



ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА: ОПРЕДЕЛЕНИЕ ОБЪЕМА ТЕСТОВЫХ ИСПЫТАНИЙ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ

Д.А. Панков, Л.А. Денисова (ОмГТУ)

Рассмотрен вопрос определения объема тестовых испытаний для программно-аппаратных устройств информационного комплекса. Обоснован критерий окончания тестирования и определены основные компоненты программы имитации неисправностей для испытаний микропроцессорного устройства. Предлагается методика оценки выполнения программных функций устройства в процессе тестирования на основе совместного использования техники фаззинга и применения математического аппарата нечеткой логики. Приводятся результаты выявления дефектов программного обеспечения при имитации сбоев и отказов устройства, подтверждающие эффективность предлагаемого подхода.

Ключевые слова: программно-аппаратный комплекс, имитация отказов и сбоев, тестирование программного обеспечения, фаззинг.

Введение

Одной из проблем при проектировании информационных комплексов, осуществляемом на приборостроительном предприятии, является обеспечение достаточного объема тестовых испытаний, выявляющих отказы и сбои микропроцессорных устройств [1]. Обычно тестирование программно-аппаратных комплексов производится путем искусственного внесения ошибок в программное и аппаратное обеспечение проверяемых устройств¹. Отметим, что при недостаточном объеме тестовых испытаний вероятность проявления не выявленных потенциально опасных для системы дефектов значительно повышается, поэтому важно определить критерий окончания тестирования.

Существуют различные критерии завершения испытаний, обеспечивающие выполнение требований к надежности функционирования системы [1, 2]. Алгоритмы тестирования программно-аппаратных устройств часто используют такой показатель надежности, как наработка на отказ, определяя время проведения испытаний, но не учитывая при этом особенности реализуемого на устройстве программного обеспечения. Кроме того, обычно алгоритмы тестирования не принимают во внимание характер изменения работы программного обеспечения, вызванного изменением входных данных, а также технические особенности проявляющихся отказов и сбоев.

Тестирование программно-аппаратных устройств с помощью имитации неисправностей регламентируется стандартами ИЕС 61508 и ГОСТ Р МЭК 61508 и является обязательным требованием при разработке устройств информационных комплексов [2].

Формирование критерия окончания испытаний, позволяющих выявить дефекты сложного микропроцессорного устройства, необходимо осуществлять на основе экспертной оценки его основных функций, определяемых техническим заданием. Обычно программа испытаний, формируемая заказчиком проектируемого устройства исходя из набора выполняемых функций, не гарантирует получение тестового сценария, применимого на практике. Набор испытаний может быть избыточен или недостаточен, так как не учитывает особенности программно-аппаратных средств, реализующих эти функции. Устранение в дальнейшем не выявленных дефектов влечет дополнительные затраты временных и технических ресурсов.

В работе при рассмотрении вопроса тестирования программно-аппаратных устройств основное внимание уделено формированию критерия завершения испытаний, учитывающего не только функции системы, но и особенности программного обеспечения, а также ключевые параметры аппаратных средств, характеризующие стабильность работы. Предлагается методика оценки выполнения основных функций устройства, реализуемая программой испытаний на основе совместного использования техники фаззинга (для изменения входных данных) и математического аппарата нечеткой логики (для определения момента останова тестирования).

Программа испытаний микропроцессорного устройства

Рассмотрим вопрос организации тестовых испытаний, необходимых для контроля качества и приемки устройств информационного комплекса. Для

¹ Пат. 2549523 РФ, МПК G 06 N 11/00 (2006.01). Способ мутационного тестирования радиоэлектронной аппаратуры и ее управляющего программного обеспечения / Недорезов Д.А. № 2014117399/08; заявл. 29.04.14; опубл. 27.04.15. Бюл. № 12.

осуществления испытаний создается специальный программно-аппаратный комплекс, а также формируется программа тестовых испытаний проверяемых устройств.

