

Ep. 6.7
696

ISSN 0136-3549

0134-3823

TALLINNA TEHNIKAÜLIKOOLI

TOIMETISED

**ТРУДЫ ТАЛЛИННСКОГО
ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА**

**TRANSACTIONS OF TALLINN
TECHNICAL UNIVERSITY**

**МАШИННОЕ ПРОЕКТИРОВАНИЕ
ЭЛЕКТРОННЫХ УСТРОЙСТВ И
СИСТЕМ**

TALLINN 1989

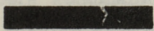
696

ALUSTATUD 1937

**TALLINNA TEHNIKAÜLIKOOLI
TOIMETISED**

**TRANSACTIONS OF TALLINN
TECHNICAL UNIVERSITY**

**ТРУДЫ ТАЛЛИННСКОГО
ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА**

 **УДК 681.5**

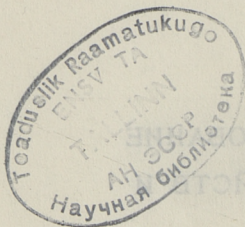
**МАШИННОЕ ПРОЕКТИРОВАНИЕ
ЭЛЕКТРОННЫХ УСТРОЙСТВ И
СИСТЕМ**

Электротехника и автоматика XXXY11

TALLINN 1989

ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Труды ТТУ № 696
МАШИННОЕ ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ
УСТРОЙСТВ И СИСТЕМ
Электротехника и автоматика ХХХУ11

В сборнике рассматриваются вопросы машинного проектирования и диагностирования электронных систем и устройств. Часть статей посвящена развитию теории декомпозиции конечных автоматов с целью автоматизации проектирования управляющих автоматов и повышения их контролепригодности. Ряд статей сборника посвящен дальнейшему решению проблем технической диагностики, предлагаются методы и алгоритмы генерирования и анализа тестов для цифровых схем, в сборник вошли также статьи, посвященные вопросам математического моделирования электронных элементов и процессов в них.



На русском языке
Отв. редактор Х. Мяги
Техн. редактор М. Тамме
Сборник утвержден коллегией Трудов ТТУ 22.06.89
Подписано к печати 06.12.89
МВ-06460
Формат 60х90/16
Печ. л. 7,25 + 0,25 приложение
Уч.-изд. л. 5,93
Тираж 300
Зак. № 645
Цена 1 руб. 20 коп.
Таллиннский технический университет,
200108 Таллинн, Эхятаате теез, 5
Ротап rint ТТУ, 200006 Таллинн,
ул. Коскла, 2/9

ИЗМЕРЕНИЕ КОЭФФИЦИЕНТОВ УДАРНОЙ ИОНИЗАЦИИ В рп-ПЕРЕХОДАХ ПРИ ОЧЕНЬ НИЗКОЙ ИНТЕНСИВНОСТИ РАЗМНОЖЕНИЯ

I. Введение

Исследованию ударной ионизации и лавинного пробоя в рп-переходах, а также методам экспериментального определения коэффициентов ударной ионизации электронов α_n и дырок α_p , посвящена обширная литература (см., например, списки литературы в [1-3]). Для измерения коэффициентов α_n и α_p в последнее время наиболее часто используется зависимость фототокса от обратного напряжения U рп-перехода, освещаемого с одной стороны или же с двух сторон светом с подходящей длиной волны. Имея экспериментальные зависимости коэффициентов лавинного размножения электронов $M_n(U)$ и дырок $M_p(U)$, можно в случае резких переходов достаточно просто определить зависимости коэффициентов ионизации от напряженности электрического поля $\alpha_n(E)$ и $\alpha_p(E)$, не делая никаких предположений о виде этих зависимостей. В случае произвольного распределения примеси требуются более сложные численные процедуры.

Однако для обеспечения достаточной точности измерения тока фотоумножения в рп-переходе необходимо провести при достаточно высоких значениях коэффициента размножения $M \geq 2$, соответствующих приведенным обратным напряжениям $U/U_B = 0,7 \dots 0,9$, где U_B есть напряжение лавинного пробоя рп-перехода. Дальнейшее увеличение напряжения U и, тем самым коэффициентов M_n и M_p нецелесообразно ввиду увеличения роли неоднородностей размножения и роста погрешности измерения в результате сближения значений коэффициентов α_n и α_p в сильных электрических полях. Необходимость проведения измерений при достаточно высоких значениях U/U_B и ис-

пользование образцов с относительно низким напряжением пробоя (обычно $U_B < 100$ В) приводит к тому, что почти все опубликованные результаты измерения α_n и α_p в различных полупроводниках относятся к достаточно высоким напряженностям электрического поля $E^{-1} < (4...5) \cdot 10^{-6}$ см·В⁻¹, в которых $\alpha_{n,p} > (10^2...10^3)$ см⁻¹.

В настоящей работе предложен метод измерения коэффициентов ударной ионизации α_n и α_p в режиме очень слабого размножения, начиная с $U/U_B \approx 0,3$ и $M \geq 1,002$. Это дает возможность надежного определения коэффициентов α_n и α_p в значительно более слабых полях с $E^{-1} = (6...11) \cdot 10^{-6}$ см·В⁻¹ и при более низких интенсивностях ударной ионизации (начиная с $\alpha_{n,p} \approx (10...100)$ см⁻¹) в структурах с типичными значениями напряжения лавинного пробоя порядка 100 В. Использование структур с более высокими U_B позволяет расширить диапазон измерений в сторону еще более низких напряженностей электрического поля.

2. Метод измерения

Предлагаемый метод измерения основывается на снятии зависимости тока базы от напряжения на коллекторном переходе биполярной транзисторной структуры, работающей в нормальном активном режиме. Для каждой величины напряжения U_{CB} ток базы I_B измеряется при двух фиксированных значениях тока эмиттера: $I_E = \text{const}$ и $I_E = 0$.

Можно показать, что в нормальном активном режиме транзисторной структуры

$$I_B = I_E (1 - M A_N) - I_{B0}, \quad (1)$$

где

$$I_{B0} \equiv I_B|_{I_E=0} = M_p I_{ps} + M_n I_{ns} + M_g I_{gr} + I_l$$

и I_{ps}, I_{ns} — дырочный и электронный токи насыщения коллекторного перехода;

I_{gr} — генерационно-рекомбинационный ток коллекторного перехода;

I_l — ток омической утечки коллекторного перехода;

M_p, M_n, M_g — коэффициенты размножения дырок и электронов, вошедших извне в коллекторный переход, а также пар электрон-дырка, генерируемых внутри коллекторного перехода.

В формуле (I) коэффициент $M = M_n$ для ррп-структуры и $M = M_p$ для рпр-структуры, а A_n обозначает коэффициент передачи постоянного тока эмиттера в нормальном активном режиме.

Входящие в (I) параметры зависят от режима работы структуры: $M(U_{CB})$, $A_n(I_E, U_{OB})$ и $I_{BO}(U_{CB})$. При этом предполагается, что коллекторный ток настолько мал, что его влиянием на распределение электрического поля в коллекторном переходе можно пренебречь и считать коэффициент M независимым от тока I_E . Принципиально важная зависимость $A_n(I_E, U_{CB})$ будет рассмотрена ниже.

