

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК 621.3.049.771.14

С.В. Гаврилов, Г.А. Иванова, П.С. Волобуев**АКТУАЛЬНЫЕ ПРОБЛЕМЫ АВТОМАТИЗАЦИИ
ЛОГИКО-ТОПОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ
БИБЛИОТЕЧНЫХ ЭЛЕМЕНТОВ И БЛОКОВ СБИС
ДЛЯ НАНОМЕТРОВЫХ ТЕХНОЛОГИЙ**

Решаются проблемы автоматизации проектирования заказных интегральных схем для перспективных технологий с минимальными нормами 45, 32, 22 нм и ниже. Основное внимание уделяется проблемам на стыке логического и топологического проектирования. Предлагается метод, обеспечивающий существенное по сравнению с другими известными подходами на логическом уровне повышение точности анализа задержек с учетом одновременного переключения нескольких входов вентиля. Рассматриваются специфические проблемы проектирования для новых перспективных технологий с трехмерной структурой затвора транзистора, так называемых FinFET технологий.

Ключевые слова: SP-граф, сложно-функциональный блок (СФ-блок), FinFET транзисторы (транзисторы с трехмерной структурой затвора в форме «плавника»), КМОП технология.

Введение. В связи с уменьшением топологических размеров транзисторов, повышением степени интеграции СБИС и сокращением сроков проектирования возникают принципиально новые требования к проектированию СБИС. Однако на протяжении нескольких последних десятилетий возможности моделирования постоянно отставали от темпов развития технологий и потребностей электронной промышленности. При переходе к технологиям с наноразмерными компонентами сложности в маршруте проектирования достигают таких масштабов, что для компаний по производству и разработке микроэлектронной аппаратуры сопровождение и развитие полного комплекта собственных средств САПР становится нерентабельным. Если до недавнего времени ведущие разработчики микроэлектронной аппаратуры (Intel, IBM, Motorola, Freescale и др.) ориентировались на собственные средства автоматизации проектирования заказных микросхем, то в последнее время преобладает стремление перехода на средства полузаказного проектирования от известных лидеров в области САПР, таких как Synopsys, Cadence, MentorGraphics. Вместе с тем количество правил

проектирования для нанометровых технологий настолько велико, что средства автоматического синтеза в маршруте полузаказного проектирования не способны обеспечить соблюдения полного набора технологических правил и ограничений для каждого прибора на кристалле.

Размеры элементов СБИС приближаются к физическим пределам, и на поведение цифровых схем существенное влияние начинают оказывать нелинейные эффекты: помехи в межсоединениях из-за емкостных связей, сопротивление шин питания и земли, индуктивность и взаимная индукция межсоединений. Если раньше все эти эффекты анализировались на заключительном этапе верификации проекта после этапа проектирования топологии, то теперь их необходимо учитывать уже на ранних стадиях логического проектирования и размещения элементов. Таким образом, с переходом на нанометровые технологии повысилась актуальность смешанного и многоуровневого моделирования, сочетающего в себе логические, схемотехнические и топологические методы.

Переход на уровень нанометровых технологий с размером каналов транзисторов 100 нм и

ниже порождает ряд принципиально новых проблем, не решенных в существующих САПР СБИС. К ним, прежде всего, можно отнести следующие.

- Высокий удельный вес ошибок в точности воспроизведения технологических размеров и параметров и, как следствие, необходимость учета вариаций (т.е. неопределенности) технологических и схемных параметров.

- Невозможность соблюдения полного набора правил и норм проектирования в автоматическом режиме при использовании существующих подходов к решению задач логического и топологического синтеза.

Целью данной работы является разработка моделей библиотечных элементов обеспечивающих повышение точности оценок задержек на этапах логического и топологического проектирования.

Модели библиотечных элементов для статического временного анализа цифровых СБИС. При переходе на техпроцессы с глубоко-субмикронными и нанометровыми проектными нормами на рабочие характеристики цифровых СБИС начинают существенно влиять электрическое сопротивление межсоединений, эффект Миллера, эффекты короткого канала, вариации технологических параметров, внешние рабочие условия и т.д. Адекватный учет этой разнородной информации в статических временных моделях стандартных цифровых элементов на стадии проектирования СБИС необходим в целях повышения эффективности методов верификации проектов и, следовательно, обеспечения приемлемого параметрического выхода годных микросхем [1].

Одним из основных этапов логико-временной верификации проектов цифровых СБИС является статический временной анализ (СВА) (StaticTimingAnalysis - STA) [2], который при заданном описании схемы на вентиляльном уровне (gate-levelnetlist – *.vg) на основе библиотеки временных параметров стандартных цифровых элементов (например, формат Liberty – *.lib, *.db) позволяет делать оценку таких временных характеристик цифровой схемы, как минимальная и максимальная задержки, минимальная ширина синхроимпульса и т. д. Если результаты СВА показывают, что схема удовлетворяет всем временным ограничениям, это снижает вероятность функционально-временных сбоев в процессе функционирования разработанной СБИС.

В качестве входных данных для статического временного анализа используется информация о временных параметрах стандартных циф-

ровых элементов и межсоединений, полученная посредством их характеристики – электрического моделирования для отобранных комбинаций входных воздействий с учетом разнородных вариаций технологических (длина/ширина канала МОП-транзисторов, толщина окисла и т.д.) и внешних (напряжение питания, температура) параметров. Таким образом, в процессе СВА достигается адекватный учет всех прогнозируемых временных нарушений. При этом время прогона СВА, как правило, имеет линейную зависимость от размеров схемы [2]. Два указанных преимущества СВА делают его неотъемлемым этапом временной верификации современных цифровых СБИС, а динамический временной анализ часто используется лишь в качестве дополнения к СВА для более точного расчета отдельных временных характеристик.

В наши дни в индустрии САПР СБИС постоянно развивается механизм моделирования временных параметров и характеристик цифровых элементов как на транзисторном, так и на вентиляльном уровне. К примеру, если в начале эры логико-временного моделирования каждому цифровому вентилю в соответствие ставилась статическая (так называемая собственная) задержка, то дальнейшее развитие программ моделирования сопровождалось постоянным усложнением моделей для достижения приемлемой точности результатов анализа [3, 4]. Можно отметить следующие основные причины, приводящие к периодической необходимости создания новых моделей временных параметров:

- Ухудшение точности (относительно SPICE-подобных программ электрического моделирования) предыдущих моделей при оценке временных параметров;

- Учет новых эффектов деградации временных параметров схемы, влияние которых усилилось вследствие масштабирования размеров транзисторов, повышения степени их интеграции на кристалле и увеличения рабочих частот СБИС и т.д.;

- Увеличение требуемого объема памяти для хранения библиотеки временных параметров вентилялей;

- Повышение требований к скорости расчета модели в программе СВА.

Нелинейная модель задержки. До недавнего времени “де-факто” стандартом модели цифровой ячейки являлась так называемая нелинейная модель задержки (*Non-Linear Delay Model - NLDM*), которая на сегодняшний день поддерживается большинством современных программ СВА (например, *Synopsys PrimeTime*, *Cadence Encounter Timing System*). В *NLDM*-модели каж-

дой комбинации “входной фронт/емкостная нагрузка” (S_i / C_0) соответствуют определенные значения задержки вентиля и фронта сигнала на его выходе [5].

Табличные значения задержек (D) и длительностей выходных фронтов (S_0) рассчитываются посредством электрического ($SPICE$) моделирования для каждого стандартного цифрового элемента из библиотеки, сохраняются в формате *Liberty* и используются затем в качестве входных данных $NLDM$ -модели в процессе СВА.

Токковые модели задержки. Перспективной альтернативой $NLDM$ -модели выступает модель цифровой ячейки на основе источника тока. Две разновидности такой модели – $ECSM$ -модель фирмы *Cadence* и CCS -модель фирмы *Synopsys* – позволяют достичь выигрыша по времени в сравнении со $SPICE$ -моделированием при сравнительно небольшой потере в точности (около 2 %), а также дают возможность учитывать влияние RC -параметров межсоединений, влияние сопротивления которых стало доминировать при определении задержек [6, 7].

Модели $ECSM$ и CCS представляют собой трехмерные таблицы типа $V_o(t, S_i, C_0)$ и $I_o(t, S_i, C_0)$ соответственно, где V_o – выходное напряжение, а I_o – выходной ток цифрового элемента, а t – моменты выборки значений соответственно напряжения или тока. Необходимость хранения трехмерных таблиц резко увеличивает объемы памяти для библиотек. С этой точки зрения практический интерес представляет поиск возможных путей упрощения данных моделей при сохранении приемлемой точности СВА.

Несомненным преимуществом токковых моделей является устранение в них существенного недостатка предыдущих моделей задержки: потери точности при замене реальной входной емкости приемника сигнала (*receiver*) на эквивалентную заземленную пассивную емкость. Реальная входная емкость приемника сигнала изменяется в процессе переключения. Кроме того, она зависит от напряжения питания, емкости нагрузки, длительности входного фронта и предыдущего состояния выхода приемника. Следовательно, замена такой емкости статическим пассивным эквивалентом является достаточно грубой аппроксимацией. В токковых моделях ($ECSM$, CCS) эта емкость представлена двумя значениями на интервале переключения выходного сигнала, что позволяет с большей точностью аппроксимировать фронты сигналов.

Метод интервальных оценок задержек и выходных фронтов библиотечных элементов нанометровых КМОП-схем. Для решения широкого класса задач моделирования цифровых

схем наряду с максимальной задержкой в узле требуется и минимальная задержка. На точный расчет минимальной задержки существенное влияние оказывают гонки и одновременные переключения нескольких входов вентиля. Вместе с тем существующие средства анализа быстродействия на логическом уровне, как правило, основаны на использовании упрощенной модели задержки вентиля с использованием единственного переключения на входе.

Основная цель СВА – нахождение максимальной задержки в комбинационном блоке для последующей оценки периода и частоты функционирования схемы. Однако для решения широкого класса задач моделирования цифровых схем с технологическими нормами 90 нм и ниже наряду с максимальной задержкой требуется и минимальная задержка. К числу таких задач относятся, в частности, определение всевозможных интервалов переключений узлов схемы при анализе помехоустойчивости [8-10], оценка максимального тока питания с учетом одновременного переключения большого количества вентилялей [11], характеристика и генерация макромоделей сложно-функциональных блоков, контроль времени удержания (*hold*) триггера и др. Другими словами, требуется анализ интервалов возможных переключений с достаточно точной оценкой как верхней, так и нижней границ переключений.

Наибольшую точность при расчете интервалов возможных переключений могут дать методы схемотехнического моделирования на транзисторном уровне, но поскольку комбинационные СФ-блоки имеют большие размеры и большое количество входов/выходов, определить их быстродействие посредством того или иного вида полного моделирования почти невозможно. СВА и его модификации представляют собой реальную возможность решить перечисленные задачи, хотя решение и является зачастую не точным в связи с использованием упрощенной модели задержки отдельных библиотечных элементов.

Очевидно, что одновременное переключение нескольких входов библиотечного элемента может привести к существенному уменьшению значения задержки элемента за счет активации нескольких путей проводимости тока/заряда. Так, например, для элемента NAND с двумя входами задержка при переключении одного входа больше, чем задержка при одновременном переключении двух входов, так как в последнем случае выход заряжается через несколько МОП транзисторов. Проблема состоит в том, что для точного анализа одновременного переключения

входов требуется переход от двумерных таблиц типа NLDМ к четырех – пятимерным зависимостям с целью учета фронтов всех переключаемых входов, а также смещения сигналов между входами. Для снижения размерности требуются эвристические упрощения.

В одной из немногих работ, ориентированных на анализ одновременного переключения двух входов вентиля [12], предложено упрощение на основе так называемой модели “V-shape”. Показано, что зависимость задержки от смещения переключений двух входов с хорошей точностью описывается кусочно-линейной зависимостью “V-формы”. Предложены также эвристические формулы для аппроксимации задержки при различных значениях входных фронтов. Использование такой модели приводит к существенному повышению точности расчета задержки СФ-блока на логическом уровне по сравнению с “pin-to-pin” подходом.

На рисунке 1 показан график зависимости задержки переключения от величины смещения между фронтами входных сигналов на примере задержки переключения выхода z элемента NAND из 0 в 1. На графике представлены отдельно задержки $D_{x \rightarrow z}$, $D_{y \rightarrow z}$ от каждого из входов до выхода. Оба переключения в данной ситуации являются контролирующими, т.е. ведут к переключению выхода, даже если второй вход не переключится. В связи с этим интерес представляет задержка от переключения первого из входов, другими словами, максимальная из двух задержек или верхняя огибающая двух графиков: $D = \max(D_{x \rightarrow z}, D_{y \rightarrow z})$.

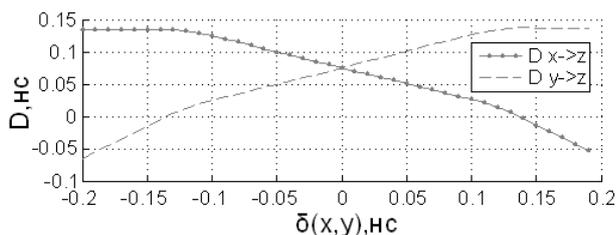


Рисунок 1 – График зависимости задержки переключения от величины смещения между фронтами входных сигналов (элемент NAND2, технология 45нм)

Очевидно, что при достаточно больших значениях модуля $|\delta(x, y)|$ зависимость $D = \max(D_{x \rightarrow z}, D_{y \rightarrow z})$ от смещения исчезает и задержка сводится к обычному варианту “pin-to-pin”.