В проверяемом информационном комплексе тестированию подлежат устройства, которые предназначены для обмена пользовательской информацией по сети, а также для ее обработки, хранения и последующего анализа. Поэтому для тестирования предлагается применить программную имитацию неисправностей, реализованную с использованием техники фаззинга, позволяющей автоматизировать процесс обнаружения ошибок с помощью генерации искаженных данных на устройствах во время приема. Кроме того, предлагается выполнить анализ последующей реакции программного обеспечения устройства, используя нечеткий логический вывод для определения момента завершения проверки выполняемых функций.

Поскольку микропроцессорные устройства функционируют, используя сложное программное обеспечение, то предлагается формировать критерий окончания тестирования на основе анализа статистических данных о работе аппаратных и программных средств этих устройств. Программа испытаний информационного комплекса использует статистические показатели для анализа и принятия решения об окончании или продолжении испытаний. Сбор статистических данных занимает длительное время и выполняется в автоматическом режиме средствами тестирования и диагностики. Начальные данные формируются по экспертным шаблонам. Это означает, что в качестве начальных данных принимаются наиболее часто применяемые при реальной эксплуатации текстовые наборы, передаваемые устройству. В результате работы программы формируются статистические показатели, характеризующие объем тестовых испытаний, соответствующий требуемой точности оценки среднего времени наработки на отказ (установленной экспертами).

Для проведения анализа и определения момента окончания испытаний предлагается использовать следующие четыре показателя. В качестве первого показателя принимается параметр, характеризующий точность оценки среднего времени наработки на отказ β (определяется при выполнении программных функций, в %):

$$\beta = P(|\tilde{P} - p| < \varepsilon_\beta), \quad (1)$$

где \tilde{P} — частота отказов и сбоев устройства; p — вероятность отказов и сбоев устройства; ε_β — допустимая погрешность оценки среднего времени наработки на отказ; p — реальное значение вероятности отказов и сбоев, определяемое при эксплуатации устройства; P — доверительная вероятность, которая определяется как отклонение между реальным значением вероятности возникновения отказов и сбоев и его оценкой по частоте.

Вторым принятым показателем является число аппаратных отказов K при исполнении программных функции за период проведения испытаний. В качестве третьего принимается показатель I_β , характеризующий вероятность выполнения устройством некорректных действий для данного алгоритма работы (соответствующих отказам и сбоям устройства) во время реализации программных функций. Показатель I_β рассчитывается по формуле [3]:

$$I_\beta \in ([\tilde{P} - t_\beta \sqrt{\tilde{P}(1-\tilde{P})K^{-1}}; [\tilde{P} + t_\beta \sqrt{\tilde{P}(1-\tilde{P})K^{-1}}]), \quad (2)$$

где K — число аппаратных отказов устройства; t_β — число среднеквадратичных отклонений от центра рассеивания среднего времени наработки на отказ; \tilde{P} — частота отказов и сбоев устройства.

В качестве четвертого показателя D принимается коэффициент корреляции между временем внесения ошибок во входные данные и временем наступления отказов и сбоев в процессе проведения испытаний [4]:

$$D = \frac{\sum_{i=1}^N (t_{d_i} - \bar{t}_{d_i})(t_{err_i} - \bar{t}_{err_i})}{\sqrt{\sum_{i=1}^N [(t_{d_i} - \bar{t}_{d_i})^2 (t_{err_i} - \bar{t}_{err_i})^2]}}, \quad (3)$$

где N — число программных функций; t_{d_i} (\bar{t}_{d_i}) — время внесения (среднее время внесения) i -й ошибки с момента начала тестирования, мс; t_{err_i} (\bar{t}_{err_i}) — время проявления (среднее время проявления) i -й ошибки с момента начала тестирования, мс. Таким образом, показатель D позволяет определить, есть ли связь между внесением ошибки и ее проявлением или ошибка случайно произошла не зависимо от внесения неисправности.

Показатель K фиксируется аппаратными средствами контроля, а показатель β устанавливается перед началом испытаний экспертами, которые учитывают требования к устойчивости к отказам и сбоям.

Показатели I_β и D вычисляются средствами диагностики в соответствии с выражениями (2) и (3). При этом средства диагностики фиксируют показатели с установленным периодом (10 мин.).

