

УДК 681.3(075.8)

В.В. СКЛЯР¹, В.А. ГОЛОВИР²

¹Государственный научно-технический центр по ядерной и радиационной безопасности, Украина

²ЗАО «Радий», Украина

МЕТРИКИ ОЦЕНКИ СЛОЖНОСТИ ПРОЕКТОВ ПЛИС, РЕАЛИЗУЮЩИХ АЛГОРИТМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ОБОРУДОВАНИЕМ АЭС

Предложены метрики оценки сложности проектов ПЛИС, основанные на учете количества элементов графической схемы проекта. Проведен сравнительный анализ новых метрик с метриками Холстеда, применяемыми для оценки сложности программного обеспечения. Разработана методика оценки сложности алгоритмов функционирования информационно-управляющих систем, представленных на проблемно-ориентированных языках.

ПЛИС, метрики сложности, технологические алгоритмы АЭС

Постановка задачи и обзор публикаций

Цифровые устройства на базе программируемых логических интегральных схем (ПЛИС) широко применяются для выполнения основных функций в информационно-управляющих системах (ИУС) критического применения. При этом проекты ПЛИС, подобно программному обеспечению (ПО), создаются в инструментальной среде и могут содержать дефекты, внесенные в процессе разработки. В то же время, при подобном подходе к построению ИУС надежность ПЛИС решающим образом влияет на надежность и безопасность ИУС в целом [1].

Проекты ПЛИС могут включать следующие составляющие:

– графическая схема цифрового устройства, разработанная с использованием библиотеки схематехнических компонентов; стандартная библиотека САПР может быть дополнена модулями IP-Cores (Intellectual Property Cores – ядра с интеллектуальными свойствами) которые являются, с одной стороны, универсальными и надежно повторяемыми, а с другой стороны, имеют возможность параметрической настройки под конкретный проект [2, 3];

– описание проекта цифрового устройства, разработанное на языке описания аппаратуры, например VHDL, AHDL, Verilog [4, 5].

– ПО эмулятора процессора, которое имплементируется в структуру ПЛИС (примером является эмулятор процессора Nios фирмы Altera).

Одним из основных характеристик проектов ПЛИС, влияющих на их надежность, является сложность. Для ПО разработан ряд подходов к оценке сложности, например, с использованием метрик Холстеда [6] и метрик Мак-Кейба [7], объектно-ориентированных метрик [8]. Ряд метрик ПО рекомендован к использованию стандартами ISO/IEC 9126 «Программная инженерия – Качество продукта» и IEEE 982 «Стандартный словарь IEEE по измерениям при производстве надежного программного обеспечения» [9, 10].

Однако, в известной литературе для проектов ПЛИС, для которых выдвигаются требования по надежности и безопасности, и которые разрабатываются в виде графической схемы цифрового устройства, отсутствуют описания метрик сложности.

Целью статьи является разработка метрического подхода к оценке сложности проектов ПЛИС.

1. Разработка метрик оценки сложности гарантоспособных проектов ПЛИС

Разработку метрик сложности выполним на примере проекта ПЛИС, реализующего алгоритмы формирования сигналов аварийной защиты (АЗ) ядерного реактора ВВЭР-1000. Такой тип реактора

является базовым для энергоблоков АЭС Украины, а также для ряда АЭС России, Болгарии, Чехии, Китая и др. При помощи указанных алгоритмов реализуется непосредственное выполнение функции безопасности АЭС – управление АЗ реактора. От сложности алгоритмов АЗ зависит потенциальная возможность внесения в них дефектов при разработке. Таким образом, сложность управляющих алгоритмов непосредственно влияет на безопасность технических комплексов критического использования. Указанный проект ПЛИС является ядром управляющего контроллера системы аварийной и предупредительной защиты реактора, разработанной ЗАО «Радий» (г. Кировоград) для оснащения энергоблоков АЭС [11 – 13].

Пример одного из алгоритмов АЗ ядерного реактора ВВЭР-1000 приведен на рис. 1. Для различных энергоблоков АЭС с реактором ВВЭР-1000 реализован различный состав алгоритмов АЗ. В рассматриваемом проекте ПЛИС реализовано в общей сложности 21 алгоритм АЗ.