Из графика видно, что верхняя огибающая с хорошей точностью может быть аппроксимирована кусочно-линейной зависимостью “V-формы”. Поскольку, $D_{y \rightarrow z} = D_{x \rightarrow z} + \delta(x, y)$ и при этом $D_{x \rightarrow z}$ монотонно убывает, а $D_{y \rightarrow z}$ монотонно возрастает, то минимум верхней огибающей

достигается в точке $\delta(x, y) = 0$, причем этот факт имеет место при любых значениях размеров транзисторов, входных фронтов, емкостей нагрузки и уровней измерения задержки.

Используя кусочно-линейную аппроксимацию “V-формы” можно вычислять значения задержки на логическом уровне $D(S_x, S_y, \delta(x, y))$ при различных значениях входных фронтов S_x, S_y и смещения $\delta(x, y)$. Для этого достаточно на этапе характеристики определить координаты точек излома.

С точки зрения интервального статического временного анализа наибольший интерес представляет минимальное значение задержки:

$$D_{\min} = D(S_x, S_y, 0).$$

В целях сокращения размеров таблиц и времени характеристики в работе [12] предлагаются эвристические формулы для оценки минимальной задержки при различных значениях входных фронтов $x = S_x, y = S_y$ в следующей форме:

$$D_{\min} = c_1 x^{1/3} y^{1/3} + c_2 x^{1/3} + c_3 y^{1/3} + c_4. \quad (1)$$

Значения коэффициентов c_1, c_2, c_3, c_4 могут быть найдены на основе метода наименьших квадратов по результатам серии схемотехнических моделирований для различных значений входных фронтов.

В работе [13] показано, что при уменьшении размеров транзисторов погрешность модели возрастает с 10-12 % для технологии 65 нм до 15-20 % для технологии 32 нм, что свидетельствует о необходимости разработки более точных моделей. Авторами настоящей статьи предлагается улучшенная модель задержки. Для повышения точности интервальных оценок задержек и выходного фронта исследованы альтернативные варианты аппроксимации минимальной задержки, а также минимального выходного фронта. Анализ поведения зависимости задержки от фронтов позволяет сделать вывод о применимости квадратичной аппроксимации задержки общего вида.

Так же, как и в случае модели [12], значения коэффициентов c_i для квадратичной аппроксимации [формула (2)] находятся на основе метода наименьших квадратов по результатам серии схемотехнических моделирований для различных значений входных фронтов $x = S_x, y = S_y$.

$$D_{\min} = c_1 x^2 + c_2 y^2 + c_3 xy + c_4 x + c_5 y + c_6. \quad (2)$$

Для более точной оценки минимальной задержки проведено сравнение результатов моделирования с учетом и без учета одновременного переключения входов:

$$\Delta_D = \min(D_1, D_2) - D_{\min}, \quad (3)$$

где D_1 – задержка, измеренная при переключении входного сигнала x , D_2 – задержка переключения входного сигнала y , Δ_D – коррекционная разница между задержкой элемента без учета и с учетом одновременного переключения входов.

Характер поведения коррекционной разницы Δ_D позволяет сделать вывод о целесообразности применения ее кубической аппроксимации с последующим расчетом минимальной задержки по формуле:

$$D_{\min} = \min(D_1, D_2) - \Delta_D. \quad (4)$$

На основе аппроксимации коррекционной разницы кубическим выражением получен результат, изображенный на рисунке 2. Из графика видно, что максимальная погрешность такого подхода составляет 0,8 %.

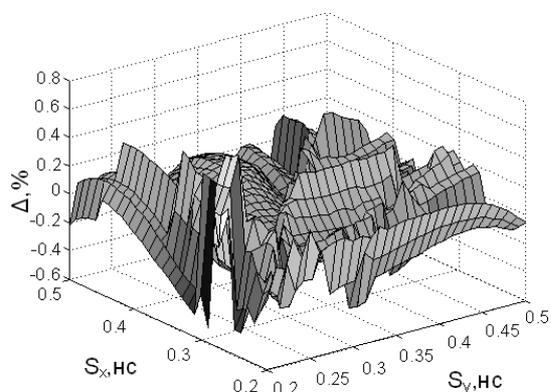


Рисунок 2 – Относительная погрешность расчета задержки на основе кубической аппроксимации коррекционной разницы между задержкой без учета и с учетом одновременного переключения

Методы проектирования заказных сложно-функциональных блоков в базисе элементов с регулярной топологической структурой в слоях поликремния и диффузии. С увеличением степени интеграции микросистем и уменьшением технологических размеров базовых элементов до 45 нм и ниже в значительной степени усложнились технологические правила и нормы проектирования. К примеру, количество правил проектирования для технологий с размерами транзистора 32 нм и ниже составляет несколько тысяч. Соблюдение полного набора правил и норм проектирования в автоматическом режиме становится невозможным при использовании существующих подходов к решению задач логического и топологического синтеза. Это ведет к необходимости большого объема ручной работы с редактированием схемы и топологии на заключительном этапе верификации проекта в целом.

Переход к использованию регулярных структур в нижних слоях топологии позволяет решить проблемы возрастающего числа проек-

ных норм для передовых технологий 22 нм и ниже. Исторически регулярность топологии в нижних слоях связывают с так называемой технологией базовых матричных кристаллов (БМК), имеющих predetermined набор библиотечных элементов. Применение аналогичного подхода при проектировании полностью заказных сложно-функциональных блоков позволяет значительно упростить проверку правил проектирования при разработке общего вида топологии.

В работах компании Intel для соблюдения требований регулярности в маршруте заказного проектирования предлагается подход к проектированию регулярных топологических структур с использованием конструкции специального типа – так называемого транзисторного шаблона [14]. Транзисторный шаблон определяется как промежуточная конструкция между транзистором и стандартной ячейкой [14].

Наибольший интерес представляет применение аналогичного подхода для новых технологий на основе так называемых “FinFET транзисторов”. FinFET транзисторы – это новое направление в проектировании микросхем, в котором за счет использования трехмерного затвора транзистора в форме плавника повышается эффективная ширина затвора при одинаковой площади логической ячейки. В последнее время появился ряд публикаций, связанных с проектированием на основе FinFET транзисторов.

Известны три основные конструкции FinFET структур [15]:

- 1) SG (Shorted-Gate) или TG (Tri-gate) FinFET, – затворы транзистора соединены друг с другом;
- 2) IG (Independent-Gate) или DG (Double-Gate) FinFET, – затворы транзистора независимы (на каждый затвор подается отдельный сигнал);
- 3) LP (LowPower) FinFET, – второй затвор (back-gate) транзистора подключен к смещенному напряжению для уменьшения статической мощности.

Использование FinFET-транзисторов для разработки набора базовых блоков для создания регулярной топологии приведено в работе [16], где рассмотрены два типа конструкций FinFET: SG-FinFET и IG-FinFET. Предлагается создавать геометрически регулярную топологию с помощью повтора однотипных блоков (элементы NAND2, NOR2 и NOT). Показано, что в плане уменьшения статической мощности и занимаемой площади для элементов NAND2, NOR2 и NOT IG-FinFET эффективнее, чем SG-FinFET. Так как не приведен эскиз топологии, сложно оценить правдоподобность полученных результатов.

В работах [17] и [18] можно найти возможные эскизы топологии FinFET-транзисторов и базовых логических элементов, таких как NOT, NOR2 и 2xNOR2. Вместе с тем на сегодняшний день проектирование на основе FinFET-транзисторов осуществляется на основе ограниченного набора библиотечных элементов, отсутствуют средства синтеза сложносоставных элементов, поэтому возникает необходимость самостоятельной разработки принципов создания топологии FinFET-структур для маршрута заказного проектирования с регулярными структурами.

Структура FinFET-транзисторов. На рисунке 3 приведена простая структура SG (Shorted-Gate) FinFET транзистора. Структура состоит из “плавника” (fin) – область диффузии для стока и истока, которая окружена затвором. Физические размеры транзистора описываются высотой плавника h_{FIN} , толщиной плавника (или кремния) T_{Si} и длиной канала L_{FIN} [19].

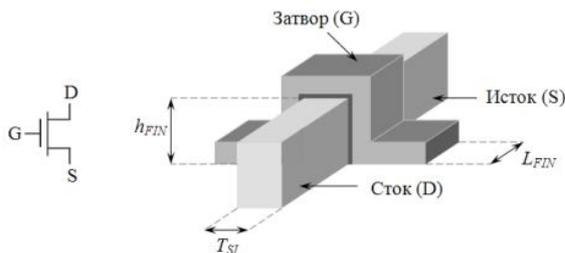


Рисунок 3 – Структура транзистора SG-FinFET

Так как верхняя часть затвора отделена от плавника сравнительно толстым слоем окисла, вклад этой части затвора в работу транзистора можно не учитывать [20]. У такого транзистора длина канала L_{FIN} , а ширина – W_{min} :

$$W_{min} \approx 2 \cdot h_{FIN} \cdot \quad (5)$$

На рисунке 4 приведена структура IG (Independent-Gate) FinFET транзистора, которую можно получить путем удаления верхней части затвора обычного SG-FinFET транзистора [21].

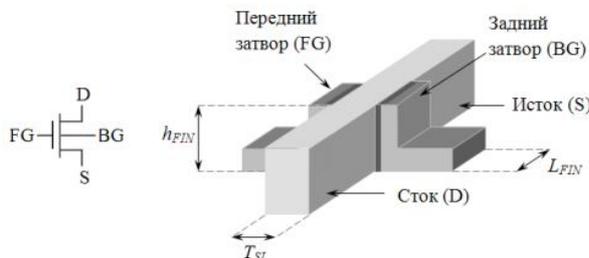


Рисунок 4 – Структура транзистора IG-FinFET

Затворы IG-FinFET транзистора независимы, это позволяет использовать один из затворов (back-gate) для введения обратного напряжения смещения, что дает возможность управлять пороговым напряжением транзистора, а тем самым утечками и значениями задержек транзистора

[22]. Часто второй затвор подключают к самому низкому (или высокому) потенциалу для проектирования на их основе маломощных схем – LP (LowPower) FinFET. Независимость затворов IG-FinFET структур также дает возможность образовывать параллельно подключенные транзисторы, ширина каждого из которых $W_{min}=h_{FIN}$ [23].

Если соединить затворы IG-FinFET транзистора, можно получить транзистор с шириной W_{min} , определяемой формулой (5), но это приводит к увеличению площади транзистора, так как контакт ко второму затвору требует определенного места и соблюдения правил проектирования [24].

Плотность топологии на FinFET транзисторах. Минимальная ширина FinFET-транзисторов равна W_{min} , из чего следует, что для получения ширины W , которая больше чем W_{min} , нужно подключить N_{FIT} транзисторов параллельно [25], где

$$N_{FIT} = W/W_{min} \cdot \quad (6)$$

Так как N_{FIT} обычно должно быть целым, то получается, что ширина транзистора кратна W_{min} .

Метод проектирования регулярных структур на основе выбора шаблона. Использование шаблонов при построении логических элементов является одним из подходов к проектированию схем с регулярной топологией.

На рисунке 5 представлено сравнение обычной SG-FinFET структуры стандартной ячейки ИЛИ-НЕ (слева) с предлагаемой регулярной SG-FinFET структурой на основе шаблона (справа).

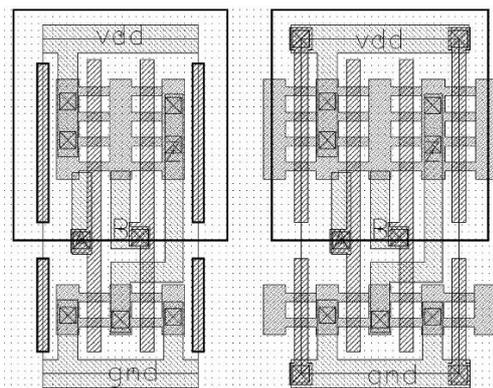


Рисунок 5 – Ячейка двухвходового ИЛИ-НЕ: SG-FinFET структура стандартной ячейки (слева), регулярная SG-FinFET структура (справа)

Особенность шаблона в том, что объединив множество таких же шаблонов, можно получить регулярность по слоям поликремния и диффузии (рисунок 6).

В регулярной структуре присутствуют изолирующие затворы, которые подключены к шинам питания, следовательно, всегда закрыты. Между изолирующими затворами имеются два

функциональных затвора. В отличие от структуры стандартной ячейки, в регулярной структуре слои диффузии выходят за пределы ячеек, и появляются изолирующие затворы.

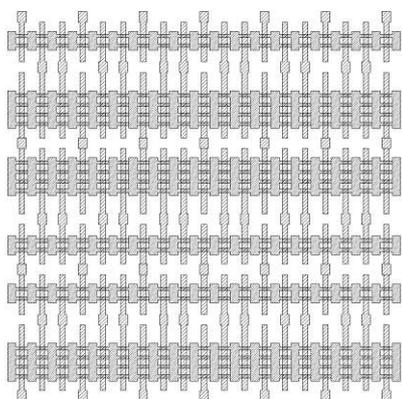


Рисунок 6 – Регулярная структура рядом стоящих ячеек по слоям диффузии и поликремния

Однако если стандартные ячейки позволяют менять размеры ячейки на кратные шаги сетки, а также менять количество подключений к затворам внутри ячейки, то в регулярной структуре количество подключений к затворам фиксировано, и ячейки должны быть кратны размерам самой маленькой ячейки. В нашем случае самые маленькие блоки – это двухвходовые, в которых можно реализовать такие элементы, как ИЛИ-НЕ, И-НЕ, буфер, инвертор.

Для реализации трехвходовых регулярных структур, как например И-ИЛИ-НЕ21 (AOI21), понадобится ячейка вдвое больше, поэтому один из входов элемента И-ИЛИ-НЕ21 подключен к транзистору, размеры которого в два раза больше, чем размеры остальных (рисунок 7), или один затвор остается неподключенным, что недопустимо.