Для определения объема тестовых испытаний устройств информационного комплекса разработана специальная программа тестирования, в которой предусмотрено определение момента завершения испытаний. При этом устанавливается, что объем проведенных испытаний достаточен на основе введенного для этой цели критерия окончания испытаний. После расчетов и фиксации средствами диагностики показатели обрабатываются, и вычисляется значение критерия окончания испытаний. Алгоритм вычисления на основе решающих правил нечеткого логического вывода по четырем показателям определяет критерий окончания испытаний. Полученное на выходе данного алгоритма решение является численным значением критерия, по которому определяется полнота проведения испытания или необходимость их продолжить.

Критерий окончания испытаний H формируется при работе специализированных программных и аппаратных средств тестовой инфраструктуры испытательного комплекса в соответствии с выражением:

$$H = \sum_{j=1}^N \sum_{i=1}^M Rule_{ij}(I_{\beta}, D, K, \beta) \cdot M^{-1}, \quad (4)$$

где N — число программных функций, M — число производственных правил для вычисления критерия окончания испытаний заданной функции; $Rule_{ij}(I_{\beta}, D, K, \beta)$ — результат вычисления критерия окончания испытаний в соответствии с i -м производственным правилом нечеткого вывода (вычисляется на основе показателей) для j -й программной функции. Объем испытаний считается достаточным в тот момент, когда получено первое значение критерия $H > 1$.

Алгоритм выполнения программы испытаний представлен на рис. 1. Для имитации эксплуатации устройства в реальных условиях осуществляется внесение ошибок модулем фаззинга [5], искажающим входные данные устройства при работе в штатных режимах. Искажения входных данных позволяют выявить ошибки при выполнении арифметических и логических команд, и, кроме того, вносимые искажения последовательности выполнения программы приводят к ошибкам в командах перехода, которые, в свою очередь, ведут к отказам и сбоям. Сформированный в результате работы алгоритма критерий окончания испытаний H позволяет определить необходимость дальнейшего проведения испытаний (для выбранного значения точности оценки среднего времени наработки на отказ β).

Следует отметить, что использование техники фаззинга для тестирования программ позволяет осуществить автоматический поиск ошибок программного обеспечения устройства с помощью искажения входных данных и последующего анализа реакции устройства. Для проверки работы программного обеспечения устройства с помощью фаззинга использованы две метрики: $PATH_COV$ (число путей в программе от общего числа, принимающее значения 0...100%) и ERR_CNT (число полученных ошибок), наиболее информативных для оценки результатов применения фаззинга [6].

Технический комплекс для проведения испытаний микропроцессорного устройства

Технический комплекс, предназначенный для проведения испытаний, представляет собой программно-аппаратную систему, используемую для выявления дефектов разработанного устройства перед прохождением приемо-сдаточных испытаний² [7]. Комплекс содержит инфраструктуру тестирования, которая настраивается для проверки работоспособности устройства в требуемых режимах работы, а также выполнения штатных функций по техническому заданию для этого устройства. Микропроцессорное устройство, как правило, при функционировании осуществляет регистрацию всех выполняемых операций. Для разных типов разрабатываемых устройств состав и структура комплекса проведения испытаний отличаются, поскольку отличаются схемотехнические решения, интерфейсы и программное обеспечение устройств. Поэтому при разработке устройства обычно проектируется и комплекс проведения испытаний, предназначенный для проверки функций данного устройства.

Рассмотрим разработанный технический комплекс проведения тестовых испытаний для устройства приема и передачи информации на базе процессоров общего назначения с поддержкой интерфейсов граничного сканирования *JTAG*. Инфраструктура тестирования реализует фиксацию реакции системы на имитацию ошибок. По реакции системы устанавливается, вызваны ли отказ или сбой системы отсутствием защиты от искаженных данных в программе или вводимая ошибка должна приводить к такому отказу (сбою).

Компоненты экспериментального комплекса реализуют следующие функции.



Рис 1. Алгоритм выполнения программы испытаний устройства

² Пат. 2697629 РФ, МПК G06F 11/261 (2018.08). Устройство для имитации неисправностей в программно-аппаратных системах / Панков Д.А. № 2018105476 заявл. 13.02.18; опублик.15.08.19. Бюл. № 23.

