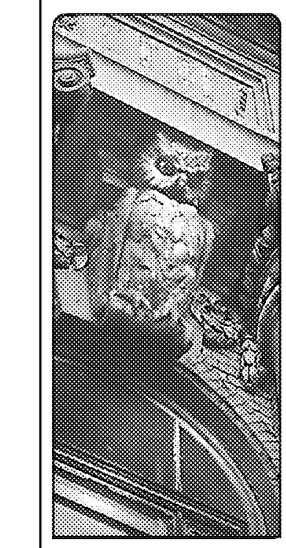

НАУЧНЫЙ ОТДЕЛ

ИНФОРМАТИКА

УДК 681.518

ХЕШ-ФУНКЦИИ ДЛЯ СОКРАЩЕНИЯ ДИАГНОСТИЧЕСКОЙ ИНФОРМАЦИИ

В.Б. Гольдштейн, С.В. Миронов

Саратовский государственный университет,
кафедра математической кибернетики и компьютерных наук
E-mail: Goldshtein@mail.ru, MironovSV@info.sgu.ru

Исследуется задача сокращения диагностической информации, используемой при локализации неисправностей дискретных устройств (ДУ). Предлагается решать эту задачу за счет подбора хеш-функции, возвращающей компактные свертки для полных реакций ДУ и реакций его неисправных модификаций на диагностический тест. Приводятся статистические данные, подтверждающие эффективность предложенного подхода.

Hash Functions for Diagnostic Information Reduction

V. B. Goldshteyn, S. V. Mironov

In this paper we present a new approach for the solution of problem of the diagnostic information reduction. This approach is based on use of hash functions delivering a compact signature for records in a fault dictionary. The experimental results show a considerable decrease in the storage requirement of diagnostic information reduced with the help of such functions.

ВВЕДЕНИЕ

Можно выделить два главных подхода к процессу диагностирования: диагностирование с использованием предварительно подготовленной диагностической информации (ДИ) и динамическое диагностирование [1,2]. При первом подходе в составе ДИ сохраняется реакция на диагностическую последовательность (тест) исправного ДУ (эталонная реакция) и реакции всех рассматриваемых неисправных модификаций этого ДУ (словарь неисправностей). Процесс диагностирования с использованием ДИ предполагает подачу на вход исследуемого ДУ диагностической последовательности, получения выходной последовательности и ее сравнение с реакциями в ДИ. Реакция в ДИ, равная полученной выходной последовательности, свидетельствует об исправности ДУ или о наличии в нем соответствующей неисправности. При динамическом диагностировании анализируется реакция исследуемого ДУ с целью локализации неисправности.

Существуют два основных препятствия к диагностированию с использованием ДИ: для современных ДУ объем ДИ, как правило, чрезвычайно велик; для генерации ДИ требуются значительные временные затраты. Однако генерация ДИ для конкретного ДУ проводится лишь один раз, тогда как время непосредственного диагностирования значительно сокращается по сравнению со временем, необходимым для динамического диагностирования [3].

К сокращению объема ДИ предложено множество подходов [3–9]. Большинство из них предполагает значительное сокращение словаря неисправностей за счет того, что вместо полной реакции на диагностические тесты в словаре сохраняется некоторая компактная свертка этой реакции. В некоторых случаях [8–9] компактная свертка заменяет и реакцию эталонного устройства.

Сокращение объема словаря неисправностей влечет за собой изменение процесса диагностирования. Для локализации неисправности

производится поиск в сокращенном словаре неисправностей не полной реакции исследуемого ДУ, а ее компактной свертки. Это означает, что генерация компактной свертки реакции исследуемого ДУ должна быть вписана в процесс диагностирования. Следовательно, время на диагностирование одного ДУ увеличивается за счет времени, необходимого для генерации компактной свертки реакции этого ДУ на тест.

Существенное сокращение объема словаря неисправностей может привести к потере «полезной» информации [3, 9], а следовательно, и к уменьшению глубины диагностирования.

Таким образом, при сокращении ДИ возникают следующие проблемы: так как алгоритм получения компактной свертки реакции ДУ выполняется в процессе диагностирования, время на его выполнение должно быть сведено к минимуму; глубина диагностирования после сокращения ДИ должна оставаться неизменной.

Далее для решения изложенных проблем предлагается использовать механизм хеш-функций.

1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Пусть $B = \{0, 1\}$ — бинарный алфавит.

Пусть рассматриваемое ДУ f_0 имеет m выходных полюсов и множество $F = \{f_i \mid 1 \leqslant i \leqslant N\}$ есть множество его неисправных модификаций. Предполагаем, что на каждом выходном полюсе упомянутых ДУ может появиться только сигнал из B.