Анализ алгоритмов АЗ показал, что они включают следующие пять классов компонентов:

1) входные сигналы, представленные входными линиями связи;

2) выходные сигналы, представленные выходными линиями связи;

3) промежуточные линии связи между элементами схемы; при наличии узлов линии связи от точки узла до входа в элемент учитываются как отдельные структурные единицы, в том числе, на входе и выходе схемы алгоритма;

4) типовые логические элементы И, ИЛИ, НЕТ;

5) типовые функциональные блоки, реализуемые в САПР путем разработки IP-Cores, включаемых в состав библиотеки проекта ПЛИС.

Рассмотрим подход к метрической оценке сложности проектов ПЛИС, реализующих алгоритмы управления технологическим оборудованием реактора АЭС. Общая формула для

метрики сложности проекта ПЛИС может быть представлена в виде:

$$CM = \sum_{i=1}^S w_i \cdot A_i, \quad (1)$$

где S – количество классов компонентов, входящих в состав графической схемы проекта ПЛИС, для рассматриваемого случая $S = 5$; A_i – количество компонентов графической схемы проекта ПЛИС, входящих в состав i -го класса; w_i – весовой коэффициент, учитывающий влияние компонентов i -го класса графической схемы проекта ПЛИС на интегральную сложность проекта.

В общем случае при равном влиянии компонентов на сложность проекта ПЛИС весовые коэффициенты могут быть определены как $w_i = 1 / S$. Однако, в рассматриваемом случае непосредственные разработчики проектов ПЛИС указали на разное влияние классов компонентов на общую сложность проектов ПЛИС. Компоновка связей между элементами схемы представляет собой более сложный процесс, чем непосредственное внесение элементов в состав схемы. Кроме того, входные и выходные связи (сигналы) отдельных схем алгоритмов необходимо интегрировать в составе проекта ПЛИС, из-за чего влияние входных и выходных на сложность проекта выше, чем влияние внутрисхемных линий связи. Учитывая вышеизложенное, для весовых коэффициентов были приняты следующие значения ($\sum w_i = 1$):

– для входных и выходных сигналов – $w_1 = w_2 = 0,25$;

– для линий связи – $w_3 = 0,2$;

– для типовых логических элементов и типовых функциональных блоков – $w_4 = w_5 = 0,15$.

Для схемы алгоритма аварийной защиты реактора, представленной на рис. 1, имеем следующие значения для количества компонентов проекта ПЛИС A_i :

– для входных сигналов – $A_1 = 3$ (на рис. 1 имеются три сигнала, соответствующие трем

Защита №1

Алгоритм формирования сигнала АЗ

Давление над активной зоной менее 14,71 МПа (150 кгс/см²) и скорость снижения давления над активной зоной более 0,0459 Мпа/сек (0,5 (кгс/см²)/сек)
 $T_d = 0,1 \text{ с}$ $Зв = 0,5\%$

Поправки: для 1-го и 2-го каналов $P_{аз} = - 0,246 \text{ МПа}$ (- 2,6 кгс/см²); для 3-го канала $P_{аз} = - 0,196 \text{ МПа}$ (- 2,0 кгс/см²)