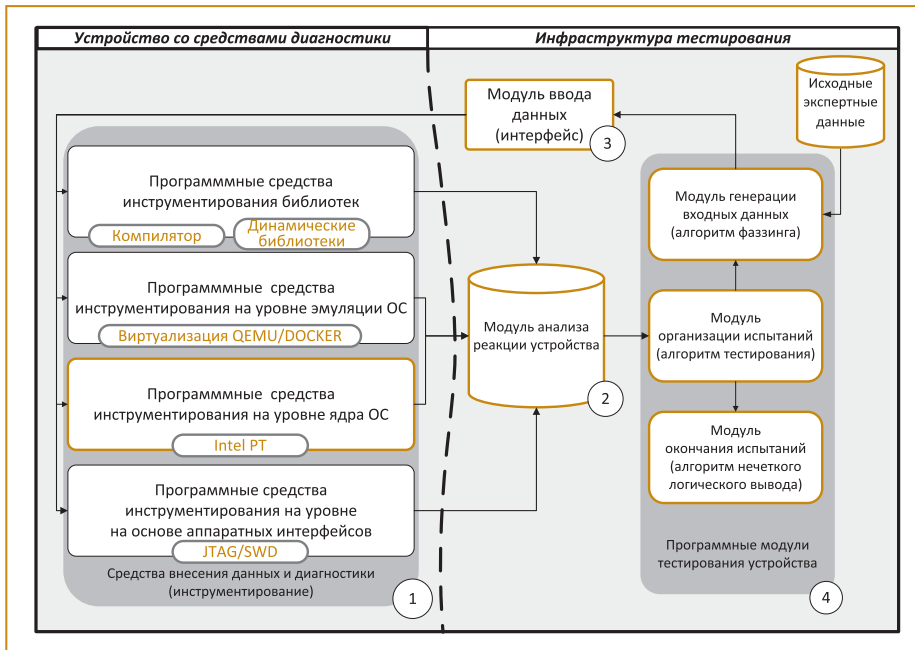


Рис. 2. Функциональная схема комплекса проведения испытаний

Настройка программных средств комплекса для проведения испытаний (инфраструктуры тестирования) для контроля показателей работы программы при проведении фазинга.

Получение и обработка данных о результатах выполнения функций программного обеспечения устройства для тестового набора данных.

Анализ выполнения программы с помощью эвристических алгоритмов модели организации испытаний (с использованием техники фазинга).

Формирование модифицированных данных путем искажения битов данных для достижения наибольшего числа переходов в программе.

Верификация инфраструктуры тестирования с использованием тестов системы с известным результатом (сравнение реакции устройства на внесенные данные с требуемой реакцией). Имеется в виду, что если внесли ошибку, которая выявляет сбой при классическом тестировании, то инфраструктура тестирования позволит увидеть, что ошибка произошла. Если ошибки нет, комплекс настроен некорректно.

На рис. 2 представлена функциональная схема экспериментального комплекса проведения испытаний. Слева показаны функции контроля ошибок при исполнении программы устройства, справа изображены программные модули инфраструктуры тестирования. Функции контроля реализуются в программном обеспечении устройства на разном уровне абстракций (аппаратные драйверы, операционная система, системные утилиты эмуляции, внедрение в пользовательскую программу). Выбор абстракции для контроля исполнения программы устройства производится исходя из экспертной оценки сложности программного обеспечения, которая формируется в результате классического тестирования.

Опишем блоки функциональной схемы устройства. Программные средства внесения данных и диагностики включают четыре блока:

– первый блок (инструментирование на уровне интерфейсов) использует аппаратный интерфейс *JTAG/SWD* для получения и внесения данных в программное обеспечение;

– второй блок (инструментирование на уровне ядра) реализует способ инструментирования на уровне операционной системы (системный уровень) и осуществляет доступ к аппаратным интерфейсам с помощью функции процессора, используя программные средства для процессоров *Intel PT*;

– третий блок (инструментирование на уровне эмуляции ОС) предполагает создание копии операционной системы (эмуляцию) для отслеживания выполнения программы с помощью программ виртуализации *DOCKER* и *QEMU*;

– четвертый блок (инструментирование библиотек) реализует способ использования непосредственно модификации отслеживаемой программы с помощью средств компилятора и динамических библиотек.