Пусть $T = t_0, t_2, \ldots, t_{k-1}$ — диагностическая последовательность для рассматриваемого устройства, где t_j, $0 \leqslant j \leqslant k-1$, — входные вектора. Тогда количество бит $n = mk$ в полной реакции ДУ назовем объемом полной реакции. Саму полную реакцию устройства f_i на тест T можно представить в виде вектора $R_i \in B^n$, в котором на s-м месте, $0 \leqslant s \leqslant n-1$, стоит значение сигнала на выходном полюсе с номером $s - m\lfloor s/m \rfloor$ после подачи входного вектора с номером $\lfloor s/m \rfloor$.

Представим ДИ для ДУ f_0 с множеством неисправностей F в виде матрицы R порядка $(N+1) \times n$, где строка с номером i, $0 \leqslant i \leqslant N$, есть вектор R_i. Не теряя общности будем предполагать, что все строки матрицы попарно различны, т. е. нет ни одной пары устройств f_i и f_j, неразличимых с помощью представленной ДИ. В противном случае каждый класс неразличимых устройств можно заменить одним представителем.

Объемом ДИ назовем количество элементов в соответствующей матрице R, т. е. величину $(N+1)mk$.

Обозначим $H(r)$, $r \leqslant n$, множество функций вида $B^n \to B^r$.

Результатом применения функции $h \in H(r)$ к ДИ R назовем матрицу R^h, в которой строка R_i^h, $0 \leqslant i \leqslant N$, есть результат применения функции h к строке R_i исходной матрицы R

$$R_i^h = h(R_i).$$

Строку R_i^h можно воспринимать как компактную свертку полной реакции i-й модификации ДУ, а матрицу R^h — как результат сокращения ДИ.

Из всех функций из $H(r)$, подходящими для сокращения ДИ, являются только такие h, для которых выполняется

$$R_i^h \neq R_j^h \quad \text{при} \quad i \neq j, \tag{1}$$

т. е. которые сохраняют неизменной глубину диагностирования.

Из того, что R^h — бинарная матрица, очевидно, что условие (1) может выполняться только тогда, когда выполняется условие

$$r \geqslant \lceil \log_2(N+1) \rceil. \tag{2}$$

В качестве меры сокращения ДИ R с помощью функции $h \in H(r)$ введем величину

$$E(h) = \frac{\lceil \log_2(N+1) \rceil}{r}, \tag{3}$$

которую назовем эффективностью сжатия. С учетом (2) ясно, что для функции h максимальная величина $E(h)$, соответствующая наилучшему сжатию информации при выполнении условия (1), равна 1.

2. ХЕШ-ФУНКЦИИ ДЛЯ СОКРАЩЕНИЯ ДИ

В литературе описано множество различных видов хеш-функций и методов их построения [10, 11]. Ряд экспериментов показал целесообразность применения для сокращения ДИ двух разновидностей хеш-функций — полиномиальной (мультипликативной) и позиционной. Преимуществом этих хеш-функций является их простота, высокая скорость вычисления.

Полиномиальная хеш-функция $h \in H(r)$ в качестве результата выдает последние r бит результата вычисления многочлена $X_{n-1} + X_{n-2}P + X_{n-3}P^2 + \cdots + X_0P^{n-1}$ т. е.

$$h(X) = (X_{n-1} + X_{n-2}P + X_{n-3}P^2 + \cdots + X_0P^{n-1}) \mod 2^r,$$

где $X = X_0, X_1, \ldots, X_{n-1}$ — входной битовый набор (вектор полной реакции ДУ), P — параметр хеш-функции, — целое нечетное число из диапазона $[1, 2^r)$.

Для вычисления результата позиционной хеш-функции $h \in H(r)$ от аргумента X вычисляются значения

$$h_0(X), h_1(X), \ldots, h_n(X).$$

Величина $h_0(X)$ принимается равной нулю, а для определения величины $h_i(X)$, $1 \leqslant i \leqslant n$ вычисляется

$$k_i = (X_{i-1} + \cdots + X_0P^{i-1} + (r-1)P^i) \mod r,$$

где P — параметр хеш-функции, после чего принимается $h_i(X) = h_{i-1}(X) \oplus (2^{k_i}X_{i-1})$. Здесь $\oplus$ — побитовая операция сложения по модулю 2. В качестве результата хеш-функция $h(X)$ возвращает значение $h_n(X)$.