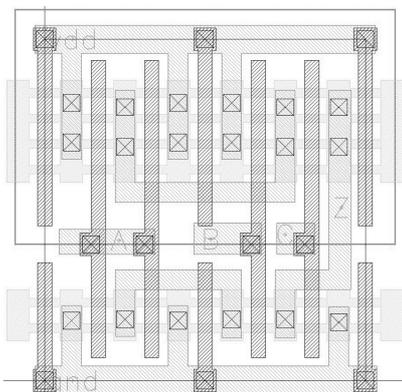


Рисунок 7 – Регулярная структура трехвходового И-ИЛИ-НЕ21

В регулярных структурах сложнее получить нужную нагрузочную способность, чем в классических КМОП стандартных ячейках, но автоматизировать проектирование такой структуры значительно легче.

Имея представление схемы на транзисторном уровне, с помощью программного обеспечения можно на регулярной структуре получить топологию нужной схемы по определенным шаблонам и алгоритму синтеза.

Математические модели логических элементов. Для описания топологии схемы предлагается использовать модели компонент в форме графа вложенности последовательно-параллельных структур (SP-граф) [26]. Отличительная особенность предлагаемого подхода состоит в независимом синтезе pull-up и pull-down цепей. Это обеспечивает дополнительные степени свободы для достижения требуемых значений характеристик схемы по площади, быстродействию и потребляемой мощности по сравнению со стандартными подходами на основе предопределенных библиотек элементов.

Граф вложенности последовательно-параллельных структур (SP-граф) определяется как бинарный ориентированный упорядоченный ациклический граф $G = (V, E)$, где каждая вершина из множества V описывает логическую функцию, а каждая дуга в множестве E , исходящая из вершины, описывает ссылки на аргументы соответствующей функции. Рассматриваются функции пяти типов для цепочки от узла земли (pull-down), а именно, конъюнкция ($f(a,b) = a \& b$) для последовательного соединения внутри вентиля, дизъюнкция ($f(a,b) = a + b$) для параллельного соединения внутри вентиля, отрицание ($f(a) = \sim a$), функция конъюнкции с отрицанием (NAND) ($f(a,b) = \sim(a \& b)$) для последовательного соединения на выходе вентиля, функция дизъюнкции с отрицанием (NOR) ($f(a,b) = \sim(a + b)$) для параллельного соединения на выходе вентиля.

Таким образом, граф описывает композицию логических функций. Вершины соответствуют логическим функциям. Дуги описывают подстановку аргументов для функции, соответствующей вершине, из которой эти дуги исходят.

Задача описания функций на уровне топологии усложняется условием взаимно однозначности между топологией и графовым деревом, что приводит к некоторым ограничениям способов описания. В данной работе предложен более широкий набор функций конъюнкции и дизъюнкции для описания топологической структуры вентиля. Так если порядок транзисторов в топологии соответствует порядку имен входов в функции, то к оператору добавляется индекс N (normal), если порядок противоположный, то M (mirror).

С помощью операций $\&_N$, $\&_M$, $+_N$, $+_M$ и \sim можно описать топологию элемента И-ИЛИ-НЕ21, изображенную на рисунке 7.

$$f = ((A +_N A) +_N B) \&_N C. \quad (7)$$

$$f = ((A \&_N A) \&_N B) +_N C. \quad (8)$$

Формула (7) – описание pull-up цепи, формула (8) – описание pull-down цепи.

Благодаря обобщенной графовой модели появляется возможность разработки эффективного алгоритма структурной оптимизации. При этом, с одной стороны, сохраняется большее число степеней свободы за счет логического аспекта (т.е. булевой формы графа), с другой – появляется нужная точность задержек, мощности и площади за счет структурной интерпретации на транзисторном уровне.

Закключение. В данной работе предложено согласованное решение проблем синтеза схемы и топологии на основе модели компонент в форме графа вложенности последовательно-параллельных структур (SP-граф) с независимым анализом цепей земли (pull-down) и питания (pull-up). Исследованы варианты топологического представления логических элементов на FinFET транзисторах. Разработан регулярный шаблон для отображения SP-структуры на FinFET транзисторах.

Предложен метод, обеспечивающий существенное повышение точности оценок задержек с учетом одновременного переключения входов вентиля по сравнению с известными подходами. На основе численных экспериментов показано, что предлагаемый метод аппроксимации коррекционной разницы обеспечивает точность расчета минимальных значений задержек и выходных фронтов в пределах 0,8% по сравнению с точным схемотехническим моделированием.

Библиографический список

1. Keller I., Tam K. H., Kariat V. Challenges in Gate Level Modeling for Delay and SI at 65nm and Below // Proceedings of DAC. 2008. P. 468-473.
2. Blaauw D., Chopra K., Srivastava A., Scheffer L. Statistical Timing Analysis: From Basic Principles to State of the Art // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2008. Vol. 27. P. 589-607.
3. Стемковский А.Л., Гаврилов С.В., Каграманян Э.Р. Методы логико-временного анализа заказных блоков СБИС // Известия ВУЗов. Электроника. 2008. № 5. С. 41-50.
4. Гаврилов С.В., Глебов А.Л., Стемковский А.Л. Методы логического и логико-временного анализа цифровых КМОП СБИС. М.: Наука, 2007. 220 с.
5. Croix J.F., Wong D.F. A Fast and Accurate Technique to Optimize Characterization Tables for Logic Synthesis // Proceedings of IEEE Design Automation Conference. 1997. P. 337-340.
6. CCS Timing: Technical White Paper. <http://www.synopsys.com>.
7. Korshak A., Lee J.-C. An Effective Current Source Cell Model for VDSM Delay Calculation // in Proc. IEEE International Symposium on Quality Electronic Design. 2001. P. 296-300.
8. Glebov A., Gavrilov S., Blaauw D. et al., False noise analysis using resolution method // ISQED 2002. P. 437-442.
9. Стемковский А.Л., Гаврилов С.В., Глебов А.Л. Анализ помехоустойчивости цифровых схем на основе метода резолюций // Электроника, Известия ВУЗов. Москва. 2004. №6. С. 64-71.
10. Стемковский А.Л., Гаврилов С.В., Глебов А.Л. Анализ помехоустойчивости цифровых схем типа "домино" // Информационные технологии и вычислительные системы. Москва. 2004. №10. С. 2-7.
11. Bobba S., Hajj I.N. Estimation of maximum current envelope for power bus analysis and design // Int. Symp. on Phys. Des. 1998. P. 141-146.
12. Chen L.-C., Gupta S.K., and Breuer M.A., A new gate delay model for simultaneous switching and its applications // in Proc. Design Automation Conference. 2001. P. 289-294.
13. Гаврилов С.В., Пирюткина Г.А., Щелоков А.Н. Метод интервальных оценок задержек и выходных фронтов библиотечных элементов нанометровых КМОП-схем // Известия ЮФУ. Технические науки. 2012. №7 (132). С. 70-76.
14. Талалай М.С., Трушин К.В., Венгер О.В. Логический синтез комбинационных схем на основе транзисторных шаблонов с регулярной топологией // Информационные технологии. 2011. №4(176). С. 2-7.
15. Muttreja A., Agarwal N., Jha N.K. CMOS Logic Design with Independent-gate FinFETs // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2007. P. 560-567.
16. Meinhardt C., Reis R. FinFET Basic Cells Evaluation for Regular Layouts // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2013. P. 1-4.
17. Mishra P., Muttreja A., Jha N.K., FinFET Circuit Design // Springer Science+Business Media LLC. 2011. P. 23-54.
18. Datta A., Goel A., Cakici R.T. Modeling and Circuit Synthesis for Independently Controlled Double Gate FinFET Devices // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2007. P. 1957-1966.
19. Huang X., Lee W.-C., Kuo C., Hisamoto D., etc. Sub 50-nm FinFET: PFET // in Tech. Dig. IEDM, Washington, DC. 1999. P. 67-70.
20. Collinge J.P. FinFETs and Other Multi-Gate Transistors. New York: Springer, 2008. P. 350.
21. Liu Y.X., Matsukawa T., Endo K., Masahara M., etc. Cointegration of high-performance tied-gate three-terminal FinFETs and variable threshold-voltage independent-gate four-terminal FinFETs with asymmetric gate-oxide thicknesses // IEEE Electron Device Lett. Jun. 2007. Vol. 28. No 6. P. 517-519.
22. Agostinelli M., Alioto M., Esseni D., Selmi L. Leakage-delay tradeoff in FinFET logic circuits: A comparative analysis with bulk technology // IEEE Trans. Very Large Scale Integr. (VLSI) Syst. Feb. 2010.

Vol. 18. No 2. P. 232–245.

23. *Tawfik S.A., Kursun V.* Low-power and compact sequential circuits with independent-gate FinFETs // IEEE Trans. ElectronDevices. Jan. 2008. Vol. 55. No 1. P. 60–70.

24. *Cakici R.T., Roy K.* Analysis of options in double-gate MOS technology: A circuit perspective // IEEE Trans. ElectronDevices. Dec. 2007. Vol. 54. No 12.

P. 3361–3368.

25. *Anil K.G., Henson K., Biesemans S., Collaert N.* Layout density analysis of FinFET // in Proc. ESSDERC. 2003. P. 139–142.

26. *Гаврилов С.В., Гудкова О.Н., Щелоков А.Н.* Логико-временной анализ нанометровых схем на основе интервального подхода // Известия ЮФУ. Технические науки. 2012. №7 (132). С. 85-91.

УДК 681.5

А.И. Таганов, Д.А. Жуков, В.Г. Псоянц

МАТЕМАТИЧЕСКАЯ ФОРМУЛИРОВКА ЗАДАЧ КОНТРОЛЯ И СОКРАЩЕНИЯ ПРОЕКТНЫХ РИСКОВ МАЛЫХ КОСМИЧЕСКИХ АППАРАТОВ ФОРМАТА CubeSat

Рассмотрен подход к формулировке и решению задач сокращения технических рисков проектов наноспутников и представлен контекст формализованного описания процесса контроля технического состояния космического аппарата формата CubeSat.

Ключевые слова: космический аппарат, контроль технического состояния, математическая модель, риск проекта, процесс сокращения рисков.

Введение. Зарождение нового направления космических аппаратов – микро/нано/пикоспутников начинается с середины 90-х годов, когда эксперты и аналитики многих стран и в том числе Агентства новейших оборонных исследований МО США DARPA (Defence Advanced Research Project Agency) совместно с головным НИИ ВВС США AF JSR (Air Force Office of Scientific Research) пришли к выводу о необходимости создания новых космических средств на основе последних достижений микроэлектроники [1,2].

К настоящему времени в мире наметился устойчивый спрос на малые космические аппараты (МКА) формата микро/нано/пикоспутников, перспектива практического применения которых может быть самой разнообразной [1-3]:

- исследование многих космических систем и в том числе систем связи и программного обеспечения различных производителей;

- дистанционное зондирование Земли по различным прикладным направлениям и в том числе для экологического мониторинга, контроля сельскохозяйственных и лесных угодий и др.;

- отработка космических технологий, связанных с калибровкой радиолокационных станций, оптических систем контроля космического пространства и др.;

- использование в образовательных, научных и оборонных целях.

Для упрощения технологий создания и целе-

вого применения микро/нано/пикоспутников в 1999 году Калифорнийским политехническим и Стэнфордским университетами были разработаны спецификации CubeSat [4], которые определили унифицированный формат малых искусственных спутников Земли для исследования космоса. Например, согласно стандарту наноспутники должны иметь размер 10x10x10 см и массу не более 1,5 кг. При этом стандарт допускает объединение двух и более стандартных кубов в составе одного более крупного МКА.

Стандартизация наноспутников CubeSat позволила разработчикам и производителям МКА создать по всему миру широкий рынок доступных комплектующих и материалов [4,5] для многочисленной аудитории университетов и малых инновационных предприятий, желающих включиться в перспективную разработку и создание как промышленных и коммерческих, так и учебных малых космических аппаратов.

Существующая проблема относительно низкой надежности и эффективности целевого функционирования МКА формата Cubesat на космической орбите требует решения целого комплекса задач на всех этапах жизненного цикла (ЖЦ) МКА с целью повышения их надежности, снижения технических рисков самих проектов МКА и повышения эффективности систем контроля технического состояния (КТС) космических аппаратов.

Цели работы: разработка формализованного подхода к формулировке и решению задач сокращения технических рисков проектов МКА и постановка задач контроля МКА в контексте основных этапов технического контроля и математических моделей представления процесса контроля.

1. Этапы и задачи процесса сокращения технических рисков проекта МКА. Практика инженерии космических проектов свидетельствует, что такие проекты являются достаточно сложными в техническом и в организационном выполнении. В жизненном цикле этих проектов присутствуют факторы неопределенности и, как следствие, возникают технические риски, которые могут приводить к отклонениям параметров проектов от запланированных технических характеристик МКА. В каждом космическом проекте технические риски рассматриваются как негативные последствия функционирования и применения МКА, которые способны вызвать ущерб космической миссии в результате отклонения контролируемых характеристик наноспутников от заданных требований заказчика, согласованных с разработчиками.

К настоящему времени разработано несколько общих и частных моделей для анализа и сокращения рисков в жизненном цикле сложных технических проектов [6-10]. Каждая из частных моделей имеет свои особенности, обусловленные свойствами и характеристиками объектов разработки. Модели отличаются спецификой интересов и квалификацией их авторов и охватывают широкий спектр реальных ситуаций проектов, в которых необходимо сокращение или исключение рисков.