Рассмотрим состав и функции инфраструктуры тестирования устройства, основой которой являются программные модули тестирования, реализованные на отдельном (сервисном) компьютере, подключаемом к тестируемому устройству для внесения входных данных и анализа реакции на их внесение.

Модуль организации испытаний реализует алгоритм запуска, останова и коррекции испытаний. Для запуска испытаний он использует модуль генерации входных данных, который формирует искаженные данные для внесения в устройство. Модуль окончания испытаний обрабатывает данные от устройства и формирует критерий окончания испытаний.

Модуль ввода данных представляет собой интерфейс обмена «сервисный компьютер — устройство для ввода/вывода информации (от средств диагностики на устройстве) на сервисный компьютер» (Патент 2549523). Модуль анализа реакции устройства реализуется на сервисном компьютере и принимает данные от устройства.

Модуль генерации входных данных с помощью фазинга является важным компонентом инфраструктуры тестирования, который с использованием эвристического алгоритма генерации данных осуществляет подготовку искаженных входных данных. Подготовка данных заключается в искажении би-

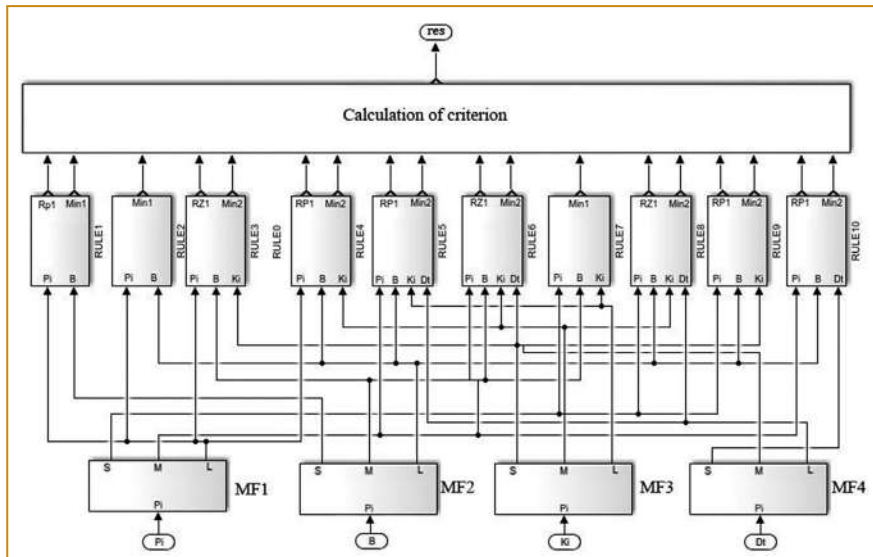


Рис. 3. Схема вычисления критерия окончания испытаний на основе нечеткого логического вывода

тов экспертных данных в псевдослучайном порядке (с нуля на единицу), применяя при этом логические операции XOR, AND и их комбинации с целью задействования большего объема выполняемой программы, что при контроле проявляется как повышение значений метрик фазинга PATH_COV и ERR_CNT (определяются экспериментально).

Анализа реакции устройства заключается в периодическом опросе через интерфейс JTAG состояния устройства и передачи данных на компьютер, на котором функционируют программные модули тестирования устройства для расчета показателей I_β , β , K , D (с использованием формул (2) – (3)). Полученные на тестируемом устройстве показатели передаются модулю окончания испытаний, где определяется численное значение критерия окончания испытаний H . Модуль окончания испытаний реализован как отдельная подпрограмма, и его работа основана на аппарате нечеткой логики [8]; модуль выполнен в среде MATLAB/Fuzzy Logic Toolbox.

Структура и функции модуля окончания испытаний

Рассмотрим реализацию модуля окончания испытаний. Данный модуль в процессе тестирования на основе показателей работы устройства формирует значение критерия окончания испытаний. Для этого полученные в результате имитации неисправностей численные значения показателей используются как входные данные для алгоритма работы нечеткого логического вывода. После получения значения критерия окончания испытаний тестовые испытания продолжаются при $H \leq 1$ и останавливаются при значении $H > 1$.