Как в полиномиальных, так и в позиционных хеш-функциях высокая скорость вычислений достигается за счет простоты выполняемых операций. Кроме того, в позиционных хеш-функциях исключается умножение длинного числа на параметр P (в общем случае тоже длинное число), и таким образом позиционная хеш-функция более эффективна с точки зрения скорости вычисления, чем полиномиальная.

Как видно из определений, полиномиальную или позиционную хеш-функцию из класса $H(r)$ можно считать определенной, если задан параметр P. Логично предположить, что не для любого параметра P для хеш-функции будет выполняться условие (1) и более того, что выполнение условия (1) для хеш-функции для некоторой ДИ R не гарантирует выполнение этого условия для другой ДИ.

Оценим вероятность того, что функция $h \in H(r)$ для ДУ с N неисправностями будет удовлетворять условию (1). Предположим хеш-функция h выдает равновероятные значения. Будем получать новые значения хеш-функции последовательно. Получая очередное значение считаем, что все полученные значения различны. Тогда вероятность того, что i-е (начиная с нуля) значение функции h не совпадает ни с одним из полученных ранее, равна $\frac{2^r-i}{2^r}$. Таким образом, вероятность того, что для $h \in H(r)$ выполняется (1) равна

$$\prod_{i=0}^{N} \frac{2^r - i}{2^r}. \tag{4}$$

Зная полученное значение вероятности можно подсчитать количество итераций (шагов) перебора, которое гарантирует с вероятностью 99% получение хеш-функции, обеспечивающей выполнение условия (1). Обозначим это количество M. Значения величины M для некоторых значений N и r приведены в табл. 1. Здесь M_i, $1 \leqslant i \leqslant 4$, — значения M для $r = r_i$. Как видно из табл. 1, с помощью небольшого перебора вероятнее всего получить хеш-функцию, обеспечивающую эффективность сжатия от 0.58 до 0.67.

Таблица 1

Количество итераций, необходимое для получения хеш-функции

N+1	r_1	$E(h_1)$	M_1	r_2	$E(h_2)$	M_2	r_3	$E(h_3)$	M_3	r_4	$E(h_4)$	M_4
100	12	0.5833	14	11	0.6364	52	10	0.7000	681	9	0.7778	$1.4 \cdot 10^5$
250	13	0.6154	212	12	0.6667	10776	11	0.7273	$3.5 \cdot 10^7$	10	0.8000	$1.2 \cdot 10^{15}$
300	15	0.6000	16	14	0.6429	71	13	0.6923	1174	12	0.7500	$3.4 \cdot 10^5$
500	15	0.6000	210	14	0.6429	10094	13	0.6923	$2.6 \cdot 10^7$	12	0.7500	$2.9 \cdot 10^{14}$
800	17	0.5882	51	16	0.6250	615	15	0.6667	85895	14	0.7143	$1.9 \cdot 10^9$
900	17	0.5882	100	16	0.6250	2271	15	0.6667	$1.2 \cdot 10^6$	14	0.7143	$3.8 \cdot 10^{11}$
1100	18	0.6111	45	17	0.6471	468	16	0.6875	49135	15	0.7333	$5.8 \cdot 10^8$
1500	18	0.6111	337	17	0.6471	25269	16	0.6875	$1.5 \cdot 10^8$	15	0.7333	$5.9 \cdot 10^{15}$

3. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Для оценки эффективности сокращения ДИ с помощью хеш-функций были проведены две серии экспериментов. В процессе экспериментов подбирались полиномиальные и позиционные хеш-функции для каждого варианта ДИ. Для каждого из значений r, начиная с наименьшего, делалось несколько попыток подбора соответствующего параметра P. При обнаружении хеш-функции, удовлетворяющей условию (1), процесс поиска прекращался.

В первой серии экспериментов производился поиск хеш-функций для исходных данных, сгенерированных случайным образом. Генерация ДИ проводилась в предположении, что полная реакция любой неисправной модификации ДУ отличается от эталонной реакции не более чем на 5%. Такая модель была выбрана в связи с тем, что одиночные неисправности в ДУ обычно приводят к незначительным изменениям реакций ДУ. Объем полной реакции варьировался от 38 до 5028 бит, а количество неисправных модификаций было взято равным 600.