Из формулы (I) непосредственно следует, что

$$1 - MA_n = \frac{I_B + I_{BO}}{I_E} = \Delta I_B^* \quad (2)$$

где ΔI_B^* — экспериментально определяемый приведенный разностный ток базы, измеряемый для различных U_{CB} при фиксированном I_E .

Обратный ток коллекторного перехода I_{BO} всегда положителен, но ток базы I_B может менять знак в зависимости от величины коллекторного напряжения U_{CB} .

Из (2) следует для коэффициента размножения

$$M(U) = \frac{1 - \Delta I_B^*(U)}{A_n(U)} \quad (3)$$

Здесь и далее опускаем индекс в обозначении обратного напряжения коллекторного перехода. Используем также вместо M приведенный коэффициент размножения

$$\Phi \equiv 1 - 1/M.$$

Тогда из (3) получаем

$$\Phi(U) = \frac{1 - A_n(U) - \Delta I_B^*(U)}{1 - \Delta I_B^*(U)} \quad (4)$$

Зависимость $I - A_n(U)$, знание которой необходимо для расчета $\Phi(U)$, обусловлена главным образом эффектом модуляции толщины базы (эффектом Эрли). На основе работ [4-6] можно записать следующую обобщенную формулу, учитывающую кроме объемных эффектов и влияние поверхности структуры на статический коэффициент передачи тока сплавных транзисторных структур с относительно толстыми однородно легированными базами:

$$1 - A_N = w^2 [c_s(I_E, w) + c_v(I_E, w) + \frac{1}{w} c_I(I_E, w)], \quad (5)$$

где w - эффективная толщина базы;

c_s, c_v - слабо зависящие от коллекторного напряжения коэффициенты, учитывающие потери носителей заряда в результате поверхностной (c_s) и объемной (c_v) рекомбинации;

c_I - коэффициент, учитывающий инжекционные потери, связанные с эмиттерным переходом.

Для образцов, использованных в настоящей работе

$$c_v > c_s \gg \frac{1}{w} c_I.$$

При этом изменения c_v и c_s не превышают 5 % при изменении напряжения коллектора в пределах от 2 до 50 В и фиксированном токе эмиттера в пределах от 0,5 до 15 мА.

Следовательно,

$$1 - A_N \approx w^2 c(I_E), \quad (6)$$

где $c(I_E)$ - некоторая экспериментально определяемая величина, зависящая, главным образом, от тока эмиттера.

Эффективная толщина базы зависит от напряжения U на коллекторном переходе:

$$w = w_0 - l(U), \quad (7)$$

где w_0 - металлургическая толщина базы,

$l(U)$ - толщина части обедненного слоя коллекторного перехода, расположенного в базе.

Из (6) и (7) получим

$$1 - A_N = c(I_E) w_0^2 \left(1 - \frac{l(U)}{w_0}\right)^2.$$

В случае резкого перехода

$$l(U) = d(\Delta\varphi_{k0} + U)^{1/2}, \quad (8)$$

где

$$d \equiv \left(\frac{2\varepsilon\varepsilon_0}{qN}\right)^{1/2};$$

$\Delta\varphi_{k0}$ - равновесная контактная разность потенциалов на коллекторном переходе;

N - концентрация легирующей примеси в базе.

Необходимые для расчета $1-A_N$ величины $w_0, N, \Delta\varphi_{к0}$ и ε относятся к сравнительно легко определяемым или задаваемым физико-геометрическим параметрам структуры, а неизвестный коэффициент $c(I_E)$ может быть определен как производная зависимости $\Delta I_B^* = f(w^2)$ в режиме низких коллекторных напряжений, когда $M \rightarrow 1$. Действительно, учитывая, что $1-A_N = \Delta I_B^*|_{M=1}$, из (6) получим

$$c(I_E) = \left. \frac{d(\Delta I_B^*)}{dw^2} \right|_{M=1} \approx \text{const.}$$

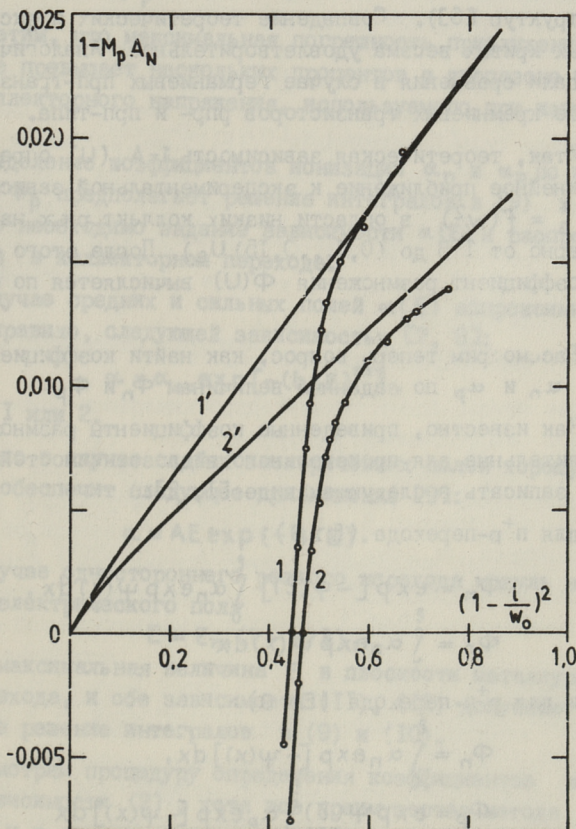


Рис. 1. Экспериментальные (1,2) и теоретические кривые (1', 2') при $M_p = 1$ зависимости $1-M_p A_N$ от $(1-l/w_0)^2$ для транзистора МП13А № 2 (образец № 2): кривые 1, 1' - $I_E = 0,8$ мА, 2, 2' - $I_E = 12,2$ мА.

Для подтверждения обоснованности предложенного приближения на рис. 1 приведены теоретические зависимости $1 - A_N = f[(w/w_0)^2]$, вычисленные по формуле (5) (кривые 1', 2'), и экспериментально снятые зависимости $\Delta I_B^* = f[(w/w_0)^2]$ (кривые 1, 2) для двух токов эмиттера - $I_E = 7,8$ и $12,2$ мА. Измерялся германиевый сплавной транзистор типа МП15А (образец 2) со следующими основными данными: $w_0 = 25$ мкм, время жизни дырок в базе $\tau_p = 2,5$ мкс, радиус эмиттерного перехода $R_E = 250$ мкм, $N = 1,2 \cdot 10^{15}$ см⁻³ и скорость поверхностной рекомбинации у эмиттерного перехода $S = 400$ см·с⁻¹ (характерная величина для травленных поверхностей германиевых структур [6]). Совпадение теоретических и экспериментальных кривых весьма удовлетворительное. Аналогичную картину дали сравнения в случае германиевых ррп-транзисторов, а также кремниевых транзисторов рпр- и рпр-типа.

Итак, теоретическая зависимость $1 - A_N(U)$ определяется как линейное приближение к экспериментальной зависимости

$\Delta I_B^* = f(w^2)$ в области низких коллекторных напряжений (примерно от 1 В до $(0,1 \dots 0,15) U_B$). После этого приведенный коэффициент размножения $\Phi(U)$ вычисляется по формуле (4).