На рис. 3 представлена схема вычисления критерия окончания тестовых испытаний (в модуле определения окончания испытаний) программно-аппаратного устройства.

Для реализации алгоритма нечеткого логического вывода входные переменные показателей I_β , β , K ,

D переводятся в значения нечетких лингвистических переменных блоками фаззификации MF1 – MF4. При выполнении фаззификации число, принадлежащее множеству действительных чисел, представляется в виде нечеткого числа (от англ. fuzzy – нечеткий) [8]. Полученные нечеткие переменные используются при выполнении операций, сформулированных в виде нечетких правил, реализуемых блоками RULE1 – RULE10.

Для фаззификации входных переменных (показателей $x = (I_\beta, \beta, K, D)$) диапазоны их изменения разбиваются на лингвистические термы (на основе экспертных оценок). При этом для каждого показателя используется три терма значений: S – малое, M – среднее, L – большое.

Для каждого из термов строится функция принадлежности $\mu(x)$ переменной x этому терму.

Для задания внутренних лингвистических термов (M – среднее) входных переменных использована симметричная гауссова функция принадлежности (gaussmf), формируемая в соответствии с выражением $\mu(x) = e^{-x-c)^2 / (2\sigma^2)}$, где параметр c задает модальное значение функции, а σ – ширину. Для задания крайних термов (S – малое и L – большое) используются сигмоидальные функции принадлежности (sigmf), которые определяются в соответствии с выражением $\mu(x) = (1 + e^{-a(x-c)})^{-1}$, где $a < c$. Параметр c определяет координату точки перегиба функции, а коэффициент a – наклон функции в этой точке. При этом $a > 0$ используется для S -образной функции принадлежности, а $a < 0$ для Z -образной функции принадлежности.

Заключение каждого правила задается в виде значения критерия окончания испытаний H , которое реализовано как одноэлементное нечеткое множество с точечной функцией принадлежности. Диапазон изменения переменной H также разбивается на три терма: PL – положительное, PZ – нейтральное,

Таблица 1. Правила определения критерия окончания испытаний H

Показатель I_β	Показатель K		
	S	M	L
При показателе $\beta = S$			
S	PZ	-	-
M	-	-	-
L	-	-	PL
При показателях $\beta = M, D = M, D = L$			
S	-	-	-
M	-	PZ	PL
L	PZ	PL	-
При показателях $\beta = L, D = L$			
S	PL	PZ	PN
M	-	-	-
L	-	-	PN

P_N — отрицательное (в соответствии с принадлежностью значения критерия этим термам принимается решение об окончании испытаний). Далее производится определение четкого числа критерия H путем выполнения процедуры дефаззификации (обратного преобразования нечетких переменных в четкие). Вычисление критерия окончания испытаний H осуществляется путем определения взвешенного среднего в соответствии с выражением:

$$H = \sum_{i=1}^m \mu(H)H_i \left(\sum_{i=1}^m \mu(H) \right)^{-1}. \quad (5)$$

Здесь H_i — значение выходной переменной для i -го термина с единичным значением степени принадлежности; $\mu(H)$ — степень принадлежности к этому терму; m — число термов.

Для осуществления нечеткого логического вывода используется совокупность нечетких правил, реализующих на основе нечетких переменных (для показателей I_β , β , K , D) вычисление критерия окончания испытаний H по формуле (5). Правила нечеткого вывода $RULE1$ — $RULE10$, которые реализуются для каждой из контролируемых программных функций, формируются в соответствии с табл. 1.

Отбор программных функций для контроля показателей производился таким образом, чтобы внесение ошибок при их выполнении было наиболее значимо для работы устройства. Функции выбраны те, которые позволяли выявить наибольшее число ошибок при классическом тестировании и отладке программ на устройстве.

При тестировании рассматривались два режима работы устройства: приема $R1$ и обновления данных $R2$. Для режима приема данных выбраны три функции $N = 3$ (f_1 — функция приема, f_2 — функция установки соединения, f_3 — функция настройки). Для режима обновления данных выбраны пять функций $N = 5$ (f_1 — функция приема, f_2 — функция установки соединения, f_3 — функция настройки, f_4 — функция проверки данных, f_5 — функция обновления данных).