По результатам анализа этих подходов и моделей управления рисками проектов в работе рассматривается общая концептуальная схема процесса сокращения технических рисков проектов МКА, которая отражает основные этапы и задачи процесса и необходимые процедурные действия.

Процесс сокращения технических рисков проекта МКА в общем случае может содержать следующие этапы.

Этап 1. Подготовка исходных данных для анализа, прогнозирования и управления техническими рисками проекта МКА включает задачи:

- описание системы, внешней среды и программной системы;
- определение целей, назначения и функций проекта;
- разработка предварительных требований к функциональной пригодности и конструктив-

ным характеристикам проекта МКА;

- формирование группы экспертов для анализа угроз и управления рисками.

Этап 2. Выделение, идентификация, анализ угроз и рисков проекта МКА включает задачи:

- выделение источников и угроз нарушения требований и ограничений ресурсов, определение критериев работоспособности проекта;
- отбор и идентификация основных угроз и рисков проекта;
- анализ причин, выделение категорий угроз и возможных последствий появления рисков функциональной пригодности проекта;
- анализ причин, выделение категорий угроз и возможных последствий проявления рисков конструктивных характеристик проекта;
- анализ причин, выделение категорий угроз и возможных последствий рисков проекта.

Этап 3. Оценивание технических рисков проекта МКА и выбор контрмер для их сокращения включает задачи:

- оценивание возможных последствий, уровней потенциальных опасностей угроз и приоритетов категории рисков проекта;
- выделение и упорядочение группы наиболее опасных, высокоприоритетных рисков проекта;
- планирование методов и ресурсов реализации контрмер для сокращения опасных, приоритетных рисков проекта;
- распределение ресурсов на контрмеры для сбалансированного сокращения интегрального риска проекта;
- распределение ответственности специалистов за реализацию сокращения опасных рисков проекта.

Этап 4. Сокращение или ликвидация опасных рисков проекта МКА включает задачи:

- реализация контрмер для сокращения интегрального риска и составление отчетов о состоянии проекта;
- корректировка требований к функциональной пригодности, конструктивным характеристикам и ограничениям ресурсов проекта;
- регистрация результатов сокращения интегрального риска на очередном этапе проекта.

Этап 5. Контроль, регистрация, мониторинг и утверждение допустимого интегрального риска проекта МКА включает задачи:

- контроль, отслеживание и мониторинг реализации сокращения интегрального риска по этапам проекта;
- мониторинг состояния проекта и интегрального риска по этапам жизненного цикла проекта;
- документирование и утверждение допус-

тимого интегрального риска по этапам жизненного цикла проекта;

- оформление итоговых данных по результатам сокращения технических рисков проекта.

Согласно предложенной схеме в отдельных процессах анализа, идентификации, планирования, контроля, мониторинга, сокращения и документирования технических рисков проекта МКА циркулирует рисковая информация, представленная как количественными, так и качественными значениями параметров проекта. Экспертные оценки и вербальные высказывания специалистов по ходу жизненного цикла проектов МКА отражают свойство нечеткости проектных данных для принятия управленческих решений по техническим рискам.

Возникает объективная необходимость в постанове и решении задач по разработке формализованных подходов, моделей и методов, реализующих нечеткие процедуры переработки сложной рискованной информации в удобную для специалистов форму, позволяющую более качественно и оперативно решать задачи управления техническими рисками проектов МКА.

2. Математическая формулировка задач процесса сокращения технических рисков проекта МКА. В соответствии с представленным выше вербальным описанием основных этапов и задач процесса сокращения технических рисков проекта МКА выделим здесь в число наиболее интересных для формализации задач – задачи идентификации и анализа рисков проекта.

2.1. Задача идентификации технических рисков проекта. Идентификация технических рисков проекта МКА должна быть ориентирована на выявление потенциальных технических рисков проекта и документирование их характеристик. Область определения этой задачи может быть представлена кортежем [13]

$$\langle Q, H, R, G, A, L \rangle, \quad (1)$$

где Q – заданное множество исходных данных (описание внешней среды проекта, описание внутренней среды проекта, описание отклонений в процессе реализации проекта, описание альтернатив реализации проекта и др.); H – заданная модель технических характеристик проекта; $R = \{R_1, R_2, \dots, R_n\}$ – искомое множество технических рисков; G – граф связности элементов множества R ; A – множество альтернатив реагирования на идентифицированное множество R ; L – набор методов и средств, ориентированных на определение множеств R , A , и построение графа G .

Успешность решения задачи идентификации

рисков проекта с областью определения (1) во многом зависит от используемого набора методов, входящих в состав множества L .

В инженерной практике такой набор методов обычно включает в себя: метод мозгового штурма, метод Дельфи, метод построения дерева решений, анализ SWOT, анализ контрольных списков, метод графического отображения моделей рисков с помощью диаграмм и др. [6,7]. Однако указанные методы являются слабо формализованными, трудоемкими и соответственно субъективными по отношению к качеству принимаемых решений в условиях нечеткости проектных данных по стадиям ЖЦ проекта.

Для решения задачи с областью определения (1) предлагается дополнительно взять за основу подходы, которые ориентированы на идентификацию технических рисков проекта в условиях нечеткости [9,13]. Это позволяет на формализованной основе построить соответствующие алгоритмы и инструментальные средства эффективной поддержки процесса идентификации рисков проекта.

2.2. Задача анализа технических рисков проекта МКА. Анализ технических рисков проекта заключается в ранжировании по важности всех идентифицированных рисков анализируемого проекта с последующим выделением по заданным критериям некоторой совокупности технических рисков, которые передаются на следующие этапы планирования и мониторинга этих рисков.

В условиях нечеткости исходных данных предлагаемый здесь подход к решению указанной задачи нечеткого анализа технических рисков проекта предполагает решение двух слабо структурированных задач [11-14]:

- задача 1. Определение вектора степеней влияния идентифицированных рисков на обобщенный (интегральный) критерий проекта;

- задача 2. Оптимизация состава контролируемых технических рисков, передаваемых на следующие этапы планирования и мониторинга рисков проекта.

Область определения первой задачи может быть представлена следующим кортежем:

$$\langle R(t), H(t), f_s, P_s, G(t), W(R(t)), Z \rangle, \quad (2)$$

где $R(t)$ – множество идентифицированных процессных рисков проекта на момент времени t ; $H(t)$ – модель технических характеристик МКА на текущем этапе проекта; f_s – отображение $R(t)$ в $H(t)$, определяющее оценки влияния последствий $R(t)$ на технические характеристики проекта; P_s – структура предпочтений спе-

специалистов (экспертов) по рискам; $G(t)$ – граф связности рисков; $W(R(t))$ – искомый вектор степеней влияния последствий $R(t)$ на обобщенный критерий проекта; $Z = \{Z_1 \div Z_5\}$ – набор предлагаемых подходов к построению алгоритмов определения вектора $W(R(t))$ на основе представленных на определенном этапе проекта исходных данных.

Для решения первой задачи с областью определения (2) предлагается исследовать следующие подходы и способы построения набора алгоритмов $Z = \{Z_1 \div Z_5\}$:

- алгоритм Z_1 , позволяющий определять (по стадиям ЖЦ проекта) вектор $W(R(t))$ на основе следующих исходных данных: модель $H(t)$ задана множеством независимых технических характеристик проекта МКА; $R(t)$ содержит множество независимых технических рисков; f_s и P_s представлены экспертным способом на основе количественных оценок;

- алгоритм Z_2 , отличающийся от Z_1 тем, что экспертные оценки специалистов по рискам заданы вербальным способом;

- алгоритм Z_3 , отличающийся от Z_1 и Z_2 тем, что модель $H(t)$ представлена деревом зависимых технических характеристик проекта;

- алгоритм Z_4 , позволяющий на основе данных, полученных при использовании одного из алгоритмов Z_1, Z_2, Z_3 , учитывать наличие однозначной связности рисков $R(t)$ по стадиям ЖЦ проекта;

- алгоритм Z_5 , отличающийся от Z_4 тем, что дополнительно позволяет учитывать субъективную связность рисков $R(t)$ проекта.

В основу обоснованного построения рассматриваемых алгоритмов $Z = \{Z_1 \div Z_5\}$ предлагаются прикладные методы теории нечетких множеств (ТНМ), модифицированный метод анализа иерархий и многокритериальные модели принятия решений в условиях неопределенности [13,14,16].

Вторая задача анализа рисков, связанная с определением совокупности контролируемых рисков $R(t)^* \subset R(t)$ по критерию их наибольшего суммарного влияния на обобщенный показатель проекта, имеет следующую область определения [14]:

$$\langle R(t), W(R(t)), P_r(t), B \rangle, \quad (3)$$

где $R(t)$ и $W(R(t))$ – соответственно заданное

множество технических рисков проекта и заданный вектор степеней влияния этих рисков на обобщенный критерий проекта; $P_r(t)$ – структура предпочтений экспертов по ресурсам проекта, которая ставит в соответствие каждому риску из $R(t)$ требуемые ресурсы на управление этим риском; $B = \{B_1, B_2\}$ – набор алгоритмов, необходимый для решения задачи по определению $R(t)^*$ с учетом заданных условий и ограничений.

Для решения задачи с областью определения (3) предлагаются два подхода к построению алгоритмов определения $R(t)^* \subset R(t)$.

Первый подход связан с построением алгоритма B_1 , ориентированного на решение задачи с областью определения (3) в том случае, когда в исходных данных присутствуют условия нечеткости процессных данных и отсутствуют жесткие ресурсные ограничения на управление техническими рисками проекта.

Второй подход отражает построение алгоритма B_2 , отличающегося от алгоритма B_1 возможностью решения задачи с областью определения (3) при наличии в исходных экспертных данных информации по требуемым ресурсам для управления проектными рисками.

Для построения алгоритмов B_1 и B_2 предлагается исследовать подходы, основывающиеся на использовании положений ТНМ и методов многокритериального выбора альтернатив в условиях неопределенности. Опыт применения указанных теорий для задач анализа рисков программных изделий позволяет адаптировать известные методы и алгоритмы [12,14,16] к задачам анализа технических рисков проекта МКА.

Сформированный такими способами оптимизированный состав $R(t)^*$ используется далее на этапе мониторинга и сокращения технических рисков проекта МКА.

3. Математическая формулировка задач контроля технического состояния МКА. Контроль технического состояния (КТС) на этапах ЖЦ МКА заключается в определении совокупности таких признаков, по которым можно судить о функциональной пригодности МКА.

Решение указанной задачи рассматривается здесь в контексте представления области определения задач КТС МКА в виде

$$M_{KTC} = \langle M_{КА}, U, M_{МПК} \rangle, \quad (4)$$

где $M_{КА}$ – модель объекта контроля, т.е. МКА;

$U = \{U_i\}, i = \overline{1, n}$ – множество этапов процесса КТС;

$M_{\text{МПК}}$ – модель организации процесса КТС.

Математическая модель $M_{\text{КА}}$ может быть представлена здесь в наиболее общем виде, как это принято в известной теории контроля и управления сложными системами [15,16]

$$M_{\text{КА}} = \langle T, X, Y, Z, F, L \rangle, \quad (5)$$

где T – множество моментов времени, в которые наблюдается МКА; X, Y – множества входных и выходных сигналов соответственно; Z – множество состояний МКА; F – оператор переходов, отражающий механизм изменения состояния объекта под действием внутренних и внешних возмущений; L – оператор выходов, описывающий механизм формирования выходного сигнала как реакции объекта на внутренние и внешние возмущения. Эти операторы реализуют отображения

$$F: T \times X \times Z \rightarrow Z, \quad (6)$$

$$L: T \times X \times Z \rightarrow Y. \quad (7)$$

Каждое техническое состояние МКА $z \in Z$ характеризуется в каждый момент времени $t \in T$ набором переменных z_r , ($r = \overline{1, k}$), изменяющихся под влиянием воздействий и внутренних возмущений.

$U_1 \in U$ в модели (4) обозначает задачу наблюдения МКА, которая состоит в том, чтобы на основе известного выходного процесса $y(t) \in Y$ однозначно определить неизвестные состояния МКА $z(t) \in Z$, где $y(t)$ и $z(t)$ – вектор-функции. Эта задача сводится к решению относительно $z(t)$ уравнения [15]:

$$L[t, x(t), \hat{z}(t), \tau] = \hat{y}(t), \quad (8)$$

где $\hat{y}(t)$ – некоторая реализация выходного процесса, доступная регистрации с помощью измерительных устройств. Космический аппарат в техническом плане может считаться наблюдаемым в состоянии $z(t) \in Z$ на множестве моментов времени $\hat{T} = \{\tau\}$, $\hat{T} \subset T$ при входном воздействии $x(t) \in X$ и отсутствии возмущений, если уравнение (8) имеет единственное решение $\hat{z}(t) = z(t) \in Z$. При этом если утверждение справедливо для любого $z(t) \in Z$, то МКА считается полностью наблюдаемым.

Таким образом, *первым этапом* реализации процесса КТС на основе применения модели (4) является решение задачи наблюдения, т.е. определение такого отображения (7), которое при фиксированных значениях $t \in T$ и $x(t) \in X$ обеспечивает выполнение условий полной наблюдаемости МКА.