Рассмотрим теперь вопрос, как найти коэффициенты ионизации α_n и α_p по заданным величинам Φ_n и Φ_p .

Как известно, приведенные коэффициенты размножения, действительные для произвольного вида зависимостей $\alpha(E)$, можно записать в следующем виде [1, 2]:

для p^+n -перехода ($E > 0$):

$$\begin{aligned}\Phi_n &= \exp[-\psi(l)] \int_0^l \alpha_n \exp \psi(x) dx, \\ \Phi_p &= \int_0^l \alpha_p \exp \psi(x) dx\end{aligned}$$

и для p^+n -перехода ($E < 0$):

$$\begin{aligned}\Phi_n &= \int_0^l \alpha_n \exp[-\psi(x)] dx, \\ \Phi_p &= \exp \psi(l) \int_0^l \alpha_p \exp[-\psi(x)] dx\end{aligned}$$

где l - толщина рп-перехода

и $\psi(x) \equiv \int_0^x (\alpha_n - \alpha_p) dx'$.

Координатная ось направлена в сторону слаболегированного слоя структуры. В случае низкой интенсивности размножения $\exp(\pm\psi) \approx 1$ и для обоих типов рп-перехода

$$\Phi_n \approx \int_0^l \alpha_n dx, \quad (9)$$

$$\Phi_p \approx \int_0^l \alpha_p dx \quad (10)$$

$$\text{и } \psi(l) \approx \Phi_n - \Phi_p.$$

Отметим, что максимальная погрешность приближений (9) и (10) не превышает нескольких процентов в диапазоне изменения коллекторного напряжения, используемого при измерениях.

Определение коэффициентов ионизации α_n и α_p по измеренным Φ_n и Φ_p предполагает решение интегралов в (9) и (10). Для этого необходимо задание зависимости $\alpha(E)$ и распределения $E(x)$ в коллекторном переходе.

В случае средних и сильных полей $\alpha(E)$ аппроксимируется, как правило, следующей зависимостью [7, 8]:

$$\alpha = \alpha_\infty \exp[-(b/E)^m], \quad (11)$$

где $m = 1$ или 2 .

Однако в случае слабых электрических полей хорошую точность обеспечит следующее приближение [9]:

$$\alpha = AE \exp(-B/E). \quad (12)$$

В случае одностороннего резкого перехода модуль напряженности электрического поля

$$E = E_m (1 - x/l),$$

где E_m — максимальная величина E в плоскости металлургического перехода, и обе зависимости (11), (12) допускают аналитическое решение интегралов в (9) и (10).

Рассмотрим процедуру определения коэффициентов α в случае зависимости (2), хотя все преимущества метода сохраняются и в случае зависимости (11).

Можно показать, что в случае зависимости (12)

$$\Phi = \frac{1}{2} \alpha(E_m) l f(z), \quad (13)$$

где

$$f(z) = 1 - z[1 + ze^z \text{Ei}(-z)],$$

$$z \equiv \frac{B}{E_m}.$$

Отметим, что произведение $\frac{1}{2} \ell f(z)$ можно рассматривать как эффективную толщину коллекторного перехода для процессов размножения носителей заряда:

$$\ell_{ef} = \frac{1}{2} \ell f(z). \quad (I4)$$

Подставляя (I2) в (I3) и учитывая, что $E_m = 2U/\ell$ и $\ell = \alpha U^{1/2}$ ($\Delta\varphi_{ko} \ll U$ в режиме измерения M), получаем

$$\Phi = A U f(z) e^{-z},$$

откуда

$$\ln \left(\frac{\Phi}{f(z) U} \right) = \ln A - \frac{\alpha B}{2} \cdot \frac{1}{\sqrt{U}}. \quad (I5)$$

Формула (I5) определяет процедуру обработки экспериментальной зависимости $\Phi(U)$: результаты измерения, представленные в виде зависимости $\ln[\Phi/(fU)] = f(U^{-1/2})$, аппроксимируются прямой линией, начальная ордината которой дает величину $a_0 \equiv \ln A$, а крутизна величину $b_0 \equiv \alpha B/2$.

Из этих параметров непосредственно следует, что

$$A = e^{a_0}, \quad (I6)$$

$$B = 2b_0/\alpha. \quad (I7)$$

Из (I6) и (I7) видно, что начальная ордината a_0 прямо определяет параметр A , а крутизна b_0 вместе с концентрацией примеси N (см. формулу (8)) – параметр B в формуле Шокли (I2). Соотношение

$$\frac{b_0 e^{a_0}}{\alpha} = \frac{1}{2} AB = \text{const}, \quad (I8)$$

полученное на основе (I6), (I7) может быть использовано для проверки точности определения концентрации N в различных опытных образцах.

Важно подчеркнуть, что для определения A и B по формулам (I6), (I7) не требуется знания величины напряжения лавинного пробоя U_B . Однако для удобства обработки результатов измерения образцов в существенно отличающимися U_B можно вместо U ввести относительное напряжение U/U_B , где U_B есть, например, напряжение, измеренное в режиме пробоя при

некотором максимально допустимом токе коллектора. Как правило, $U_{в\text{эксп}} \leq U_{в\text{теор}}$. В этом случае формула (15) принимает следующий вид:

$$\ln \left(\frac{\Phi}{f(z) \frac{U}{U_b}} \right) = \ln A U_b - \frac{\alpha B}{2\sqrt{U_b}} \left(\frac{U}{U_b} \right)^{-\frac{1}{2}}. \quad (19)$$

Определение A и B по (19) приводит к тем же результатам, что и по формуле (15), независимо от точности экспериментально определенного напряжения пробоя.

Зависимость функции f от z , т.е. от $E_m = f(U, N)$, обуславливает необходимость введения итерационной процедуры для уточнения значений этой функции. В случае слабых электрических полей ($z = 5 \dots 8$) функция f , которая, по существу, равна двукратной относительной толщине эффективного слоя размножения (см. (14)), сравнительно мало зависит от z . Поэтому в качестве начального приближения можно принимать постоянное значение $f = 0,24$, которое соответствует $z \approx 5,6$. Расчеты показали быструю сходимость итерационного процесса.

3. Результаты измерения

Процесс измерения представляет собой снятие зависимости $I_b = f(U_{св})$ при $I_E = \text{const}$ транзисторной структуры в схеме включения с общей базой. Поскольку разогрев испытуемого транзистора существенно влияет на результаты измерений, то измерения проводились на испытательной установке, работающей в импульсном режиме. Измерение базового тока транзистора осуществлялось осциллографическим способом с помощью безындуктивного токового шунта, включенного в цепь базы транзистора.