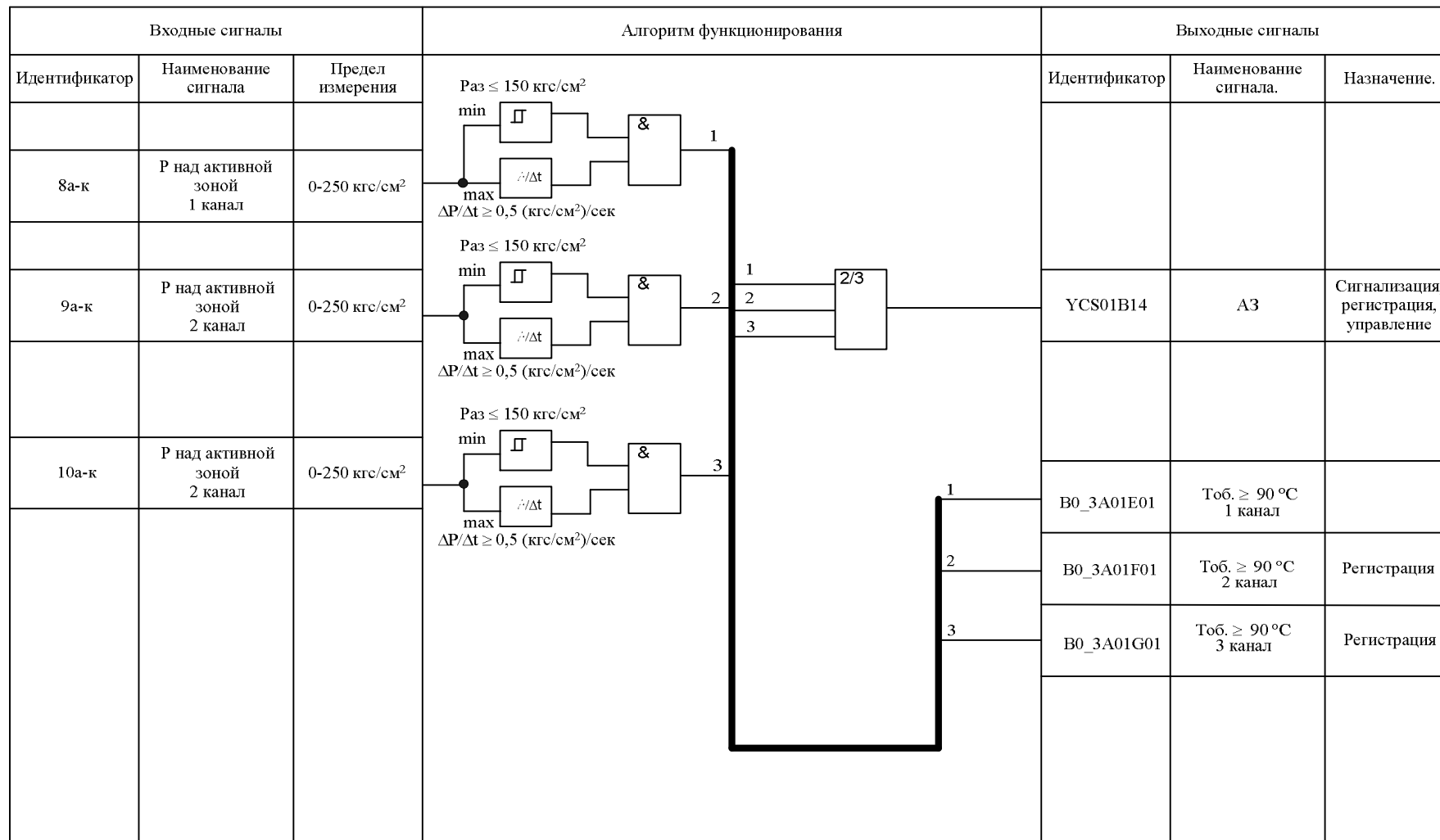


Рис. 1. Пример алгоритма формирования сигналов аварийной защиты ядерного реактора ВВЭР-1000, реализуемого проектом ПЛИС

каналам измерения значения давления над активной зоной реактора);

– для выходных сигналов $A_2 = 4$ (на рис. 1 имеются один выходной управляющий сигнал аварийной защиты и три выходных регистрационных сигнала);

– для линий связи – $A_3 = 12$ (на рис. 1 имеются разветвления на входе, поэтому кроме трех входных сигналов также учитываются три линии связи – от узлов до входа в элементы дифференциаторов $\Delta P/\Delta t$; аналогично учитываются три линии связи между выходами элементов И и входами мажоритарного элемента; кроме того, на рис. 1 имеется еще шесть внутренних линий связи);

– для типовых логических элементов – $A_4 = 3$ (на рис. 1 в состав схемы алгоритма входит три элемента И);

– для типовых функциональных блоков $A_5 = 7$ (на рис. 1 в состав схемы алгоритма входит три компаратора на минимум, срабатывающих при снижении давления до значения 150 кгс/см^2 , три дифференциатора $\Delta P/\Delta t$, реализующих реальное дифференцирующее звено для измерения скорости изменения давления, а также один мажоритарный элемент «2 из 3»).

Таким образом, для составляющей проекта ПЛИС, представленной на рис. 1, имеем следующее значение метрики сложности: $CM = 0,25 \cdot 3 + 0,25 \cdot 4 + 0,2 \cdot 12 + 0,15 \cdot 3 + 0,15 \cdot 7 = 5,65$.

2. Адаптация метрик Холстеда для оценки сложности проектов ПЛИС

Еще одним способом оценки сложности проекта ПЛИС является применение метрик Холстеда [6, 11].

Исходными данными для расчета метрик Холстеда являются:

– число различающихся простых операторов (словарь операторов) $n1$;

– число различающихся простых операндов (словарь операндов) $n2$;

– общее число всех операторов $N1$;

– общее число всех операндов $N2$.