Другая задача $U_2 \in U$ в модели (4) обозначает задачу классификации, т.е. отнесения наблю-

даемого состояния МКА к одному из заданных видов его технического состояния. Решение этой задачи является *вторым этапом* реализации процесса КТС и заключается в отыскании отображения

$$\eta: Y \rightarrow E, \quad (9)$$

где E – множество заданных видов технического состояния МКА. При этом каждому виду технического состояния соответствует определенное подмножество его текущих состояний, объединенных некоторыми общими свойствами, о которых может быть принято одно и то же решение. При этом отдельные состояния, входящие в эти подмножества, должны находиться в отношении эквивалентности $Q = Y \times Y$ и обладать свойствами:

рефлексивности $\forall y \in Y, (y, y) \in Q$;

симметричности $(y_1, y_2) \in Q \Rightarrow (y_2, y_1) \in Q$;

транзитивности

$$(y_1, y_2) \in Q \cap (y_2, y_3) \in Q \Rightarrow (y_1, y_3) \in Q.$$

Т.е. отношение эквивалентности задает разбиение множества Y всех состояний МКА на непересекающиеся классы, т.е. осуществляет факторизацию этого множества в виде отображения

$$\vartheta: Y \rightarrow Y/Q, \quad (10)$$

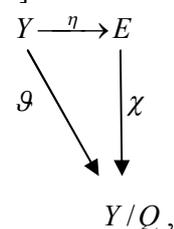
где Y/Q – фактор-множество класса эквивалентности.

Отображение (10) по своему смыслу является наложением и отсюда, по существу контекста модели (4), множество E видов технического состояния и фактор-множество Y/Q должны находиться во взаимно однозначном соответствии, т.е. чтобы отображение

$$\chi: E \rightarrow Y/Q, \quad (11)$$

было взаимно однозначным. При этом согласно теореме о гомоморфизме для множеств, взаимная однозначность отображения (11) может достигаться, когда выполняется условие $\eta\chi = \vartheta$, а это возможно когда отображение (10) – естественное, а отображение (9) – наложение.

В процессе выполнения задач КТС взаимосвязь между рассматриваемыми задачами и отображениями задается в модели (4) специальной схемой процесса $M_{\text{МПК}}$. В первом приближении модель этого процесса может быть представлена диаграммой [15,16]



в которой отображение $\eta: Y \rightarrow E$ есть наложение, т.е. любой элемент $e_i \in E$ является образом, по крайней мере, одному элементу $y \in Y$.

Следующий – **третий этап** реализации процесса КТС заключается в реализации следующего отображения

$$\psi: E \rightarrow S, \quad (12)$$

которое формально определенному виду технического состояния МКА E ставит в соответствие решение о его истинном техническом состоянии S с учетом вероятностных характеристик, возможных ошибок при контроле, погрешностей выполняемых измерений и помех.

Тогда общая диаграмма этапов процесса КТС примет вид [15,16]

$$\begin{array}{ccccccc} T \times X \times Z & \xrightarrow{L} & Y & \xrightarrow{\eta} & E & \xrightarrow{\psi} & S \\ & & \searrow \vartheta & & \downarrow \chi & & \nearrow \xi \\ & & & & Y/Q & & \end{array}, \quad (13)$$

Практическая реализация процесса КТС

согласно диаграмме (13) может быть представлена двумя более крупными этапами.

Этап 1. Проведение экспериментов и моделирование технических состояний МКА.

Заключается в проведении экспериментов, связанных с имитацией различных технических состояний [15,16]. При каждом воспроизведении очередного состояния МКА реализуются последовательно отображения L и ϑ .

Результатом этапа является определение фактор-множества состояний МКА Y/Q и множества его технических состояний E . Условно этот этап можно назвать этапом построения модели контролируемого МКА, а сам процесс построения модели назвать идентификацией.

Задача идентификации в прямой постановке заключается в определении оптимальной в некотором смысле оценки оператора F по реализации входных x и выходных y сигналов контролируемого МКА. Согласно (4) оператор F в явном виде не определяется, а оценивается принадлежность состояния МКА, характеризуемого оператором F , к одному из априорно заданных видов его технического состояния.

Этап 2. Контроль технического состояния МКА.

Этот этап согласно (13) заключается в определении вида реального технического состояния МКА по результатам выполнения в нем измерений. В диаграмме (13) это отображено последовательным выполнением отображений L , η и ψ . По результатам непосредственного контроля

МКА может быть произведено уточнение ранее сформированного фактор-множества Y/Q за счет реализации отображения ξ .

Следует отметить, что представленная общая постановка задачи контроля технического состояния МКА инвариантна к виду определяемых его технических состояний. Это позволяет в постановке (4) сформулировать как задачу проверки правильности функционирования МКА, так и задачу поиска дефектов в его отдельных узлах космического аппарата. При этом каждая из этих задач в соответствии с диаграммой (13) включает общие этапы, но в силу различия этих задач по смысловому содержанию и конечным целям способы разбиения множества состояний МКА на классы эквивалентности будут, очевидно, различными. Так при поиске дефектов факторизация множества Y должна производиться на множестве состояний МКА в фиксированный момент времени, а при контроле функционирования МКА факторизация должна выполняться на множестве состояний МКА в различные моменты времени.

Заключение. Представленные в работе математические формулировки задач и этапов процесса сокращения проектных рисков и вопросы организации процесса контроля технического состояния МКА обобщают и очерчивают круг проблем и задач, которые необходимо решать на этапах проектирования и контроля с целью повышения надежности и эффективности целевого функционирования МКА.

Работа выполнена при финансовой поддержке Министерства образования и науки РФ.

Библиографический список

1. [http:// www.darpa.mil](http://www.darpa.mil).
2. [http:// www.answers.com>topic/air-force-research-laboratory](http://www.answers.com>topic/air-force-research-laboratory).
3. [http:// www.vpk.name>Библиотека>Dauria-Aerospace](http://www.vpk.name>Библиотека>Dauria-Aerospace).
4. [http:// www.cubesat.ru/ru/cubesats.html](http://www.cubesat.ru/ru/cubesats.html).
5. <http://www.cubesatshop.com>.
6. ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами (Руководство РМВОК). - 388 с.
7. Лунаев В.В. Анализ и сокращение рисков проектов сложных программных средств. – М.: СИНТЕГ, 2005. – 224 с.
8. Таганов А.И., Таганов Р.А. Метод определения важности субъективно связанных рисков качества программных проектов // Вестник Рязанского государственного радиотехнического университета. 2002. № 10. С. 59 – 63.
9. Таганов А.И., Таганов Р.А. Методологические основы методов идентификации рисков событий

проекта // Вестник Рязанского государственного радиотехнического университета. 2003. № 12. С. 70 – 77.

10. Таганов А.И., Таганов Р.А. Метод определения оптимальной альтернативы реагирования на этапе мониторинга рисков проекта // Вестник Рязанского государственного радиотехнического университета. 2003. № 11. С. 115 – 118.

11. Корячко В.П., Таганов А.И., Таганов Р.А. Программный метод управления рисками качества проекта информационной системы // Известия Белорусской инженерной академии, 2004. № 1. С. 168 – 179.

12. Везенов В.И., Таганов А.И., Таганов Р.А. Применение процедуры нечеткого вывода для анализа рисков программного проекта // Системы управления и информационные технологии. 2006. № 2 (24). С. 34– 39.

13. Таганов А.И. Применение нечетких множеств для формализации процессов анализа и идентификации важности рисков программного проекта // Системы управления и информационные технологии. 2007. № 4 (30). С. 46 – 51.

14. Таганов А.И. Методика анализа и сокращения рисков проектов сложных программных систем по характеристикам качества // Вестник Рязанского государственного радиотехнического университета. 2010. № 31. С. 77 – 82.

15. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. – Л.: Энергоатомиздат, 1988. – 192 с.

16. Корячко В.П., Скворцов С.В., Таганов А.И., Шибанов А.П. Эволюция автоматизированного проектирования электронно-вычислительных средств // Радиотехника, 2012. № 3. С. 97 – 103.

УДК 518.681

С.И. Перевозников, В.В. Колодный, В.С. Озеранский

ФОРМИРОВАНИЕ КОНТРОЛЕПРИГОДНЫХ СТРУКТУР ЦИФРОВЫХ УСТРОЙСТВ ДЛЯ СИСТЕМ ПОКОМПОНЕНТНОГО ДИАГНОСТИРОВАНИЯ

Проанализированы особенности формирования искусственных контролепригодных структур с помощью внесения в состав схемы цифрового устройства (ЦУ) временных связей между внутренними узлами. Выявлены условия реструктуризации схемы ЦУ, которые приспособлены к организации процесса тестирования.

Ключевые слова: *декомпозиция, компонент, цифровое устройство, диагностика, тестирование.*

Введение. Среди известных систем диагностирования цифровых устройств определенное место занимают средства внутрисхемного тестирования. Особенность таких систем заключается в том, что возможность физического контакта с внутренними узлами [контрольными точками (КТ)] цифровых устройств дает возможность реализовывать разнообразные стратегии направленного поиска неисправностей. Конструктивный доступ аппаратных средств к внутренним узлам печатных плат лежит в основе композиционного подхода относительно формирования эффективных компонентных структур их разбивки. Таким образом, при создании независимых электрических условий защиты элементов за счет временной коммутации многих внутренних КТ, создаются фрагменты (компоненты), которые наиболее приспособлены к диагностированию особенно во время массового производства цифровых устройств (ЦУ), где доминирующим фактором становится время их качествен-

ного изготовления. Подобные вопросы рассматривались в работе [1]. Соответствующим образом формируются, например, генерирующие или параллельные структуры [2, 3]. Первые экономят объем машинной памяти для хранения тестов, вторые – экономят время диагностирования созданных фрагментов за счет меньшего числа повторных вызовов именно процедур поиска неисправностей.

Известно, что аппарат графов (а именно – гиперграфов) наиболее точно описывает особенности декомпозиционного представления сложных объектов исследования с целью следующего их пошагового тестирования. Взаимоотношения сформированных компонентов (например, их сложность, характер пересечения, электрическая изолированность подструктур и тому подобное) значительным образом влияют на порядок проведения поисковых процедур, а также формируют основания для выбора эффективных стратегий диагностирования. Если взять за основу мо-

дель разбивки гиперграфа G цифрового устройства, предложенную, например, в работе [3], то особенности такого анализа (выбора) конечных компонентных структур объясняются комбинаторным механизмом их оценивания.

Цель исследований. Целью исследования является автоматизация создания искусственных подструктур и их преобразование в процессе формирования тестопригодных компонентов ЦУ. Такие преобразования удобно выполнять, используя аппарат графов, в частности гиперграфов, которые наиболее точно описывают структуру схемы ЦУ и позволяют эффективно использовать свои моделирующие свойства.

Постановка задачи. Одной из задач, которые нужно решать в процессе диагностирования устройств, является задача разбивки схем ЦУ на определенные составные части (компоненты), которые позволяют сократить общее время поиска неисправности в таких устройствах. Но задачи временной композиции схем ЦУ формально сводятся к введению в структуру схемы ЦУ искусственных связей со стороны средств диагностирования. Анализ современных методов диагностирования показал, что не существует стандартных способов разбивки схемы ЦУ, которые упрощают и убыстряют процесс поиска неисправностей.

Задачей данной статьи является усовершенствование методов реструктуризации схемы ЦУ, которые базируются на использовании аппарата гиперграфов.

Решение задачи. Одним из основных преимуществ графовых моделей, например цифровых устройств, есть наглядность отображения структурных особенностей самих объектов. Причем способность их компьютерного представления является основой для следующего моделирования с целью решения разных оптимизационных (поисковых) задач технической диагностики.

Установлено, что именно аппарат теории гиперграфов имеет соответствующие моделирующие свойства [2, 3], которые можно привлекать как для описания и формирования подструктур тестирования ЦУ, так и во время подготовки процедур их тестирования. При этом учитывается, что процесс создания любых подструктур должен базироваться на ряде формальных предположений в самих моделях (например, особенностей установки начальных условий диагностирования), а также ограничением удержания электрических режимов разбивки цифровых устройств [4].

При этом основные характеристики декомпозиции любого объекта диагностирования свя-

заны соответствующими зависимостями [5], что составляет именно аппарат формирования (моделирования) компонентных структур, который предусматривает расчет как числовых показателей, так и такое размещение (соединение) компонентов, которое дает минимальное значение времени диагностирования всего устройства в целом.

Приведем ряд определений для типов фрагментов и некоторых компонентных образований λ -разбиения, а также их характеристики [5].

Последовательность простых компонентов, которые связывают сложные фрагменты, назовем внутренней простой цепью (ВПЦ).

Цепь простых компонентов, которая связывает сложный и терминальный компоненты, назовем полупростой цепью (ППЦ).

Моделирование структур разбиения графа G (с целью оптимизации) удобно осуществлять на основе α -операций [5]. При этом формируются процессы коррекции графа.

ST-процессом будем считать последовательность α -операций перемещения простых компонентов из s -цепей в t -ветви графа G .

TS-процессом будем считать последовательность α -операций перемещения простых компонентов из t -ветвей в s -цепи графа G .

TS-процесс имеет центростремительное направление перемещения простых компонентов, с которыми связано устранение терминальных ветвей. Максимизация или минимизация числа терминальных компонентов связаны с ST- или TS-процессами при неизменных параметрах самого разбиения.

Не нарушая общности, рассмотрим алгоритм расчета числовых показателей формирующейся компонентной структуры.

Алгоритм А1

Шаг 1. Задать параметр λ -разбиения.

Шаг 2. Если генерация векторов L сложностей компонентов закончилась, то идти к шагу 20.

Шаг 3. Сгенерировать вектор $L = (l_1, l_2, \dots, l_h)$

где $2 \leq h \leq \left\lfloor \frac{\lambda}{2} \right\rfloor + 1$, $l_k \in \{0, 1\}$, а также $k = \overline{1, h}$.

Шаг 4. Если значения показателя пересечений закончились, то идти к шагу 2.

Шаг 5. Задать значение Δ ($\frac{\lambda}{2} \leq \Delta \leq \lambda$), а также подсчитать характеристику $N = \lambda + \Delta$ и показатель $m = \sum_k l_k$.

Шаг 6. Если варианты разбиения числа N закончились, то идти к шагу 4.

