартных решений и особенностей архитектуры РСнК.

Также модель, представленная в статье, кроме СФ-блоков, включает в себя формализованное представление базового кристалла и проектируемой схемы, формирующейся в процессе генерации СФ-блоков из базиса РСнК, выполняющейся на основе архитектурно-независимых СФ-блоков.

Литература

1. Заплетина М.А., Железников Д.А., Гаврилов С.В. Иерархический подход к трассировке реконфигурируемой системы на кристалле островного типа // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2020. Выпуск 3. С. 16-21.
2. Hauck S., Denon A. Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation. Morgan Kaufmann, 2008. 944 p.
3. Farooq U., Parvez H., Marrakchi, Z. Exploration of Heterogeneous FPGA Architectures // Hindawi Publishing Corporation International Journal of Reconfigurable Computing, 18 p.
4. Тельпухов Д.В., Рухлов В.С., Иванова Г. А., Рыжова Д.И., Надоленко В.В., Деменева А.И. Исследование вариантов частичного резервирования при проектировании сбоеустойчивых логических блоков ПЛИС// Инженерный вестник Дона, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4681 (дата обращения: 12.06.2023).
5. Ristimaki T., Nurmi J. Reconfigurable IP blocks: a survey // 2004 International Symposium on System-on-Chip. 2004, pp. 117 – 122.
6. Гаврилов С.В., Хватов В.М. Преимущества специализированных ресурсов программируемых логических интегральных схем для гибких сложно-функциональных блоков // Инженерный вестник Дона, 2021, №5. URL: ivdon.ru/ru/magazine/archive/n5y2021/6997 (дата обращения: 29.05.2023).

7. Intel Quartus Prime Standard Edition User Guide Getting Started // Intel Corporation. URL: intel.com/content/dam/altera-global/en_US/pdfs/literature/ug/ug-qps-getting-started.pdf (дата обращения: 15.09.2021).
8. Xilinx Vivado Design Suite User's Guide // URL: docs.xilinx.com/viewer/book-attachment/pYmnSeekkH6djEKjge4YUQ/InhHg5Xz61ZW6cwJAZZFhQ (дата обращения: 04.04.2023).
9. Koch A. Structured Design Implementation - A Strategy for Implementing Regular Datapaths on FPGAs // Fourth International ACM Symposium on Field-Programmable Gate Arrays. 1996. pp. 151-157.
10. Kafafi N., Bozman K., Wilton S. Architectures and Algorithms for Synthesizable Embedded Programmable Logic Cores // Proceedings of the 2003 ACM/SIGDA Eleventh International Symposium on Field Programmable Gate Arrays (FPGA '03), 2003, pp. 3-11.
11. Эннс В. И., Гаврилов С. В., Хватов В. М., Курбатов В. Г. Проектирование ПЛИС и реконфигурируемых СнК с использованием методов программного анализа и прототипирования // Микроэлектроника. 2021. Т. 50, № 6. С. 467-480.
12. Иванников А.Д., Стемповский А.Л. Математическая модель цифровых блоков для системы совместного моделирования технических средств и программно-микропрограммного обеспечения // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2021. Выпуск 1. С. 2-8.

References

1. Zapletina M.A., Zheleznikov D.A., Gavrilov S.V. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh sistem (MES), 2020, № 3, pp. 16-21.
2. Hauck S., Denon A. Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann, 2008. 944 p.

3. Farooq U., Parvez H., Marrakchi Z. Exploration of Heterogeneous FPGA Architectures, Hindawi Publishing Corporation International Journal of Reconfigurable Computing, 18 p.
 4. Tel'pukhov D.V., Rukhlov V.S., Ivanova G. A., Ryzhova D.I., Nadolenko V.V., Demeneva A.I. Inzhenernyj vestnik Dona, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4681 (accessed 12 June 2023).
 5. Ristimaki T., Nurmi J. Reconfigurable IP blocks: a survey, 2004 International Symposium on System-on-Chip, 2004, pp. 117 – 122.
 6. Gavrilov S.V., Khvatov V.M. Inzhenernyj vestnik Dona, 2021, №5. URL: ivdon.ru/ru/magazine/archive/n5y2021/6997 (accessed 29 May 2023).
 7. Intel Quartus Prime Standard Edition User Guide Getting Started. URL: intel.com/content/dam/altera-/global/en_US/pdfs/literature/ug/ug-qps-getting-started.pdf (accessed 15 September 2021)
 8. Xilinx Vivado Design Suite User's Guide. URL: docs.xilinx.com/viewer/book-attachment/pYmnSeekkH6djEKjge4YUQ/InhHg5Xz61ZW6cwJAZZFhQ (accessed 04 April 2023).
 9. Koch A. Structured Design Implementation - A Strategy for Implementing Regular Datapaths on FPGAs, Fourth International ACM Symposium on Field-Programmable Gate Arrays, 1996. pp. 151-157.
 10. Kafafi N., Bozman K., Wilton S. Architectures and Algorithms for Synthesizable Embedded Programmable Logic Cores, Proceedings of the 2003 ACM/SIGDA Eleventh International Symposium on Field Programmable Gate Arrays (FPGA '03), 2003. pp. 3-11
 11. Enns V.I., Gavrilov S.V., Khvatov V.M., Kurbatov V.G. Mikroelektronika, 2021, T. 50, № 6, pp. 467-480.
 12. Ivannikov D.A., Stempkovsky A. L. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh sistem (MES), 2021. № 1. pp. 2-8.
-