Поскольку выходные данные от устройства передаются в модуль окончания испытаний не равномерно по времени (в связи с задержками в интерфейсе передачи информации), то в рамках программы

испытаний предусмотрено, что вычисленные показатели передаются для формирования критерия окончания испытаний за установленный временной интервал. Это означает, что весь набор данных загружается в модуль одновременно, что позволяет вычислять значение критерия окончания испытаний на основе достоверных данных.

Результаты экспериментальных исследований

При проведении тестовых испытаний работы устройства информационного комплекса и имитации ошибок с помощью техники фаззинга определены значения контролируемых показателей устройства. Результаты расчета показателей приведены в табл. 2.

На основе анализа показателей для режимов приема данных R_1 и обновления данных R_2 сформированы интервалы изменения показателей, внутри которых обеспечиваются требования к критерию окончания испытаний (для выбранного значения точности оценки среднего времени наработки на отказ β). На основе полученных значений критерия окончания испытаний H принимается решение о необходимости прекращения испытаний. На основе экспертных оценок, а также исходя из условий эксплуатации системы и требований технического задания, принималось значение требуемого показателя $\beta=80\%$, остальные параметры рассчитывались по результатам работы средствами инфраструктуры тестирования в рамках программы испытаний. Как можно видеть, критерий окончания испытаний устройства в значительной степени зависит от наличия аппаратных отказов (значения параметра K), что учитывается при формировании решения о завершении испытаний. Кроме того, при формировании критерия окончания испытаний учитывается корректность работы программного обеспечения. Анализ значения параметра D позволяет выявлять изменения при выполнении алгоритма работы (отклонения от работы по штатному алгоритму) под влиянием вносимых в устройство искаженных данных.

Результаты тестирования для программного обеспечения устройства иллюстрирует рис. 4. Столбцы значений показателей содержат допустимые интервалы, выделенные серым цветом. Для каждой проверяемой функции принимается решение об окончании тестирования на основе величины степени принадлежности критерия к множеству значений, соответствующих множеству положительных решений. Общий критерий окончания тестирования для всего устройства определяется на основе результатов, полученных при работе модуля окончания испытаний для всех функций, и это решение представляется на дисплей. Значение критерия окончания тестирования иллюстрируется цветом: светлый цвет — достаточно

Таблица 2. Результат тестирования устройства информационного комплекса

Число функций программы, ед.	Показатели				Критерий окончания испытаний H	Загрузка процессора, %	Время, ч	Найдено ошибок, ед.
	I_β	β	D	K				
Прием данных (режим $R1$)								
$N = 3$	0,5	0,8	0,5	0,3	1,3	67,36	12	24
	0,7	0,8	0,4	0,7	1,5			
	0,6	0,8	0,4	0,8	0,8			
Обновление данных (режим $R2$)								
$N = 5$	0,4	0,8	0,17	0,1	0,5	78,27	12	18
	0,7	0,8	0,4	0,6	1,2			
	0,8	0,8	0,35	0,7	1,3			
	0,8	0,8	0,76	0,9	0,3			
	0,5	0,8	0,17	0,32	1,1			

для завершения испытаний, серый цвет — не определено, темный цвет — не достаточно для завершения испытаний. Как можно видеть, значения критерия H , которые соответствуют завершению испытаний, соответствуют пересечению допустимых интервалов для всех показателей (выделено серым цветом). Светло-серым и темным цветом соответственно показаны значения критерия H , для которых испытания должны быть продолжены.

Следует отметить, что при увеличении числа функций возрастает и число ограничений для имитации неисправностей, поэтому время на проведение испытаний значительно возрастает. Как показали результаты исследования, выбранные показатели работы устройства представляется возможным применять для тестирования и других устройств из этой же партии при серийном производстве (идентичных в части программно-аппаратных решений), используя предлагаемый критерий для окончания испытаний при серийном тестировании.

Кроме того, в результате исследований установлено, что с ростом числа выполняемых функций каждый из четырех показателей по отдельности (в связи со сложностью определения допустимых диапазонов их изменения для большого числа функций) становится недостаточно информативен для принятия решения об окончании испытаний. Поэтому возникает необходимость выбора показателей, характеризующих достаточность объема проведенных испытаний для проверки корректного выполнения функций, возможно, на основе совместной оценки принятых при исследовании показателей.