Шаг 7. Сгенерировать разбиение

$\varphi(N, m) = (A_1, A_2, \dots, A_m)$.

Шаг 8. Выполнить изображение $\psi : I \rightarrow J$

где $\psi(i) = j$ а также $I = \{i | i \in \overline{1, m}\}$,

$$J = \{j | j \in \overline{1, \lfloor \frac{\lambda}{2} \rfloor + 1}\}.$$

Шаг 9. Если для всех $i \in \overline{1, m}$ выполняется

условие $\lfloor \frac{A_i}{\psi(i)} \rfloor = \frac{A_i}{\psi(i)}$, то определить для всех i

показатели $a_i = \frac{A_i}{\psi(i)}$, а также подсчитать

$s = \Delta - a_2, p = a_1 + a_2 + \dots + a_m$, перейти к шагу 10, иначе к шагу 6.

Шаг 10. Сформировать упорядоченный вектор разбивки подструктур

$$((\underbrace{m, m, \dots, m}_{a_m}), (\underbrace{m-1, m-1, \dots, m-1}_{a_{m-1}}), \dots, (\underbrace{1, 1, \dots, 1}_{a_1})) =$$

$$= (A_m^1, A_{m-1}^1, \dots, A_1^1).$$

Шаг 11. Реализовывать алгоритм А1 до нулевого результата вычитания векторов связей.

Шаг 12. Определить для каждой пары

A_i^1, A_j^1 характеристику $\delta_2^{ij} = s^{ij} - 1$, где $i, j \in \overline{1, m}$.

Шаг 13. Подсчитать δ_2 для всей структуры

$$\text{разбиения } \delta_2^1 = \frac{1}{2} \sum_i \sum_j \delta_2^{ij}.$$

Шаг 14. Если $a_2 < \delta_2^1$, то идти к шагу 6.

Шаг 15. Если число пар k связанных между собой компонентов V_i и V_j (где $|V_i|, |V_j| = 1, a i, j \in \overline{1, m}$) $k \geq 2$, то идти к шагу 6.

Шаг 16. Если $k = 1$, то откорректировать структуру по типу 1, а также подсчитать $\Delta := \Delta - 1$ $p := p - 1$.

Шаг 17. Если число пар k связанных между собой компонентов V_i и V_j (где $|V_i|, |V_j| = 2, a i, j \in \overline{1, m}$) $k > 0$, то откорректировать структуру по типу 2. Подсчитать $\Delta := \Delta - k$ $\delta := \delta - k$, иначе идти к шагу 6.

Шаг 18. Если $z = \sum_i V_i > 0$ (где $|V_i| \leq 2, i \in \overline{1, m}$), связанные со сложными компонентами, то откорректировать структуру по типу 3 и 4. Образовать $k \geq 0$ ($k \leq \min\{a, z\}$) полных фрагментов схем.

Шаг 19. Подсчитать $\delta_a = a - k$ $\delta_2 = \delta - \delta_a$. Зафиксировать структуру. Устранить избыточные пересечения. Перечислить основные характеристики структуры разбиения. Идти к шагу 21.

Шаг 20. Структуры не существует.

Шаг 21. Конец алгоритма.

На рисунке 1 приводится пример расчета компонентной структуры, который начинается с определения глубины диагностирования (λ) устройства.

$$\lambda = p + \Delta - \delta, N = \sum_{i=1}^z a_i \cdot i,$$

где дополнительно приходится соотношение

$$N = p + 2 \cdot \Delta - \delta = \lambda + \Delta.$$

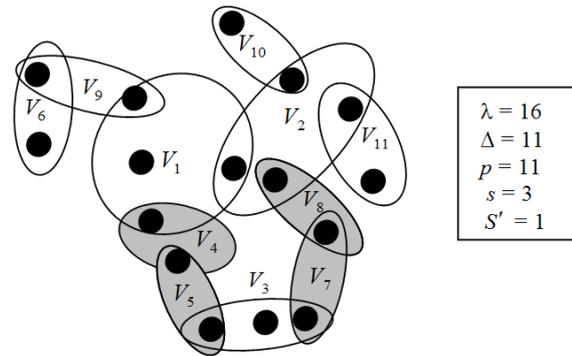


Рисунок 1 – Пример компонентной структуры при $\lambda = 16$

Для примера, который приведен на рисунке 1, получим равенство $16 = 11 + 11 - 6$, а сам ряд имеет значение $N = 2 \cdot 8 + 3 \cdot 1 + 4 \cdot 2 = 27$. Межкомпонентные связи (их число) определяются зависимостью $s = \Delta - a_2 = 11 - 8 = 3$. На рисунке 1 такие связи составляют следующие подмножества компонентов разбиения: $\{V_4, V_5\}$ и $\{V_7, V_8\}$. Непосредственное пересечение фрагментов $\{V_4 \cap V_5\}$ составляет вырожденную цепь (одноэлементную). Совместимое моделирование характеристик s, t, δ , а также процесс p -разбиения дает, при $N = const$, основу относительно формирования разных структур и, как следствие, – выбор оптимального покрытия множественного числа исходных элементов устройства, степень декомпозиции которого представляется следующей суммой:

$$p = \sum_i a_i = a_1 + a_2 + \dots + a_z = 8 + 1 + 2 = 11.$$

При разбиении графа G с параметрами $\langle p, \Delta, \delta \rangle$ число t^* терминальных компонентов находится в границах $[5, 6]$

$$t_{max} = t + \min\{a - \delta_a, \delta_2\} \geq t^* \geq t_{min} = t - \min\{\delta_a, a_2 - \delta_2\},$$

где t – текущее число терминальных компонентов, которое задается при генерации параметров декомпозиции графа G , a – число сложных компонентов разбиения ($p = a + a_2$).

При этом следует заметить, что характеристики полноты компонентов (δ_a и δ_2 , где

$\delta = \delta_a + \delta_2$) представляют собой взаимозависимые величины, пример которых показан на рисунке 1. Пример пошагового формирования (при $\lambda = 16$) максимального числа терминальных ветвей (t_{max}) структуры разбиения графа G показан на рисунке 2.

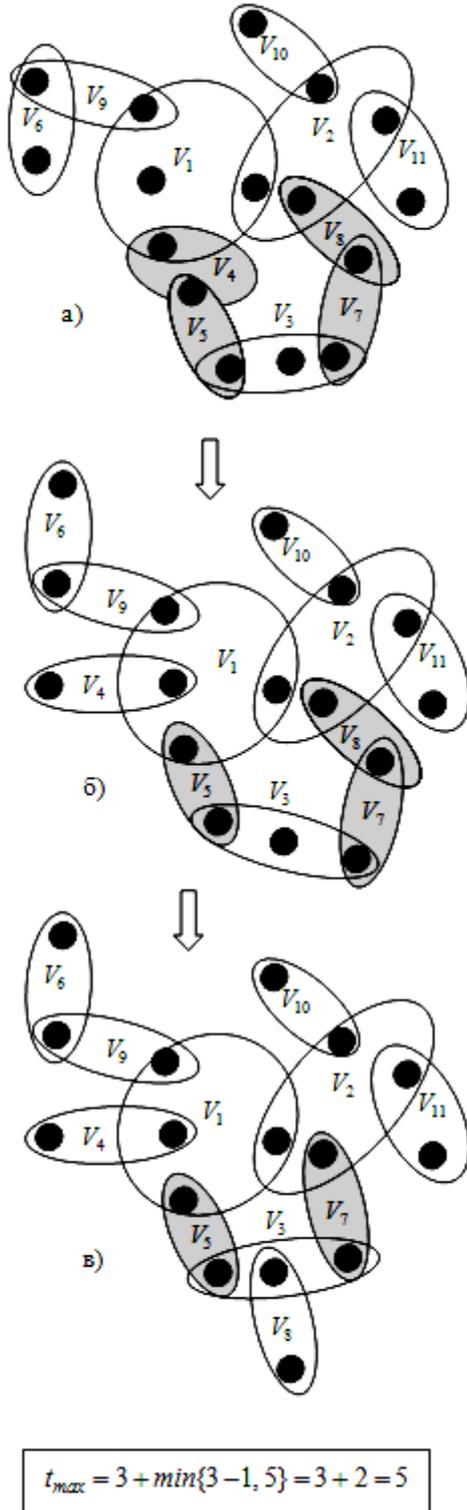


Рисунок 2 – Формирование максимального числа терминальных ветвей графа G

Следует заметить, что формирование терминальных ветвей осуществляется последователь-

ностью α -операций (ST-процесс). На каждом шаге ST-процесса уменьшается на единицу значения δ_2 и формируется одна терминальная ветвь. Известно, что между любыми двумя сложными компонентами число непосредственных пересечений не должно превышать единицу. Общее число связей между каждой парой связанных сложных компонентов составляет S (как результат разбиения).

Например, для структуры, которая представлена на рисунке 1, процесс преобразования состоит из 2-х шагов.

Вначале (рисунок 2, а) фиксируем:

$$t_{max} = 5, \text{ а } t = 3.$$

На первом шаге (рисунок 2, б) имеем:

$$t_{max} = 5, \text{ а } t = 4.$$

На втором шаге (рисунок 2, в) имеем:

$$t_{max} = 5, \text{ и } t = 5.$$

Прекращаем преобразование структуры (при достижении t_{max}).

Пример пошагового формирования (при $\lambda = 16$) минимального числа терминальных ветвей (t_{min}) структуры разбиения графа G приведен на рисунке 3.

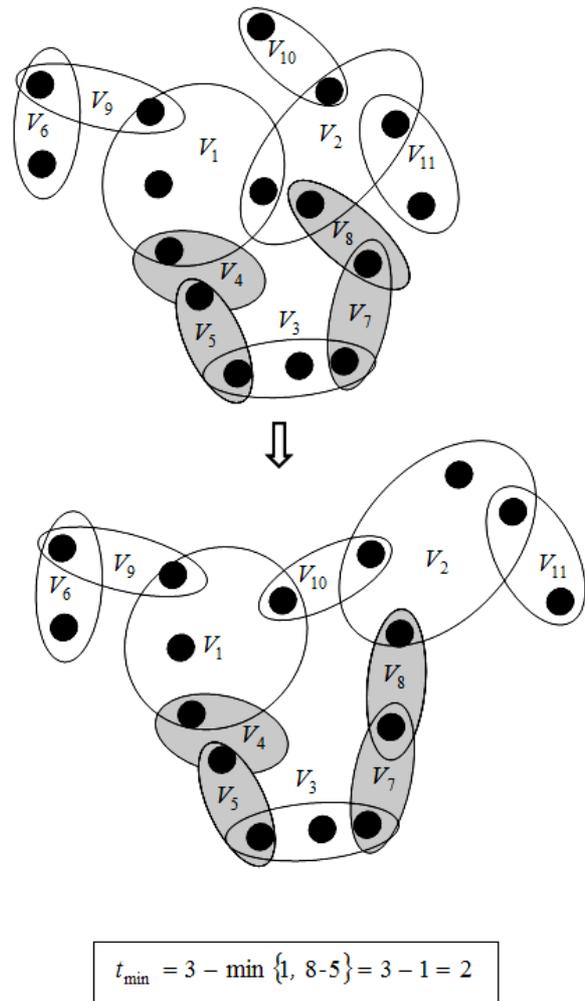


Рисунок 3 – Формирование минимального числа терминальных ветвей графа G

Поиск конечной структуры разбиения, как правило, сводится к выбору таких сформированных подмножеств элементов, при которых фиксируется меньшее время диагностирования всего ЦУ.

Причем незначительное изменение любой структуры непосредственно влияет на общее время поиска неисправностей при одинаковых параметрах разбиения.

Моделирование таких процессов основывается на правилах «перемещения» простых компонентов по цепям ВПЦ и ПЦ подструктур, а также формирования или устранения терминальных ветвей графа G в границах $t_{max} - t_{min}$.

Заключение. Формирование полных компонентов целесообразно связывать с тем, что ядра в таких структурах ставятся в соответствие элементам, которые имеют относительно длительное время тестирования и проверяются в последнюю очередь или решение принимается способом исключения (без их тестирования). Существование терминальных ветвей эффективно, когда тесты элементов, что их образуют, согласованы между собой по входам и выходам. Примером такой структуры является последовательность триггеров (регистры). Стоит отметить, что при моделировании таких искусственных структур возможны случаи, когда появляется пересечение компонентов, содержащее два или больше общих элементов. Это приведет к уменьшению глубины диагностирования ЦУ из-за того, что такие элементы не могут быть однозначно протестированы. Предложенный алгоритм позволяет избегать подобных ситуаций, останавливая процесс преобразования структуры при достижении условий корректности. Одним из преимуществ данного алгоритма является

возможность генерирования «компактной» структуры, при которой достигается минимальное количество терминальных элементов. В такой ситуации общее время тестирования всей структуры уменьшается из-за тестирования «ядер» в сложных компонентах методом исключения. На основании предложенного алгоритма было разработано программное обеспечение, являющееся составной частью общего программного обеспечения диагностического комплекса, который применяется в условиях крупносерийного производства цифровых устройств.

Библиографический список

1. Рудаков И.В. Декомпозиция формальной модели сложного дискретного устройства // Вестник Рязанского государственного радиотехнического университета. 2011. № 36. С.108–111.
2. Городецкий Ами. Снова о внутрисхемном тестировании // Компоненты и технологии. 2011. № 7. С. 57 – 59.
3. Батищев Д.И., Старостин Н.В. Задачи декомпозиции графов. Н.Новгород: ННГУ, 2001.
4. Скворцов Н. В., Скворцов С. В. Автоматизация проектирования процессов самодиагностики для многопроцессорных систем с активной отказоустойчивостью // Вестник Рязанского государственного радиотехнического университета. 2013. № 4 – 2 (46). С. 71-77.
5. Перевозников С.И., Озеранский В.С., Снугур А.В. Алгоритмические основы и критерии формирования компонентных структур диагностирования цифровых устройств // Вестник ВПИ. 2008. № 5. С. 56 – 61.
6. Перевозников С.И., Очуров Н.А., Озеранский В.С. Стратегии ускоренного диагностирования цифровых схем // Информационные технологии и компьютерная инженерия. 2008. № 11. С. 44 –55.