Измерительная установка состоит из испытательного поля, источника импульсов коллекторного напряжения (ИИKN), источника импульсов эмиттерного тока (ИИЭТ) и устройства управления (УУ). ИИKN осуществляет подачу импульсов напряжения регулируемой амплитудой от 0 до 300 В в коллекторную цепь испытуемого транзистора. Ключевая схема, входящая в состав ИИKN, ограничивает величину тока импульса коллекторного напряжения на уровне 45 мА. ИИЭТ служит для подачи в эмиттерную цепь испытуемого транзистора импульсов тока

стабилизированной амплитуды. Для осуществления возможности испытания транзисторов обоих типов структур (прп и рпр) ИИЭТ состоит из двух каналов, выполненных по типу генераторов тока, управляемых напряжением. Величина импульсов эмиттерного тока каждого из каналов может быть задана в пределах от 0,5 до 10 мА. УУ предназначено для задания режима измерения путем подачи импульсов управления на ИИКН и ИИЭТ, а также импульса для синхронизации осциллографа. Частота следования импульсов коллекторного напряжения может регулироваться в пределах от 3 до 25 Гц, а их длительность в пределах от 150 до 400 мкс. Импульс эмиттерного тока (длительность от 25 до 100 мкс) подается относительно начала импульса коллекторного напряжения с регулируемой задержкой в 100...350 мкс. Импульс синхронизации осциллографа совпадает с началом импульса коллекторного напряжения.

Методика измерения была апробирована на большом количестве промышленных кремниевых и германиевых транзисторов рпр- и прп-типа, изготовленных по различным технологиям. Ниже изложены результаты измерения германиевых структур, поскольку опубликованные данные по измерению α_n и α_p в германии немногочисленны и в некоторых случаях даже противоречивы (см. кривые 2 [III] на рис. 3).

При комнатной температуре были измерены германиевые сплавные транзисторы рпр-типа МП26Б № 1 (образец № 1) при $I_E = 0,94$ и $9,4$ мА, МП15А № 2 (образец № 2) при $I_E = 0,5$; $0,8$ и $12,2$ мА и прп-типа МП37Б № 4 (образец № 3) при $I_E = 7,93$ мА и МП38 № 2 (образец № 4) при $I_E = 0,75$ мА. Концентрации примеси N в базе определялись вольт-емкостным методом. Напряжения пробоя $U_{B_{\text{эксп}}}$ были измерены в конце цикла измерений в режиме $I_E = 0$, $I_C = 30...45$ мА. Основные данные образцов приведены в таблице.

Результаты измерения представлены на рис. 2. Примечателен очень небольшой разброс экспериментальных точек, полученных при различных токах эмиттера. Это является непосредственным подтверждением обоснованности методики определения зависимости $A_N(I_E, U_{CB})$, рассмотренной выше.

Параметры аппроксимирующих прямых (начальная ордината $a_0^* = a_0 + \ln U_B$ и наклон $b_0^* = b_0 / \sqrt{U_B}$) были найдены методом

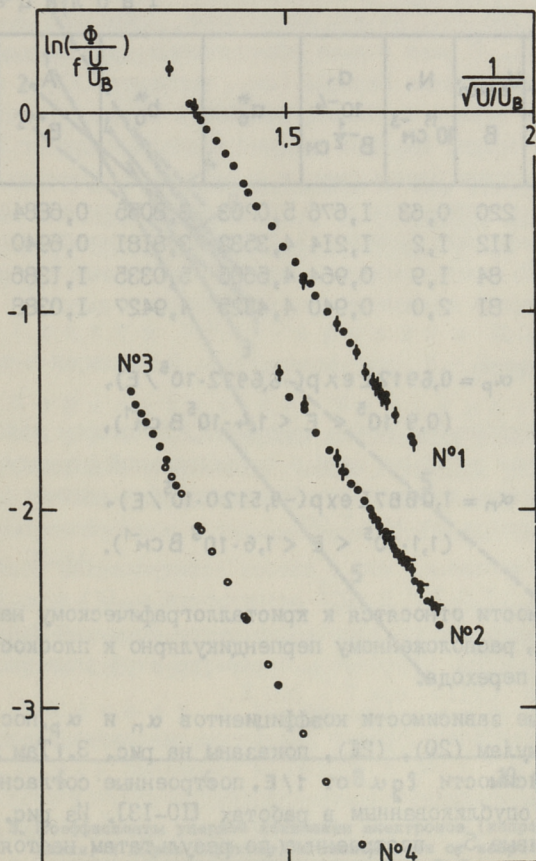


Рис. 2. Результаты измерения образцов №1...4 (см. таблицу):

№ 1 - $I_E = 0,94 \text{ mA}$ (•); $I_E = 9,42 \text{ mA}$ (♦);

№ 2 - $I_E = 0,8 \text{ mA}$ (•); $I_E = 12,2 \text{ mA}$ (♦),

$I_E = 0,5 \text{ mA}$ (↗); № 3 - $I_E = 7,93 \text{ mA}$; № 4 - $I_E = 0,75 \text{ mA}$.

наименьших квадратов. Для уточнения функции $f(z)$ потребовалось до 4...5 итерационных циклов.

Основные результаты обработки экспериментальных данных приведены в таблице.

Если в качестве окончательных A и B принимать арифметические средние A и B двух однотипных образцов, получим

Т а б л и ц а

№ об- раз- ца	Тип струк- туры	$U_{\text{вэксн}}$ В	N , 10^{15} см^{-3}	a , $10^{-4} \text{ В}^{-\frac{1}{2}} \text{ см}$	a_0^*	b_0^*	A , В^{-1}	B , 10^5 В см^{-1}
1	рпр	220	0,63	1,676	5,0203	3,8055	0,6884	6,7376
2	рпр	112	1,2	1,214	4,3532	3,8181	0,6940	6,6567
3	прп	84	1,9	0,964	4,5606	5,0335	1,1386	9,5631
4	прп	81	2,0	0,940	4,4325	4,9427	1,0388	9,4609

$$\alpha_p = 0,6912 E \exp(-6,6972 \cdot 10^5 / E), \quad (20)$$

$$(0,9 \cdot 10^5 < E < 1,4 \cdot 10^5 \text{ В см}^{-1}),$$

$$\alpha_n = 1,0887 E \exp(-9,5120 \cdot 10^5 / E), \quad (21)$$

$$(1,1 \cdot 10^5 < E < 1,6 \cdot 10^5 \text{ В см}^{-1}).$$

Эти зависимости относятся к кристаллографическому направлению $\langle III \rangle$, расположенному перпендикулярно к плоскости коллекторного перехода.

Полевые зависимости коэффициентов α_n и α_p , построенные по формулам (20), (21), показаны на рис. 3. Там же приведены зависимости $\lg \alpha$ от $1/E$, построенные согласно результатам, опубликованным в работах [10-13]. Из рис. 3 видно, что кривые 5, построенные по результатам настоящей работы, вполне удовлетворительно согласуются с кривыми, снятыми при более высоких напряженностях электрического поля. Однако ход кривых 5 свидетельствует об уменьшении наклона кривых в области слабых электрических полей. Аналогичное обстоятельство отмечено и в случае кремния [14].