Анализ преимуществ предлагаемой методики тестирования показал, что ее использование не менее чем на 5% сокращает объем испытаний для трех функций (тестирование устройства в режиме приема данных $R1$) и не менее чем на 10% — для пяти функций (тестирование устройства в режиме обновления данных $R2$) по сравнению с классическим тестированием.

Заключение

В результате выполненных исследований разработана методика испытаний для устройств информационного программно-аппаратного комплекса, которая позволяет определить объем тестовых испытаний. Обосновано формирование критерия окончания тестирования в автоматизированном режиме и опре-

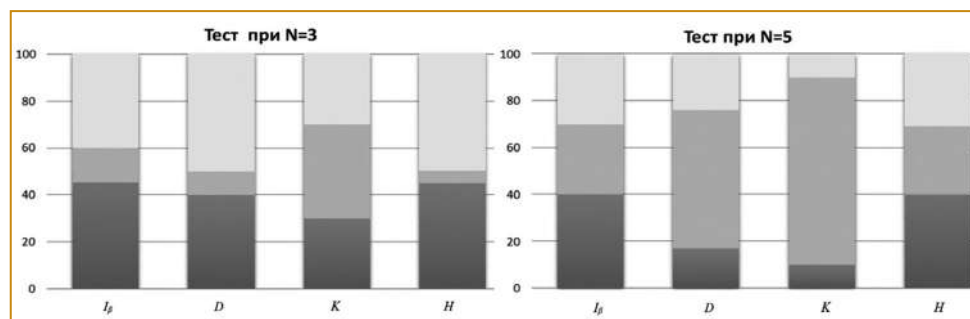


Рис. 4. Результаты тестирования программного обеспечения устройства

делены основные компоненты программы имитации неисправностей устройства.

Предлагается проверять корректность выполнения программных функций устройства в процессе тестирования на основе совместного использования техники фаззинга (для имитации сбоев и отказов устройства) и математического аппарата нечеткой логики (для принятия решения о завершении испытаний).

Полученные результаты подтвердили эффективность использования предлагаемой методики тестирования для выявления дефектов программного обеспечения устройства и аппаратных отказов. Представляется возможным применять предлагаемую методику испытаний на приборостроительном предприятии при массовом выпуске микропроцессорных устройств с использованием предварительно рассчитанных (рекомендуемых) диапазонов показателей работы, что позволит сократить объем испытаний и временные затраты на тестирование.

Список литературы

1. Увалов Д.В. Бортовая микропроцессорная система управления с повышенной сбоеустойчивостью // Решетневские чтения. 2014. С. 248 -250.
2. Pankov D.A., Denisova L.A. Model studies of systems with diagnostics based on fault simulation // IOP Conf. Ser.: Mater. Sci. Eng. 2020, 734 (1), 012021.
3. Вентцель Е.С. Теория вероятностей. М.: изд. «Наука», 1969. 576 с.
4. Черняк М.Ю., Эльберг М.С. Надежность технических систем. Красноярск. Изд. СибГУ им. М.Ф. Решетнева. 2017. С. 13-17.
5. Созин М.Ф. Фаззинг на основании состояния исполнения программы// Решетневские чтения. 2018. № 6. С 344-345.
6. Wang Y, Jia P, Liub L et al. A systematic review of fuzzing based on machine learning techniques. 2019. <https://arxiv.org/ftp/arxiv/papers/1908/1908.01262.pdf>.
7. Панков Д.А., Денисова Л.А. Контроль и диагностика неисправностей программно-аппаратного комплекса // Омский научный вестник. 2018. № 2. С. 128 -130.
8. Zadeh L.A. Fuzzy sets // Information and Control, 1965, vol. 8. Pp. 338-353.

Панков Денис Анатольевич — аспирант,
Денисова Людмила Альбертовна — д-р техн. наук, профессор Омского государственного
 технического университета (ОмГТУ).
 Контактный телефон: (381-2) 65-20-84.
 E-mail: pankovDDD@yandex.ru, denisova@asoiiu.com