Основными метриками Холстеда являются:

1) словарь: $n = n1 + n2$;

2) длина программы: $N = N1 + N2$;

3) объем программы: $V = N \log_2 n$;

4) сложность программы: $D = (n1/2)(N2/n2)$;

5) усилия на разработку программы: $E = VD$;

6) количество ошибок в программе:

$$B = V/3000 = E^{2/3}/3000.$$

Очевидно, что для проекта ПЛИС аналогами операторов являются логические элементы и функциональные блоки, а аналогами операндов – линии связи, включая входные и выходные сигналы, а также внутрисхемные связи. Для составляющей проекта ПЛИС, представленной на рис. 1, имеем $N1 = A_4 + A_5 = 3 + 7 = 10$; $N2 = A_1 + A_2 + A_3 = 3 + 4 + 12 = 19$. При учете различающихся простых операторов определяется количество видов логических элементов и функциональных блоков. Поскольку на рис. 1 имеются элементы И, компараторы на минимум, дифференциаторы и мажоритар «2 из 3», $n1 = 4$. При учете различающихся простых операндов определяется количество видов сигналов без учета разветвлений линий связи. Для схемы на рис. 1 $n2 = 13$.

3. Анализ результатов оценки сложности гарантоспособного проекта ПЛИС

В табл. 1 представлены результаты оценки рассматриваемого проекта ПЛИС, реализующего 21 алгоритм АЗ ядерного реактора ВВЭР-1000. В таблице 1 приведены все параметры для расчета метрики сложности по формуле (1), а также для расчета сложности D и объема V по Холстеду.

Метрики сложности проектов ПЛИС могут применяться для оценки критического объема алгоритма и проекта ПЛИС, после превышения которого резко возрастает вероятность внесения ошибок; в случае превышении объема рекомендуется провести структурирование проекта ПЛИС на модули.

Таблиця 1

Результаты оценки сложности проекта ПЛИС, реализующего алгоритмы АЗ ядерного реактора ВВЭР-1000

Номер АЗ	A_1	A_2	A_3	A_4	A_5	CM	$N1$	$N2$	$n1$	$n2$	D	V
АЗ 1	3	4	12	7	3	5,65	10	19	4	13	3,80	118,54
АЗ 2	16	26	28	24	3	20,15	27	70	5	33	6,48	509,05
АЗ 3	3	4	3	4	0	2,95	4	10	2	7	2,50	44,38
АЗ 4.1	16	17	20	16	5	15,40	21	53	4	37	5,05	396,46
АЗ 4.2	16	17	24	20	5	16,80	25	57	5	41	5,70	452,93
АЗ 5	16	17	32	28	5	19,60	33	65	5	49	4,92	563,98
АЗ 6	9	7	15	9	1	8,50	10	31	7	22	10,85	199,18
АЗ 7	6	9	9	3	7	7,05	10	24	4	16	4,80	146,95
АЗ 8	9	4	12	4	9	7,60	13	25	2	22	1,92	174,23
АЗ 9,13	6	8	6	2	6	5,90	8	20	2	14	2,50	112,00
АЗ 11	3	4	3	1	3	2,95	4	10	2	7	2,50	44,38
АЗ 14	3	4	3	1	3	2,95	4	10	2	7	2,50	44,38
АЗ 15	3	4	3	1	3	2,95	4	10	2	7	2,50	44,38
АЗ 19	6	4	9	3	7	5,80	10	19	4	16	3,80	125,34
АЗ 20	3	4	13	7	3	5,85	10	20	4	13	4,00	122,62
АЗ 21	3	4	3	4	0	2,95	4	10	2	7	2,50	44,38
АЗ 22	3	4	3	4	0	2,95	4	10	2	7	2,50	44,38
АЗ 23	3	4	4	2	4	3,45	6	11	3	8	2,75	58,81
АЗ 24	3	4	4	2	4	3,45	6	11	3	8	2,75	58,81
АЗ 25	24	37	84	68	5	43,00	73	145	7	40	6,95	1210,90
АЗ 26	3	4	3	4	0	2,95	4	10	2	7	2,50	44,38

Согласно метрикам Холстеда критическим объемом является $V = 3000$. Результаты оценки данного показателя для схем алгоритмов АЗ показали, что для данного проекта ПЛИС критический объем превышен не был. Для разработанной метрики сложности рекомендуется установить значение критического уровня $CM = 50$. Для анализируемого проекта ПЛИС данный уровень сложности также не был превышен.