УДК 681.24+519.1

В.Н. Ручкин, В.А. Романчук, В.А. Фулин

ПРОЕКТИРОВАНИЕ НЕЙРОПРОЦЕССОРНЫХ СИСТЕМ НА ОСНОВЕ НЕЧЕТКОЙ КЛАСТЕРИЗАЦИИ

Предложена обобщенная модель проектирования нейропроцессорной системы на основе нечеткой кластеризации и анализа получаемых нечетких структур по различным техническим характеристикам. Разработана производственная модель экспертной системы, анализирующей конвейерную (К), почти-конвейерную (ПК), конвейерно-векторную (КВ), вроде конвейерно-векторную (ВКВ), матричную (М), вроде векторно-конвейерную (ВВК), векторно-конвейерную (ВК), почти векторную (ПВ), векторную (В) со своими техническими параметрами, количество модулей НМС, объем микропрограммной памяти, производительность системы и др.

Ключевые слова: нейропроцессорная система, теоретико-множественная модель, классы эквивалентности, рефлексивность, транзитивность, симметричность, кластеризация, нечеткая кластеризация, лингвистические переменные.

Введение. Архитектура современных вычислительных систем на принципах Дж. фон Неймана имеет ряд существенных недостатков [1]. Последовательный характер организации вычислительного процесса определяет принципиально низкую производительность системы в целом. Другой недостаток обуславливает низкую эффективность использования дорогостоящей памяти при наличии одного процессора-ядра. Это связано с тем, что подавляющее большинство ячеек памяти в процессе работы бездействует и поэтому коэффициент использования аппаратуры будет очень низким [4]. Согласно третьего недостатка разработчик вынужден подстраивать алгоритм решения задачи под ее «жесткую» структуру.

С другой стороны, инженерная реализация теоретических нейронов У. МакКаллока и У. Питтса в «Логическом исчислении идей, имманентных нервной активности» [2] в виде нейропроцессоров позволяет избавиться от этих недостатков. Кроме того, нейропроцессорные системы (НПС) различаются также по принципу взаимодействия структуры машины и решаемой задачи [5]. При использовании НПС разработчик подстраивает ее структуру под решаемую задачу.

С появлением кластерных систем появилась проблема кластеризации, т. е. эффективного разбиения на кластеры [8,10,14], что на практике не всегда решается оптимально или рационально.

Цель работы. Актуальным и перспективным направлением является разработка и проектирование современных нейропроцессорных структур, на базе которых может появиться новый класс интеллектуальных систем [7]. Работа проводилась в рамках гранта РФБР №14-07-00261 «Кластеризация и организация распределенных вычислительных систем на основе нейропроцессоров».

Постановка задачи. Разработать методику проектирования нейропроцессорной системы на основе кластеризации и анализа получаемых структур по различным техническим характеристикам: быстродействие, объем микропрограммной памяти, число микропроцессорных модулей.

Продукционная модель экспертной системы выбора НПС. Предлагаемая модель выбора аппаратных и программных средств НПС базируется на функциональном принципе, в соответствии с которым главную роль играет

множество операций в отношении многопроцессорных структур [3].

На **первом** этапе согласно функциональному принципу **нейропроцессорная система** (НПС) рассматривается как совокупность устройств, реализованных на основе выбранной элементной базы. Поэтому здесь осуществляется выбор элементной базы, которая характеризуется: набором схем – chipset; набором команд; производительностью и стоимостью [9].

Проектируемая вычислительная структура НПС функционирует по алгоритмам обработки и передачи информации $A^{(j)}, \forall j = \overline{1, N}$, каждый из которых представляет собой кортеж операций $A^{(j)} = \langle O_1, O_2, \dots, O_L, O_2, O_1, \dots, O_r \rangle$ длиной $|A^{(j)}|$ или алгоритмов, например: отбраковки, сжатия, калибровки, автоконтроля и других.

Структура нейропроцессорной системы рассматривается на уровне устройств. Поэтому на первом этапе определяется элементная база, в качестве которой используется система команд NM 640X. Решение такой задачи математически сводится к нахождению отображения

$$O_l \rightarrow \{MK_m\}; \forall m = \overline{1, K_m}; \forall l = \overline{1, L}, \quad (1)$$

где под множеством команд $MK = \{MK_1, MK_2, \dots, MK_m\}$ конкретного комплекта СБИС подразумевается внутренний язык вычислительной системы (ВС). MK_m – минимальное количество команд, реализующее операцию O_l .

На **втором** этапе в зависимости от решения задачи (1) проектирования каждому j -му алгоритму обработки ставится в соответствие программа обработки $PR^{(j)}$ посредством определяемого отображения:

$$A^{(j)} \rightarrow PR^{(j)}, j = \overline{1, N} \quad (2)$$

Под программой обработки информации $PR^{(j)}$ понимается кортеж команд: $PR^{(j)} = \langle MK_1, MK_2, \dots, MK_i, \dots, MK_M \rangle$. При этом характеристиками рассматриваемой программы $PR^{(j)}$ являются: длина программы $|PR^{(j)}|$, определяемая как общее число команд, входящих в программу; время выполнения программы $T = \sum_{m=1} t_m$, где t_m – время выполнения m -й команды, причем

$t_m \in T : PR^{(j)} \Rightarrow (PR^{(j)}, T)$. При этом выражение (2) определяет однопроцессорный вариант НПС, что соответствует централизованной структуре обработки информации.

Далее вводится понятие структуры $S_w \in S$, под которой понимается отношение параллельности выполнения подпрограмм RO_l, RO_k в виде кластеров CL_l и CL_k [6] двумя различными нейропроцессорами (модулями): $CL_l S_w CL_k; CL_l, CL_k \in PR$ проектируемой нейропроцессорной системы.

Тогда на **третьем** этапе проектирования определяется множество всевозможных структур $S_w \in S$, позволяющих некоторой J -й программе обработки PR поставить в соответствие множество кластеров CL_l

$$S_w \in S : PR \Rightarrow \{CL_l\}; \forall l = \overline{1, L}; \forall w = \overline{1, W}. \quad (3)$$

Выражение (3) определяет уже распределенную структуру $S_w \in S$ нейропроцессорной системы.

На **четвертом** этапе осуществляется оптимальный выбор структуры нейропроцессорной системы обработки информации по заданной стратегии проектирования, включающей в себя минимум аппаратных и программных средств и удовлетворяющей требованиям по производительности.

Таким образом, структура S_w проектируемой нейропроцессорной системы обработки информации ставит в соответствие некоторой J -й программе обработки PR совокупность независимых и неравных между собой кластеров CL_l , число которых равно числу классов эквивалентности L , а кратность (RO_l) определяется порядком класса эквивалентности (a_l):

$$\forall o = \overline{1, N}, PR^{(j)} \Rightarrow \{(RO_l)^q\}, \forall q = 1, a_l; \forall l = \overline{1, L} \quad (4)$$

Далее рассмотрим возможные следующие варианты решения задачи (4) в виде продукционной модели экспертной системы.

Продукционная модель экспертной системы выбора нейропроцессорной системы

В результате решения задачи (4) возможны следующие варианты структурной организации нейропроцессорной системы, которые составляют основу продукционной модели на базе классификации (4) для дальнейшего анализа с учетом исходных данных.

Число классов эквивалентности равно L , и порядок каждого l -го класса равен $|a_l|$. Тогда

i -му алгоритму обработки данных НКС ставится в соответствие совокупность данных, определяемых согласно (4), и, как нетрудно видеть, этот случай является наиболее общим и характеризует мультинейропроцессорную структуру, типа МКМД (по Флинну), так как предлагает отношения не только между элементами внутри класса, как в матричной структуре, но и между элементами различных классов, как в конвейерной структуре.

Рассматриваемые ниже варианты структур НПС обработки информации оказываются возможными в зависимости от построения конкретной J -й структуры данных $PR^{(j)}$, а точнее от передачи данных между этими структурами.

Пусть обмен информацией между представителями кластеров $CL_l, l = \overline{1, L}$ осуществляется последовательно друг от друга.

Если элементы структуры внутри l -го класса обмениваются последовательно, тогда назначим число НМ для случая $|a_l| = q, \forall l = \overline{1, L}$, равное $N_0 = L \cdot q$, и для общего случая $N_0 = L \sum_{l=1}^L |a_l|$. В результате имеем уже известную конвейерную структуру, аппаратные и программные средства и временные параметры для которой определяются следующими соотношениями.

Получаемая система с числом $N_0 = L \cdot q$ вычислительных средств функционирует по кластеру для случая $|a_l| = q, \forall l = \overline{1, L}$:

$$CL = q \sum_{l=1}^L CL_l \quad \text{и для общего случая} \\ CL = \sum_{l=1}^L \sum_{m=1}^M CL_{lm}. \quad \text{Время обработки данных DS}$$

находится по формулам: $T_r = q \sum_{r \in RO_l} t_i^{(1)}$,

$$T_r = \sum_{l=1}^L |a_l| \sum_{r \in RO_l} t_i^{(l)} \quad \text{для первого и второго случаев соответственно.}$$

Если для элементов внутри l -го класса требуется информация одновременно, тогда назначим число модулей НКС для случая $|a_l| = q, \forall l = \overline{1, L}$, равное $N_0 = L \cdot q$, и для общего случая $N_0 = L \cdot \max_{l \in L} |a_l|$. Получаемую при этом структуру нейрокомпьютера назовем **конвейерно-векторной** структурой НКС.

Если при обработке информация требуется

одновременно для всех пользователей, тогда число N_0 процессорных модулей назначается равным $N_0 = L \cdot q \quad \forall l = \overline{1, L}; \quad |a_l| = q$ и $N_0 = \sum_{l=1}^L |a_l|$ для общего случая. В результате получаем **матричную** структуру нейрокомпьютера обработки информационных данных.

В самом общем случае при обработке поступающей информации исходные данные могут требоваться одновременно для некоторых НС и последовательно передаваться между рядом оставшихся. Другими словами, ряд задач распараллеливается, как в векторной структуре, а ряд элементов - как в конвейере. В результате получается сложная конвейерно-векторная или векторно-конвейерная структура вычислительной системы, число нейропроцессоров в которой определяется по уже известным соотношениям. Однако следует отметить два аспекта. Во-первых, под конвейерной, конвейерно-векторной, векторной, векторно-конвейерной или в общем виде матричной структурами понимается множество возможных конфигураций, каждая из которых также должна быть классифицирована. Например, с использованием лингвистических переменных на базе нечетких множеств [11]. Во-вторых, сложные мультинейропроцессорные системы можно реализовать на кристалле на базе многоядерных процессоров. Полученные соотношения являются базовыми для обучения экспертной системы.

Многокритериальный нечеткий выбор системы НПС

На практике процесс проектирования нейрокомпьютерной системы является сложной многокритериальной задачей выбора структуры из полученного многообразия $S_w \in S$. Лицо, принимающее решение на основании обобщенной оценки каждой альтернативы, выбирает наилучшую по совокупности противоречивых критериев: быстродействие ($1/T_R$), объем микропрограммной памяти ($|MP|$), число микропроцессорных модулей (N_0), время простоя (T_{nn}) и др. Дополнительно можно рассмотреть и другие критерии: надежность (P), гибкость перестройки — адаптация (A), готовность промышленности (K), стоимость (Q), патентная чистота (Π) и т.д. Для конкретизации выбора все вышеперечисленные критерии предлагается ранжировать в порядке важности технического проектирования и в дальнейшем называть **стратегией проектирования**.

Предлагаемая в работе методика многокритериального выбора базируется на основе нечетких множеств. Характеристическая функция в этой методике имеет область определения отрезок $E = [a, b]$ и элементами являются лингвистические переменные. Введение нечеткости позволяет моделировать плавное изменение свойств мультимикропроцессорных реализаций нейрокомпьютера с учетом качественных структурных связей, что особенно важно для нейроподобных сетей. При этом решение задачи выбора происходит одновременно с процедурой принятия решения [12].

ЛПР оценивает структуры и критерии с помощью понятий нечеткой переменной или термов $t \in T$. Причем каждая переменная имеет область определения в виде отрезка $E = [a, b]$ и функцию принадлежности $m_t(e)$, т.е. $t = \{m_t(e), e\}$.

В предлагаемой методике используются два вида нечетких переменных: **бинарная** лингвистическая переменная — «отношение значимости критерия» $R_{j|j+1}$ или «отношение полезности структуры» $Q_{j|j+1}$; и **унарная** лингвистическая переменная — «качественная оценка значимости критерия» $m_{z_j}(x)$ или «качественная оценка полезности структуры» $m_{z_j, si}(x, y)$.

Значения бинарных лингвистических переменных «отношение значимости критерия» и «отношение полезности структуры» строятся методом нечеткого алгоритма высказывания правил вида: «если A , то B ». Два и более правила нечетких высказываний объединяются связкой «иначе»: «если A , то B , иначе C » = $[A][B] + [A]*[C]$ или V («если A , то B , иначе C ») = $m_A / V(B), 1 - m_A / V(C)$.