Таблица 2

Получение хеш-функции для случайных данных

Доля вариантов ДИ, %	r	$E(h)$	Вероятность нахождения h, %	M
5	15	0.6667	0.40148	1200
61.6	16	0.6250	6.39021	70
33.4	17	0.5882	25.33211	16

В табл. 2 приведены результаты этой серии экспериментов. В первой колонке табл. 2 указана доля вариантов ДИ, для которых была найдена хеш-функция h с эффективностью сжатия $E(h)$, приведенной в третьей колонке. Четвертая и пятая колонки показывают соответственно значение вероятности (4) нахождения такой хеш-функции и необходимое количество M итераций для получения такой хеш-функции с вероятностью 99%. Из табл. 2 видно, что хеширование позволяет добиться сокращения исходной информации до размеров, всего лишь на несколько бит превосходящих оптимальный.

Таблица 3

Зависимость эффективности сжатия ДИ от объема полной реакции

Объем полной реакции, бит	Среднее значение $E(h)$
38–1028	0.6154
1038–2028	0.6135
2038–3028	0.6158
3038–4028	0.6150
4038–5028	0.6109

В табл. 3 приводятся результаты тех же экспериментов, но с несколько другой точки зрения. Табл. 3 показывает, что средняя эффективность сжатия ДИ с помощью найденной хеш-функций не зависит от длины полной реакции ДУ.

Первая серия экспериментов показала, что поиск полиномиальной хеш-функции для одних и тех же параметров r и P происходит быстрее, чем поиск позиционной хеш-функции. Так, в поставленных экспериментах полиномиальная хеш-функция осуществляет наиболее эффективное сокращение ДИ в 83.33% случаев, тогда как позиционная — только в 16.67%.

Для следующей серии экспериментов была использована ДИ, полученная для ДУ из каталога ISCAS'89 при моделировании одиночных неисправностей с помощью тестовых последовательностей HITEC [12]. Результаты экспериментов приведены в табл. 4.

Из табл. 4 видно, что эффективность сжатия ДИ для реальных ДУ с использованием хеш-функций не уступает результатам, полученным для случайных данных. Объем сокращенной с помощью хеш-функции ДИ в большинстве случаев составляет менее процента от исходного объема.

Стоит отметить, что дальнейшее увеличение длины диагностического теста приводит к довольно малому росту числа обнаруженных с его помощью неисправностей. По этой причине увеличение длины теста, следствием которого является возрастание объема ДИ, по-видимому, может только улучшить показатели в предпоследнем столбце табл. 4, но не ухудшить их.

Время поиска компактной свертки с помощью уже найденной хеш-функции для одной реакции на тест любого ДУ, упомянутого в табл. 4, составило менее 1 мс.

Все проведенные эксперименты подтверждают тот факт, что поиск хеш-функции с эффективностью сжатия от 0.58 до 0.67 наиболее вероятно завершается успехом за небольшое число итераций перебора.

Таблица 4

Сокращение ДИ для ДУ из каталога ISCAS'89 с помощью хеш-функций

ДУ	N	Объем полной реакции, бит	Объем ДИ, бит	r	$E(h)$	Объем сокращенной ДИ, бит	Доля сокращенной ДИ от полной ДИ, %	Время подбора h, мс
S298	177	1932	343896	13	0.6154	2314	0.673	235
S344	240	1397	336677	14	0.5714	3374	1.002	266
S349	243	1474	359656	14	0.5714	3416	0.950	266
S382	190	12444	2376804	13	0.6154	2483	0.091	3453
S386	274	2002	550550	15	0.6000	4125	0.749	406
S400	194	13284	2590380	14	0.5714	2730	0.105	2000
S444	191	13440	2580480	13	0.6154	2496	0.097	6359
S526	138	13548	1883172	13	0.6154	1807	0.095	750
S641	345	5016	1735536	15	0.6000	5190	0.299	1390
S713	343	3979	1368776	15	0.6000	5160	0.377	1281
S820	712	21185	15104905	17	0.5882	12121	0.080	14469
S832	719	21603	15554160	17	0.5882	12240	0.079	2344
S1423	293	750	220500	15	0.6000	4410	2	219
S1488	1359	22230	30232800	19	0.5789	25840	0.085	20234
S1494	1360	23655	32194455	19	0.5789	25859	0.080	31641
S1488	55	100	5600	10	0.6000	560	10	16

Для проведения сравнительной оценки сокращения ДИ с помощью хеш-функций был реализован жадный метод поиска маски ДИ, описанный в [13] и примененный, в частности, в [7].

Этот метод находит маску, применяя которую к вектору полной реакции ДУ можно получить компактную свертку для этой реакции. Количество r бит в компактной свертке обусловлено так называемым объемом найденной маски. Результат применения этого метода к тем же исходным данным, что использовались во втором наборе экспериментов приведены в табл. 5.