И, наконец, покажем возможности использования результатов измерения для определения параметров теории ударной ионизации Шокли [9]. Согласно простой модели Шокли

$$A = \frac{q}{r E_R}, \quad (22)$$

$$B = \frac{E_i}{q \lambda_R}, \quad (23)$$

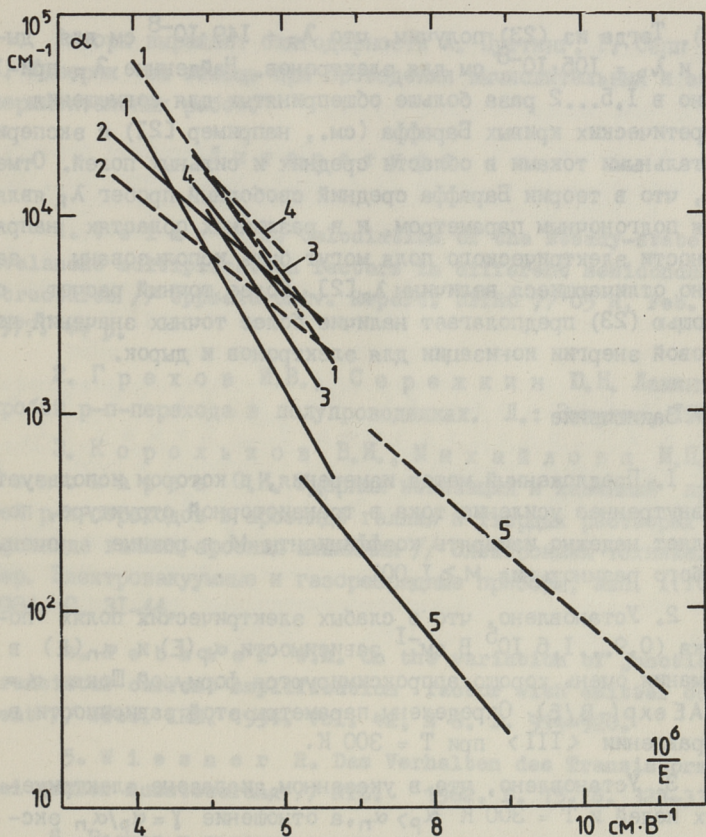


Рис. 3. Коэффициенты ударной ионизации электронов (непрерывная линия) и дырок (пунктир) в зависимости от напряженности электрического поля: кривые 1 [10], 2 [11], 3 [12], 4 [13] и 5 - данная работа.

где E_i - пороговая энергия ионизации;
 E_R - энергия "рамановских" фононов;
 r - среднее количество фононов, возникающих на одну ионизацию;
 λ_R - средний свободный пробег носителя заряда в случае рассеяния на "рамановских" фононах.

Принимая $E_R = 0,037$ эВ [9], из (22) получим $r = 39,1$ для дырок и $r = 24,8$ для электронов. Для оценки длины свободного пробега λ_R предположим одинаково для электронов и дырок $E_i = 1,5 E_g \approx 1$ эВ (для Ge при $T = 300$ К $E_g = 0,66$ эВ

[9]). Тогда из (23) получим, что $\lambda_R = 149 \cdot 10^{-8}$ см для дырок и $\lambda_R = 105 \cdot 10^{-8}$ см для электронов. Найденные λ_R примерно в 1,5...2 раза больше общепринятых для совмещения теоретических кривых Бараффа (см., например [2]) с экспериментальными токами в области средних и сильных полей. Отметим, что в теории Бараффа средний свободный пробег λ_R является подгоночным параметром, и в различных областях напряженности электрического поля могут быть использованы заметно отличающиеся величины λ_R [2]. Более точный расчет с помощью (23) предполагает наличие более точных значений пороговой энергии ионизации для электронов и дырок.

Закключение

1. Предложенный метод измерения, в котором используется внутреннее усиление тока в транзисторной структуре, позволяет надежно измерять коэффициенты M в режиме очень слабого размножения $M \geq 1,001$.

2. Установлено, что в слабых электрических полях порядка $(0,9...1,6 \cdot 10^5 \text{ В см}^{-1})$ зависимости $\alpha_p(E)$ и $\alpha_n(E)$ в германии очень хорошо аппроксимируются формулой Шокли $\alpha = A E \exp(-B/E)$. Определены параметры этой зависимости в направлении $\langle 111 \rangle$ при $T = 300 \text{ К}$.

3. Установлено, что в указанном диапазоне электрических полей и $T = 300 \text{ К}$ $\alpha_p > \alpha_n$, а отношение $\gamma = \alpha_p/\alpha_n$ экспоненциально зависит от электрического поля ($\gamma = 3,7$ и $14,5$ при $E = 1,6 \cdot 10^5$ и $0,9 \cdot 10^5 \text{ В см}^{-1}$, соответственно).

4. Предложенный метод измерения α_n и α_p дает возможность непосредственного определения параметров γE_R и E_i/λ_R простой модели Шокли [9].

5. Для проверки точности предложенного метода измерения был проведен вычислительный эксперимент с помощью программы ДИНАМИТ-1 [15], предназначенный для численного моделирования процессов в многослойных биполярных структурах в одномерном приближении. Погрешность параметров зависимостей $\alpha_n(E)$ и $\alpha_p(E)$, найденных из результатов вычислительного эксперимента, не превышала 4 % от величин, введенных в модель в качестве исходных данных расчета.

Авторы выражают благодарность А. Путтинг, Р. Сепп и К. Кыверик за помощь при проведении вычислительных и экспериментальных работ.

Л и т е р а т у р а

1. V e l m r e E. Calculation of the steady-state avalanche multiplication factors in different semiconductor structures // Uppsala Univ. Report. UPTES 77 05 R, Feb. 1977. 44 p.

2. Г р е х о в И.В., С е р е ж к и н Ю.Н. Лавинный пробой р-п-перехода в полупроводниках. М.: Энергия, 1980.

3. К о р о л ь к о в В.И., М и х а й л о в а М.П., П о н о м а р е в С.И. Ударная ионизация и лавинный пробой р-п-переходов в арсениде галлия и твердых растворах арсениде галлия-арсенид алюминия // Электронная техника. Сер. Электрорадиодетекторы и газоразрядные приборы, вып. I(100). 1984. С. 31-44.

4. W e b s t e r W.M. On the variation of junction transistor current amplification factor with emitter current // Proc. IRE. 1954. Vol. 42, N 6. P. 914-920.

5. W i e s n e r R. Das Verhalten des Transistors bei großer Aussteuerung // NTZ. 1962. B. 15. S. 323-332.

6. Расчет и проектирование полупроводниковых приборов. М.: Оборонгиз, 1963.

7. C h y n o w e t h A. Ionisation rates for electrons and holes in silicon // Phys. Rev. 1958. Vol. 109, N 5. P. 1537-1540.

8. W o l f f P.A. Theory of electron multiplication in Si and Ge // Phys. Rev. 1954. Vol. 95, N 6. P. 1415-1420.

9. S h o c k l e y W. Problems related to p-n junctions in silicon // Solid State Electron. 1961. Vol. 2, N 1. P. 35-67.

10. M i l l e r S.L. Avalanche breakdown in Ge // Phys. Rev. 1955. Vol. 99, N 4. P. 1234-1241.

11. D e c k e r D.R., D u n n Ch.N. Determination

of germanium ionization coefficients from small-signal IMPATT diode characteristics // IEEE Trans. Electron Dev. 1970. Vol. ED-17, N 4. P. 290-299.