Кроме того, с использованием метрик сложности может быть проведена оценка трудозатрат на разработку и модификацию проектов ПЛИС. Для этого должна быть установлена норма на реализацию условных единиц сложности проекта ПЛИС в единицу времени. Для анализируемого проекта ПЛИС, включающего 21 модуль, необходимо просуммировать затраты на разработку схем каждого из модулей и учесть затраты на интеграцию проекта.

Выводы и перспективы дальнейших исследований

Проведенное исследование в перспективе позволяет распространить полученные результаты на оценку сложности любых алгоритмов функционирования гарантоспособных ИУС, представленных на проблемно-ориентированных языках. Такие алгоритмы могут быть реализованы, как путем разработки проектов ПЛИС, так и путем разработки программного обеспечения для микроконтроллеров.

Методика оценки сложности алгоритмов, представленных на проблемно-ориентированных языках включает следующие действия:

- анализ проблемно-ориентированного языка, на котором разработаны алгоритмы, и выделение классов операторов и операндов;
- задание весов с точки зрения сложности для классов операторов и операндов;

– установление для интегральной метрики сложности граничного значения, в случае превышения которого резко возрастает вероятность внесения ошибок в программном обеспечении или в проект ПЛИС;

– нормирование трудозатрат путем задания количества единиц сложности, которые разработчик условно может реализовать за единицу времени;

– непосредственное измерение сложности, включающее подсчет количества операторов и операндов для каждого из классов и определение значения интегральной метрики сложности;

– анализ полученных значений метрики сложности, формулирование выводов и рекомендаций.

Метрика сложности, предложенная в данной статье для оценки проектов ПЛИС, реализующих алгоритмы управления технологическим оборудованием АЭС, позволяет более точно, по сравнению с известными метриками, учесть специфику САПР и процесса разработки.

Литература

1. Laprie J.-C. Dependability Handbook. LAAS Report n 98-346. – Toulouse: Laboratory for Dependability Engineering, 1998. – 365 p.
2. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. – М.: Горячая линия – Телеком, Радио и Связь, 2001. – 636 с.
3. Харченко В.С., Тарасенко В.В., Ушаков А.А. Отказоустойчивые встроенные цифровые системы на программируемых логических интегральных схемах (ПЛИС). – Х.: Нац. аэрокосм. ун-т «ХАИ», 2004. – 210 с.
4. Угрюмов Е.В. Цифровая схемотехника. – С.-Пб.: БХВ–Петербург, 2001. – 528 с.
5. Стешенко В.Б. ПЛИС фирмы ALTERA: Проектирование устройств обработки сигналов. – М.: ДОДЭКА, 2000. – 128 с.
6. Холстед М.Х. Начала науки о программах. – М.: Финансы и статистика, 1981. – 128 с.
7. McCabe T. A Complexity Measure // IEEE Transactions on Software Engineering. – 1976. – 4, n SE-2. – P. 308-320.
8. Chidamber S., Kemerer C. A Metrics Suite for Object Oriented Design // IEEE Transactions on Software Engineering. – 1994. – 21, n 6. – P. 476-493.
9. Харченко В.С., Скляр В.В., Тарасюк О.М. Методы моделирования и оценки качества и надежности программного обеспечения. – Х.: Нац. аэрокосм. ун-т «ХАИ», 2004. – 159 с.
10. IAEA-TECDOC-1328. Solutions for cost effective assessment of software based instrumentation and control systems in nuclear power plants. Report of the Technical Group on Nuclear Power Plant Control and Implementation. – Vienna: IEIA, 2002. – 131 p.
11. Использование метрик Холстеда при оценке безопасности критического программного обеспечения / В.С. Харченко, В.В. Скляр, А.А. Гордеев, В.И. Токарев, А.Д. Герасименко, Ю.А. Белый // Радиоелектронні і комп'ютерні системи. – 2003. – Вип. 4. – С. 145-150.
12. Головир В.А., Скляр В.В., Харченко В.С. Методы внесения и оценки версионной избыточности при разработке информационно-управляющих систем на базе ПЛИС // Вісник Хмельницького національного університету. – 2005. – № 4. – Ч. 1, Т. 1. – С. 94-97.
13. Ushakov A., Kharchenko V., Golovir V. Self-Repairing FPGA-Systems Using Multi-Parametrical Adaptaion to Cluster Faults // Proceedings of IEEE East-West Design & Test Workshop (EWDTW'05). – Odessa (Ukraine). – 15-19 September 2005. – P. 225-230.

Поступила в редакцию 24.01.2006

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.