Оценка значимости каждого из предыдущих в списке критериев производится согласно стратегии проектирования по правилу композиции $Z_j = Z_{j+1} \circ R_{j|j+1}, j = n-1, n-2, \dots, 1$. При этом функция принадлежности нечеткого множества Z_j является результатом макс-минной операции:

$$m_{z_j}(x) = \bigcup_{y \in Y} m_{z_{j+1}}(y) \bigcap_{x \in X} m_{R_{j|j+1}}(y, x).$$

Для получения весов значимости критериев нечеткие множества преобразуются в скаляры одним из известных способов, например выбором такого значения бинарной переменной, при котором функция принадлежности достигает

максимума. Затем скалярные оценки значимости критериев шкалируются и нормализуются. В результате каждый критерий получает вес $W_i, \forall i = \overline{1, n}; \sum_{l=1}^n W_l = 1$.

Аналогичным образом из лингвистических оценок отношения полезности получается для каждой альтернативы S_w нормализованная оценка полезности по каждому J -му критерию F_{ij} .

Интегральной характеристикой предложенного многокритериального нечеткого выбора является аддитивная функция полезности U_i , которая вычисляется в заключительной части предлагаемой методики для альтернативных вариантов структур S_w обработки информации по совокупности всех критериев. Кроме того, ранжируются варианты по значению $U_i = \sum_{j=1}^n W_j * F_{ij}$.

По полученным данным и определяется предпочтительный вариант структуры $S_w \in S$ НПС с максимальной функцией полезности.

В результате предлагается **алгоритм логического вывода**, включающий в себя следующие шаги.

1. Выбор носителей и получение начального распределения.

2. Определение стратегии проектирования, которое выражается в ранжировании заданных критериев.

3. Определение значения бинарных лингвистических переменных «отношение значимости критерия» $R_{j|j+1}$, которое производится по правилу композиции $Z_j = Z_{j+1} \circ R_{j|j+1}$, с выбором критериев, начиная с последнего $j = n - 1, n - 2, \dots, 1$.

4. Получение «качественной оценки значимости критерия» по формуле $m_{z_j}(x) = \bigcup_{y \in Y} m_{z_{j+1}}(y) \bigcap_{x \in X} m_{R_{j|j+1}}(y, x)$.

5. Получение скалярных оценок X_j^k , т. е. определение номера тех носителей, при которых значения критериев достигают единицы.

6. Шкалирование оценок по формуле $H_i = 3(X_j^k + 1) + 1$.

7. Получение нормализованных весов по

формуле: $W_j = \frac{H_j}{\sum_{j=1}^n H_j}$.

8. Определение значений бинарных лингвистических переменных «отношение полезности структуры» по формуле: $Z_j = Z_{j+1} \circ R_{j|j+1}$, с выбором структуры, начиная с первой.

9. Определение значения «качественной оценки полезности структур» по формуле $m_{z_j}(x) = \bigcup_{y \in Y} m_{z_{j+1}}(y) \bigcap_{x \in X} m_{R_{j|j+1}}(y, x)$.

10. Получение скалярных оценок Y_{ij}^k , т. е. определение номера тех носителей, при которых значения структур достигают единицы.

11. Шкалирование оценок по формуле $H_{ij} = 3(Y_{ij}^k + 1) + 1$.

12. Получение нормализованных весов $F_{ij} = \frac{H_{ij}}{\sum_{j=1}^n H_{ij}}$.

13. Повторение вычислений, начиная с п.8, для всего множества критериев.

14. Вычисление аддитивной функции полезности по формуле: $U_i = \sum_{j=1}^n W_j * F_{ij}$ альтер-

нативных вариантов структур нейрокомпьютерных реализаций проектируемого нейрокомпьютера по совокупности всех критериев согласно стратегии проектирования.

15. Вывод результата логического решения задачи проектирования нейрокомпьютера.

Экспертная система выбора НПС. Как видно из вышеприведенного материала, нечеткий выбор структуры нейрокомпьютерной системы представляет собой сложный трудоемкий процесс с большим объемом исходной информации. Поэтому для сокращения сроков проектирования таких систем была разработана экспертная система (ЭС) [6, 13], т. е. основные операции проектирования осуществляются экспертом. Исходными данными базы знаний являются характеристики алгоритмов: число классов эквивалентности L , порядок каждого класса $|a_l|$, обмен данными между фрагментами классов, поступление входных, выдача выходных данных, множество микрокоманд выбранного нейропроцессорного набора и упорядоченное множество критериев в виде стратегии проектирования Z . Причем обмен данными между фрагментами классов эквивалентности может быть последова-

тельным, параллельным, последовательно-параллельным или параллельно-последовательным.

Основой базы знаний была выбрана продукционная модель представления знаний, в которой связи между фактами представлены эвристическими правилами — выражениями декларативного знания об отношениях между объектами. Каждое такое правило имеет составляющую «ЕСЛИ» (предпосылка) и компонент «ТО» (заключение), которые определяют прямую и обратную причинно-следственную связь. Именно эта информация базы знаний и обрабатывается с помощью машины вывода. В результате работа экспертной системы сводится к следующей последовательности действий.

1. Если число классов эквивалентности равно L , И порядок каждого класса $|a_l|=1$, И

- обмен данными между фрагментами в классах осуществляется последовательно, — **ТО** “конвейерная структура”;

- информация требуется одновременно всем фрагментам классов, **ТО** — “векторная структура”.

2. Если число классов эквивалентности L равно единице И порядок каждого класса $|a_l|$ равен q , И

- обмен данными между фрагментами в классах осуществляется последовательно, **ТО** — “конвейерная структура”;

- информация требуется одновременно всем фрагментам классов, **ТО** — “векторная структура”.

3. Если число классов эквивалентности L , И порядок каждого класса $|a_l|=1$, И

- обмен данными между фрагментами в классах осуществляется последовательно, **ТО** — “конвейерная структура”;

- обмен данными между фрагментами в классах осуществляется последовательно с небольшим исключением фрагментов l -го класса, **ТО** — “почти конвейерная структура”;

- обмен данными между фрагментами в классах, кроме l -го, осуществляется последовательно, а l -му классу информация требуется одновременно, **ТО** — “конвейерно-векторная структура”;

- фрагменты обмениваются информацией последовательно, однако в некоторых классах информация требуется одновременно, **ТО** — “вроде конвейерно-векторная структура”.

4. Если порядок каждого класса равен числу классов эквивалентности L , **ТО** —

“матричная структура”;

- информация сразу требуется фрагментам всех классов, но в некоторых классах требуется последовательный обмен, **ТО** — “вроде векторно-конвейерная структура”;

- информация требуется одновременно фрагментам всех классов, но фрагменты l -го класса обмениваются последовательно, **ТО** — “векторно-конвейерная структура”;

- информация требуется одновременно фрагментам всех классов, кроме некоторых фрагментов l -го класса, **ТО** — “почти векторная структура”;

- информация требуется одновременно всем фрагментам классов, **ТО** — “векторная структура”.

Выходными данными предложенной методики многокритериального нечеткого выбора структуры нейрокомпьютерной системы являются полученные в результате классификации структуры нейрокомпьютерной системы в виде нечетких переменных термов: конвейерная (К), почти-конвейерная (ПК), конвейерно-векторная (КВ), вроде конвейерно-векторная (ВКВ), матричная (М), вроде векторно-конвейерная (ВВК), векторно-конвейерная (ВК), почти векторная (ПВ), векторная (В) функции принадлежности по каждой структуре и все технические параметры нейрокомпьютерной системы: количество модулей НМС, объем микропрограммной памяти, производительность системы в виде времени реализации алгоритмов обработки информации, время простоя и др.

Вышеприведенные данные и заданные стратегии проектирования Z являются входными данными для работы алгоритма логического вывода. Вся полученная информация заносится в таблицу, в которую также включены разделы качественной оценки значимости критериев $Z_j = Z_{j+1} \circ R_{j|j+1}$, скалярные оценки X_j , шкалированные оценки $H_{ij} = 3(Y_{ij}^k + 1) + 1$ и, наконец, нормализованные веса W_j каждого критерия

$$W_j = \frac{H_j}{\sum_{j=1}^n H_j}, j = 1, n$$

Далее строятся лингвистические бинарные отношения полезности вариантов по каждому критерию и производятся аналогичные вычисления скалярных Y_{ij} , шкалированных оценок H_{ij} , формализованных весов F_{ij} , которые заносятся также в таблицу.

Качественные оценки значимости критериев

№	Структура	Качественная оценка значимости критериев	Скалярные оценки X_{ij}	Шкалированные оценки W_{ij}	Нормализованные веса F_{ij}
	Критерий				
1	1/Tr	0 0 0 0.50.50.70.7 1 0.5	0.9	6.7	0.3
2	MP	0.50.7 1 0.50.50.30.1 0 0 0	0.2	4.6	0.2
3	N_0	0 0 0 0.10.40.50.7 1 0.50.4	0.8	6.4	0.28
1	S_1	1 0.50.50.3 0 0 0 0 0 0	0	4	0.24
	S_2	0.20.50.70.7 1 1 0.5 0 0 0	0.5	5.5	0.32

	S_m	0 0 0 0 0 0.20.7 1 1	1	7	0.44
2	S_1	1 1 0.50.50.5 0 0 0 0 0 0	0.1	4.3	0.26
	S_2	0.20.40.50.7 1 1 0.50.5 0 0	0.6	5.8	0.35

	S_m	0 0 0 0 0 0.20.4 1 1 0.5	0.8	6.4	0.39
3	S_1	0 0 0 0.10.40.50.7 1 0.40.4	0.7	6.1	0.35
	S_2	0 0 0 0.10.4 1 0.40.1 0 0	0.5	5.5	0.325

	S_m	0 0 0 0.10.4 1 0.40.1 0 0	0.5	5.5	0.325

На основании полученных данных вычисляются аддитивная функция полезности U_i альтернативных вариантов структур нейромикропроцессорной системы реализации нейрокомпьютера по совокупности всех заданных критериев:

$$U_1 = \sum_{j=1}^n W_j * F_{1j} = 0.3; U_2 = \sum_{j=1}^n W_j * F_{2j} = 0.32;$$

$$U_3 = \sum_{j=1}^n W_j * F_{3j} = 0.38.$$

Заключение. В статье предложена обобщенная модель проектирования нейромикропроцессорной системы на основе нечеткой кластеризации. Разработана экспертная система анализа получаемых нечетких структур по различным техническим характеристикам согласно стратегии проектирования: быстроедействие ($1/T_R$), объем микропрограммной памяти ($|MP|$), число микропроцессорных модулей (N_0). Предложенная модель не зависит от конкретных алгоритмов и не определяет класс задач. Тогда как получение конкретной структуры НПС будет зависеть от алгоритмов, т.е. класса задач. Результаты экспериментального исследования, приведенные в таблице, указывают на предпочтительный вариант векторно-конвейерной реализации структуры с максимальной функцией полезности. Что указывает на общность полученных результатов распараллеливания как по командам, так и по данным.

Библиографический список

1. Архитектура компьютерных систем и сетей: учеб. пособие, Т.П. Барановская, В.И. Лойко,

М.И. Семенов, А.И. Трубилин; под ред. В.И. Лойко. – М.: Финансы и статистика, 2003. –256 с.: ил.

2. Корячко В. П., Скворцов С. В., Телков Т. А. Архитектуры многопроцессорных систем и параллельные вычисления. – М.: Высш. школа, 1999.

3. Корячко В.П., Скворцов С.В., Таганов А.И., Шибанов А.П. Эволюция автоматизированного проектирования электронно-вычислительных средств // Радиотехника. 2012. № 3. С. 97–103.

4. Корячко В.П., Скворцов С.В., Телков И.А. Модель планирования параллельных процессов в суперскалярных процессорах // Информационные технологии. 1997. № 1. С. 8-12.

5. Злобин В.К., Ручкин В.Н., Нейросети и нейрокомпьютеры, СПб.: БХВ-Петербург, 2011. 256 с.

6. Григоренко Д.В., Ручкин В.Н. Повышение восстанавливаемости кластерных нейропроцессорных систем обработки данных // Цифровая обработка информации № 1, 2013.

7. Виксне П.Е., Фомин Д.В., Черников В.М. Однокристалльный цифровой нейропроцессор с переменной разрядностью операндов //Известия ВУЗов. Приборостроение. 1996. №. 7. С. 13-21.

8. Романчук В.А., Ручкин В.Н. Алгоритмы анализа вычислительных структур на базе нейропроцессоров (статья ВАК) // Вестник Рязанского государственного радиотехнического университета. 2012. № 40. С. 60 – 66.

9. V. Ruchkin, V. Romanchuk, R. Sulitsa, “Clustering, Restorability and Designing Of Embedded Computer Systems Based On Neuroprocessors “, 2nd Mediterranean Conference on Embedded Computing MECO - 2013 Budva, Montenegro.

10. Акименко Т.А., Аршакян А.А., Ларкин Е.В. Управление информационными процессами в робототехнических комплексах //Тула, Издательство ТулГУ, 2012.

11. Карпов В. С., Ивутин А. Н., Суслин А. А. Подход к реализации методики оценки надежности ПО на основе комплексных метрик //Известия Тульского государственного университета. Технические науки. 2009. №. 4.

12. Костров Б.В., Ручкин В.Н., Тхьюнг Н.К. Нейромикропроцессорная обработка информации //Научное издание. 2008. Т. 9. №. 1.

13. Романчук В.А., Ручкин В.Н. Разработка программных средств анализа нейропроцессорных систем (статья ВАК) // Вестник Рязанского государственного радиотехнического университета. 2010. № 32. С. 61 – 67.

14. Колмыков М.В., Романчук В.А., Ручкин В.Н. Возможности программного комплекса NM Model для разработки и отладки программ обработки изображений (статья ВАК) // Вестник Рязанского государственного радиотехнического университета. 2008. № 24. С. 83 – 85.