12. Dai B.T., Chang C.Y. Temperature dependence of ionization rates in Ge // J. Appl. Phys. 1971. Vol. 42, N 12. P. 5198-5201.

13. Mikawa T., Kagawa S., Kaneda T., Toyama Y., Mikami O. Crystal orientation dependence of ionization rates in germanium // Appl. Phys. Lett. 1980. Vol. 37, N 4. P. 387-389.

14. Кузьмин В.А. Крюкова Н.Н., Кургян А.С., Мнацаканов Т.Т., Шуман В.Б. О коэффициентах ударной ионизации электронов и дырок в кремнии // ФТП. 1975. Т. 9, в. 4. С. 735-738.

15. Велмре Э.Э., Удал А.Э. Программа "ДИНАМИТ-1" для одномерного численного моделирования тиристорных структур // Силовые полупроводниковые приборы: Сб. статей. Таллинн, Валгус. 1986. С. 58-63.

E. Velmre, M. Pikkov

Põrkeionisatsioonitegurite mõõtmine pn-siirdes
väga nõrga paljunemisintensiivsuse korral

Kokkuvõte

Töös on esitatud pooljuhi elektronide ja aukude põrkeionisatsioonitegurite mõõtmismeetod juhuks, kui paljunemisintensiivsus on väga nõrk ($M \approx 1,002$). Meetod põhineb normaalses aktiivrežimis töötava bipolaartransistori struktuuri baasivoolu mõõtmisel sõltuvalt kollektorsiirdele rakendatud pingest. Kirjeldatud meetodil on mõõdetud elektronide ja aukude põrkeionisatsioonitegurid germaaniumis kristallograafilises suunas $\langle 111 \rangle$ temperatuuril $T = 300\text{ K}$ ja elektrivälja tugevuste vahemikus $9 \cdot 10^4 \dots 1,6 \cdot 10^5 \text{ V.cm}^{-1}$.

E. Velmre, M. Pikkov

Ionization Coefficient Measurements in a PN-Junction
at Very Low Multiplication Rates

Abstract

A method for electron and hole ionization coefficient measurements in a pn-junction at very low multiplication rates ($M \approx 1,002$) has been described. The method is based on the base current versus collector junction voltage measurements in the normal active regime of a bipolar transistor structure. The electron and hole ionization coefficients in germanium in the direction $\langle 111 \rangle$ at $T = 300\text{ K}$ with electric fields in the range of 9×10^4 to $1,6 \times 10^5 \text{ V.cm}^{-1}$ have been evaluated.

СПЕЦИАЛИЗИРОВАННЫЕ БАЗЫ ДАННЫХ ДЛЯ ПРОГРАММНЫХ СРЕДСТВ МОДЕЛИРОВАНИЯ И АНАЛИЗА ЭЛЕМЕНТОВ ТВЕРДОТЕЛЬНОЙ ЭЛЕКТРОНИКИ

Введение. Создание разных прикладных пакетов программ (ППП) для анализа и моделирования элементов твердотельной электроники (ЭТЭ) содержит и создание т.н. специализированных баз данных (СБД). Как показано в работах [1, 2] СБД должны быть включены в содержание ППП на всех уровнях моделирования. В данной работе рассматриваются основные требования для создания СБД и приводится описание СБД для ППП моделирования и анализа полупроводниковых структур. В статье не рассматриваются теоретические основы создания СБД. При разработке СБД используются стандартные решения типа DATA BASE.

Требования к СБД. Большая вычислительная сложность задач моделирования и анализа ЭТЭ приводит к проблеме эффективного использования вычислительных ресурсов. Поэтому возникает необходимость для статистической обработки данных и создание специализированных программных средств обработки данных. Исходя из вышесказанного, данные в СБД могут быть разделены на две группы:

- 1) параметры (данные), определяемые экспериментальными исследованиями;
- 2) параметры (данные), которых невозможно получить экспериментальным путем.

В первом случае в составе СБД должны быть специальные программные средства, которые оценивают имеющиеся уже данные в СБД, а также обрабатывают данные при введении новых. Во втором случае в состав СБД включаются средства с помощью которых обрабатываются экспериментальные данные.

Кроме вышеуказанных общих требований следует определить и некоторые специфические свойства СБД для ППП ЭТЭ:

1) описание разных физических и химических явлений с помощью разных моделей (многомодельность);

2) включение в состав СБД возможности пользоваться собственными моделями пользователя;

3) в случае неопределенности параметров (данных) в СБД программа выбирает из СБД т.н. стандартные величины и автоматически блокирует неопределенные (а также неточные) величины параметров.

Таким образом, для каждого ППП имеются свои СБД, учитывающие все особенности ППП. Например, СБД ППП IMPL содержит данные для математических параметров, которые требуются для определения распределения примеси, также данные кривых Пирсона и Гаусса и т.д. Для ППП DYNAMIT/SCNOTSIM требуются данные для определения подвижностей носителей, лавинного размножения и др.

Описание СБД для ППП DYNAMIT/SCNOTSIM. ППП DYNAMIT/SCNOTSIM предназначены для моделирования и анализа одномерных биполярных приборов и структур Шоттки. В работе [1] показана графмодель такой ППП. Следовательно, в СБД должны быть отражены следующие физические модели:

- 1) изменения ширины запрещенной зоны;
- 2) определения собственной концентрации носителей;
- 3) определения эффективных масс носителей;
- 4) определения времен жизни носителей;
- 5) определения подвижностей носителей;
- 6) размножения носителей;
- 7) определения высоты барьера Шоттки.

В данной статье не рассматриваются конкретные модели, так как они с достаточной точностью отражены в литературе [3-8].

Для СБД используются следующие EXEC программы:

SSS-EXEC - программа трансляции FORTRAN программы в случае, когда отсутствуют рабочие модули программы. Транслируются FORTRAN программы SQ, SFORMA и SELECT. Листинг стирается.

SS.EXEC – программа выводит на экран дисплея имена и номера имеющихся таблиц и впускает программу SO FORTRAN.

FORTRAN – программы:

SO FORTRAN – основная программа. Спрашивает номера таблиц с помощью соответствующих подпрограмм. Таблицы выводятся на экран дисплея вместе с заглавиями (объем 60 строк). По желанию имеется возможность рассматривать более одной таблицы. Работа заканчивается только после подачи отрицательного ответа.

SELECT FORTRAN – подпрограмма, которая сравнивает символы с цифрами (0...9) (объем 16 строк).

SFORMA FORTRAN – подпрограмма, которая описывает три формата для требуемой таблицы. Эти форматы следующие:

- 1) формат для чтения данных из файла в цифровом виде;
- 2) формат для полного чтения данных из файла;
- 3) формат для выведения данных на экран дисплея (объем 75 строк).

LININTØ FORTRAN – подпрограмма для обработки новых входных данных. Имеются три разные версии обработки данных LIN1, LIN2 и LIN3 (объем 130 строк).

Кроме вышеуказанных программ СБД содержит еще несколько вспомогательных программ, которые защищают СБД от внешнего, нежелательного доступа, а также для организации печати и т.д.