Таблица 5

Сокращение ДИ для ДУ из каталога ISCAS'89 с помощью масок

ДУ	N	Объем полной реакции, бит	Объем ДИ, бит	r	$E(h)$	Объем сокращенной ДИ, бит	Доля сокращенной ДИ от полной ДИ, %	Время поиска маски, мс
S298	177	1932	343896	62	0.0363	11036	3.209	984
S344	240	1397	336677	58	0.0411	13978	4.152	3688
S349	243	1474	359656	61	0.0391	14884	4.138	1218
S382	190	12444	2376804	55	0.0415	10505	0.442	11421
S386	274	2002	550550	90	0.0271	24750	4.496	2453
S400	194	13284	2590380	57	0.0402	11115	0.429	13594
S444	191	13440	2580480	60	0.0381	11520	0.446	13328
S526	138	13548	1883172	39	0.0550	5421	0.288	6328
S641	345	5016	1735536	128	0.0198	44288	2.552	11968
S713	343	3979	1368776	131	0.0194	45064	3.292	9219
S1423	293	750	220500	93	0.0265	27342	12.400	906
S2081	55	100	5600	17	0.1029	952	17	31

Сравнивая табл. 4 и 5 можно утверждать, что сокращение ДИ с помощью хеш-функций оказалось эффективнее сокращения с помощью масок, полученных реализованным алгоритмом. Прежде всего следует отметить, что объем информации, сокращенной с помощью найденных хеш-функций, в среднем в пять раз меньше объема информации, сокращенного с помощью масок. Кроме того, время, затраченное на получение маски, в среднем в шесть раз превышает время поиска хеш-функции.

ЗАКЛЮЧЕНИЕ

Приведенные в предыдущем разделе результаты решения задач минимизации диагностической информации с помощью предложенных в работе хеш-функций показывают, что такой подход оказался весьма эффективным. Достоинством этого подхода является прежде всего малое время поиска хеш-функции для реальных объектов и устройств и незначительное время на получение компактной свертки отдельной реакции ДУ. Кроме того, найденные хеш-функции позволяют весьма существенно сократить объем информации, подлежащей хранению в памяти ЭВМ. Так, по совокупности параметров это сокращение достигает в среднем 99%, что следует признать хорошим результатом.

Работа выполнена при финансовой поддержке РФФИ (*проект* 05-08-18082)

Библиографический список

1. *Abramovici M., Breuer M.A., Friedman A.D.* Digital Systems Testing and Testable Design. N.Y.: Computer Science Press, Inc., 1996.
2. *Ryan P.G., Rawat S., Fuchs W.K.* Two-stage fault location // Proc. of International Test Conf. 1991. P. 963–968.
3. *Boppana V., Hartanto I., Fuchs W.K.* Full fault dictionary storage based on labeled tree encoding // Proc. of 14th VLSI Test Symposium. 1996. P. 174-179.
4. *Pomeranz I., Reddy S.M.* On the generation of small dictionaries for fault location // Proc. of the 1992 IEEE/ACM International Conf. on Computer-Aided design (ICCAD '92). 1992. P. 272–279.
5. *Ryan P.G., Fuchs W.K., Pomeranz I.* Fault dictionary compression and equivalence class computation for sequential circuits // Proc. of IEEE International Conf. on Computer-Aided Design (ICCAD'93). 1993. P. 508-511.
6. *Chess B., Larrabee T.* Creating small fault dictionaries // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1999. Vol. 18. № 3. P. 346–356.
7. *Arslan B., Orailoglu A.* Fault dictionary size reduction through test response superposition // Proc. of the 2002 IEEE International Conf. on Computer Design: VLSI in Computers (ICCD'02). 2002. P. 480–485.
8. *Сперанский Д.В.* Об одном подходе к решению задач сокращения объема диагностической информации // Автоматика и телемеханика. 1984. № 3. С. 151–160.
9. *Барашко А.С., Скобцов Ю.А., Сперанский Д.В.* Моделирование и тестирование дискретных устройств. Киев: Наук. думка, 1992.
10. *Ахо А., Хопкрофт Дж., Ульман Дж.* Структуры данных и алгоритмы. М.: Изд. дом «Вильямс», 2003.
11. *Кармен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы, построение и анализ. М.: МЦНМО, 2002.
12. *Niermann T., Patel J.* HITEC: a test generation package for sequential circuits // Proc. European Design Automation Conf. 1991. P. 214–218.
13. *Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д.* Основы логического проектирования. Кн. 1. Комбинаторные алгоритмы дискретной математики. Минск: ОИПИ НАН Беларуси, 2004.