Выводы. Описана СБД для ППП DYNAMIT/SCHOTSIM. Вышеуказанный подход позволяет пользователю выбрать более подходящую модель описания конкретного физического (химического, математического) явления. Следовательно, СБД является универсальным и открытым снизу и поэтому может быть использована и в других ППП ЭТЭ.

Л и т е р а т у р а

1. Р а н г Т. Комплексный подход к системам проектирования элементов полупроводниковой и интегральной электроники // Тр. Таллинск. политехн. ин-та. 1988. № 674. С. 149-164.
2. Р а н г Т. Инженерное обеспечение САПР элементов ФЭА: Сб. тез. респ. совещ. Численные методы и средства проектирования и испытания элементов ФЭА. Таллин, 7-II дек. 1987. С. 12-23.
3. Р а н г Т. Моделирование подвижностей носителей I. Рассеивание на акустических фоновых и на нейтральной и ионизированной примеси // Тр. Таллинск. политехн. ин-та. 1986 № 629. С. 103-116.
4. Р а н г Т. Моделирование подвижностей носителей II. Рассеивание на носителях и поверхности, эффект насыщения дрейфовой скорости носителей // Тр. Таллинск. политехн. ин-та. 1986. № 629. С. 117-130.
5. Р а н г Т. Моделирование концентрации носителей заряда в полупроводниках // Тр. Таллинск. политехн. ин-та. 1986. № 639. С. 26-36.
6. R a n g T. Computer aided examination of the avalanche multiplication in Si, GaAs and InP in complementary pn-junctions // Modeling, Simulation and Control, A, AMSE Press. 1985. Vol. 3, N 1. P. 23-40.
7. Р а н г Т. Коэффициенты ударной ионизации носителей и их температурная зависимость в кремнии // Изв. вузов СССР. Радиоэлектроника. 1985. 28. С. 83-85.
8. Р а н г Т., Пуусепп М. Коэффициенты ударной ионизации носителей заряда в <100> арсениде галлия // Электронная техника. Сер. 2. Полупроводниковые приборы. 1987. Вып. 2 (187). С. 98-100.

S. Orro, T. Rang

Tahkekehaseadiste modelleerimisvahendite
spetsialiseeritud andmebaas

Kokkuvõte

Artiklis on kirjeldatud globaalseid ja spetsiifilisi nõudeid, mida esitatakse spetsialiseeritud andmebaasidele (SAB), mis on kasutusel tahkekehaseadiste modelleerimis-/analüüsiprogrammides. Töös kirjeldatakse konkreetset SAB-d bipolaarsete ja Schottky pooljuhtstruktuuride modelleerimise/analüüsi programmipakettide tarbeks.

S. Orro, T. Rang

Special Data Base of Solid-State
Device Analysis Programs

Abstract

Global and special demands for special data base (SPD) are dealt with. The SPD for analysis programs of bipolar and Schottky semiconductor structures is described.

МАТРИЦЫ ФУРЬЕ И ОБОБЩЕННАЯ ФОРМА ПРЯМОГО
ПРОИЗВЕДЕНИЯ

Известно, что на базе обобщенного ядра можно формировать весьма многообразные матрицы, такие как матрицы Адамара, Фурье, Хаара и т.д. [1, 2].

В случае матриц Адамара имеем одинаковый набор ядер и в соответствии с [3] каждый элемент такой синтезируемой матрицы размерностью $N \times N$ можно получить на основании выражения

$$A_{\vec{z}, \vec{u}} = \prod_{i=1}^n \alpha^i_{z_i u_i}, \quad (1)$$

где $\alpha_{z_i u_i}$ — элемент обобщенного ядра Адамара, и представляющего собой для основания исчисления равного двум, квадратную матрицу размером 2×2 :

$$\begin{aligned} \alpha^i &= [\alpha_{\vec{z}_i \vec{u}_i}] = \\ &= \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \begin{bmatrix} \alpha^i_{00} & \alpha^i_{01} \\ \alpha^i_{10} & \alpha^i_{11} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \end{aligned} \quad (2)$$

$\vec{z}$ и $\vec{u}$ — векторы строк и столбцов индексов элементов матрицы A , представленные набором двоичных чисел $\vec{z} = z_1 z_2 \dots z_n$, $\vec{u} = u_1 u_2 \dots u_n$. При этом $N = 2^n$.

Поскольку все ядра Адамара одинаковы, то, например, для случая $N = 2^3$ имеем четыре ядра $\alpha^1 = \alpha^2 = \alpha^3 = \alpha^4 = \alpha$, а крснекеровское представление (1) будет при этом в соответствии с [1, 4] выглядеть как (3)

$$A_{\vec{z}, \vec{u}} = \alpha^1 \otimes [\alpha^1, \alpha^2] \otimes [\alpha^1, \alpha^2, \alpha^3, \alpha^4]. \quad (3)$$

Элемент матрицы Адамара $A_{ij} = A_{5,6}$ к примеру, на основании (I) будет:

$$\begin{aligned} A_{5,6} &= A_{z_3 z_2 z_1, u_3 u_2 u_1} = A_{101, 110} = \alpha_{z_1, u_1}^1 \cdot \alpha_{z_2, u_2}^2 \cdot \alpha_{z_3, u_3}^3 = \\ &= \alpha_{10}^1 \cdot \alpha_{01}^2 \cdot \alpha_{11}^3 = \alpha_{10} \cdot \alpha_{01} \cdot \alpha_{11} = 1 \cdot 1 \cdot (-1) = -1. \end{aligned}$$

Известно, что обобщенное ядро позволяет трактовать матрицы Адамара как частный случай матриц Фурье, которые, однако, имеют разные ядра. Для фурьеподобных матриц, в соответствии с [4] кронекеровское представление для случая, например, $2^n = 2^3 = N$ представлено следующим выражением:

$$B_{\vec{z}, \vec{u}} = [\alpha^1] \otimes [\alpha^1, \alpha^2]^T \otimes [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T, \quad (4)$$

из которого видна повторяемость четырех различных ядер в первом и втором сомножителях (T - знак транспонирования).

Нахождение элементов синтезируемой при этом матрицы Фурье будет отличаться от (I) и в соответствии с [5] иметь вид формы произведения:

$$B_{\vec{z}, \vec{u}} = \prod_{i=1}^n \alpha_{z_i u_i}^{i, l}, \quad (5)$$

где

$$l = 1 + z_{i+1} \cdot 2^0 + z_{i+2} \cdot 2^1 + \dots + z_n \cdot 2^{n-2},$$

$$i = \overline{1, n} \quad l = \overline{1, \frac{N}{2}}. \quad (5a)$$

Отметим, что l - это номер микроблока в матрице Гуда, а i -я матрица Гуда [II].

Пример: $N = 2^n = 2^3$; $i = 1, 2, 3, 4$; тогда для $B_{5,6}$ имеем

$$i = 1 \quad l = 1 + z_2 \cdot 1 + z_3 \cdot 2 = 3$$

$$i = 2 \quad l = 1 + z_3 \cdot 1 = 2$$

$$i = 3 \quad l = 1 + z_3 \cdot 2 = 3$$

$$B_{101, 110} = B_{z_3 z_2 z_1, u_3 u_2 u_1} = \alpha_{z_1 u_1}^{1,3} \cdot \alpha_{z_2 u_2}^{2,2} \cdot \alpha_{z_3 u_3}^{3,1} = \alpha_{10}^{1,3} \cdot \alpha_{01}^{2,2} \cdot \alpha_{11}^{3,1}.$$

Если обозначить обобщенное ядро в виде

$$\alpha_{z_i u_i}^{i, l} = \begin{matrix} z_i \backslash u_i & 0 & 1 \\ 0 & a_{il} & b_{il} \\ 1 & c_{il} & d_{il} \end{matrix}, \quad (6)$$

то

$$B_{101,110} = c_{13}^{1,3} \cdot b_{22}^{2,2} \cdot d_{31}^{3,1}.$$

Далее в выражении (7) в качестве иллюстрации данного примера и для дальнейшего обобщения (4) представлены подчеркнутые элементы в произведениях факторизованных матриц Гуда [2, 6]:

$$B_{z, \bar{z}} = \begin{matrix} l=1 \\ l=2 \\ l=3 \\ l=4 \end{matrix} \left[\begin{array}{ccc|ccc} \underline{a_{31}} & & & & b_{31} & \\ c_{31} & & & & d_{31} & \\ & a_{32} & & & b_{32} & \\ & c_{32} & & & d_{32} & \\ & & a_{33} & & b_{33} & \\ & & c_{33} & & d_{33} & \\ & & & a_{34} & b_{34} & \\ & & & c_{34} & d_{34} & \end{array} \right] \times$$

$$\begin{matrix} \times \\ \times \end{matrix} \left[\begin{array}{ccc|ccc} a_{21} & & & & b_{21} & \\ c_{21} & & & & d_{21} & \\ & a_{22} & & & b_{22} & \\ & c_{22} & & & d_{22} & \\ & & a_{23} & & b_{23} & \\ & & c_{23} & & d_{23} & \\ & & & a_{24} & b_{24} & \\ & & & c_{24} & d_{24} & \end{array} \right] \times \left[\begin{array}{ccc|ccc} a_{11} & & & & b_{11} & \\ c_{11} & & & & d_{11} & \\ & a_{12} & & & b_{12} & \\ & c_{12} & & & d_{12} & \\ & & a_{13} & & b_{13} & \\ & & c_{13} & & d_{13} & \\ & & & a_{14} & b_{14} & \\ & & & c_{14} & d_{14} & \end{array} \right] =$$

$$= G_3 \cdot G_2 \cdot G_1.$$

Отметим, что выражения (4) и (7) будут эквивалентны, если микроблоки в матрице G_3 разные при $l = 1, 2, 3, 4$; в матрице G_2 первый и второй микроблоки при $l = 1, 2$ совпадают с аналогичными микроблоками при $l = 1, 2$; в матрице G_3 при $l = 3, 4$ они для G_2 повторяют микроблоки при $l = 1, 2$ соответственно; для матрицы Гуда G_1 все микроблоки при $l = 1, 2, 3, 4$ равны и эквивалентны первому микроблоку матрицы G_3 .

Представляет несомненный интерес рассмотреть дальнейшее обобщение формулы (5) для случая всех неодинаковых мик-

роблоков в разных матрицах Гуда, но с расположением их в соответствии с выражением (7). В [6] такое обобщение названо обобщенной формой произведения. Поскольку по алгоритму (7) получаются фурьеподобные матрицы, логично называть алгоритм (7) как получение фурьеподобных матриц. Представляет также интерес рассмотреть алгоритмы формирования фурьеподобных матриц, которые вообще не потребуют операций формирования матриц Гуда [2].

Действительно, если обозначить микроблоки в (7) через обобщенные ядра $\alpha^{i,l}$, то кронекеровское представление (7) примет вид:

$$B_{\vec{z}, \vec{u}} = [\alpha^{1,1}, \alpha^{1,2}, \alpha^{1,3}, \alpha^{1,4}]^T \otimes [\alpha^{2,1}, \alpha^{2,2}, \alpha^{2,3}, \alpha^{2,4}]^T \otimes [\alpha^{3,1}, \alpha^{3,2}, \alpha^{3,3}, \alpha^{3,4}]^T, \quad (8)$$

где учтен тот факт, что при обычном матричном перемножении слева-направо кронекеровское представление имеет на базе этого обычного произведения справа-налево [5]. Важным при этом является то, что микроблоки в матрицах Гуда занимают везде одинаковые позиции, поэтому можно воспользоваться теоремой, данной в [6].

Теорема

Пусть $\alpha^{i,l}$ ($i = 1, \overline{n}$; $l = 1, \overline{\frac{N}{2}}$ параметры) образуют совокупность квадратных и, в общем случае, неодинаковых матриц размерностью 2×2 и пусть G_l также образуют совокупность квадратных, размерностью $2^n \times 2^n = N \times N$, матриц с различными ненулевыми элементами, но одинаковых по своей структуре (матриц Гуда), тогда выполняются равенства:

$$B_{\vec{z}, \vec{u}} = \otimes \prod_{l=1}^n \alpha^{i,l} = \prod_{l=1}^n G_{n-i-1} \quad (9a)$$

$$B_{\vec{z}, \vec{u}} = \prod_{l=1}^n \alpha_{z_i, u_i}^{i,l}$$

$$\begin{aligned} l &= z_n, \dots, z_2 & \text{при } i &= 1 \\ l &= u_1, z_n, \dots, z_3 & \text{при } i &= 2 \\ &\vdots & & \\ l &= u_{n-1}, \dots, u_n & \text{при } i &= n \end{aligned} \quad (9b)$$

В $\alpha - i$ - номер матрицы Гуда, l - номер микроблока этой матрицы, $z_i u_l$ - номер элемента в микроблоке.

Исходя из вышеизложенного, для получения элементов матрицы $B_{\vec{z}, \vec{u}}$ через элементы матриц - обобщенных ядер $\alpha^{i,l}$ (выражение 9б) для $n = 3$ имеем:

$$B_{z_3 z_2 z_1, u_3 u_2 u_1} = \prod_{i=1}^3 \alpha_{z_i u_i}^{i,l} = \alpha_{z_1 u_1}^{1, z_3 z_2} \cdot \alpha_{z_2 u_2}^{2, z_3 u_1} \cdot \alpha_{z_3 u_3}^{3, z_3 u_1}. \quad (10)$$

Для получения элементов матрицы $B_{\vec{z}, \vec{u}}$, исходя из обобщенных ядер, достаточно по правилам (9б) определить номер матрицы Гуда i - первый параметр, номер микроблока l - второй параметр и конкретный элемент обобщенного ядра - $z_i u_l$.

Например, для элемента синтезируемой матрицы $B_{\vec{z}, \vec{u}} = B_{\vec{5}, \vec{6}} = B_{z_3 z_2 z_1, u_3 u_2 u_1} = B_{101, 110}$, в соответствии с (10) и (7), имеем:

$$B_{5,6} = \alpha_{1,0}^{1,10} \cdot \alpha_{0,1}^{2,10} \cdot \alpha_{1,1}^{3,11} = c_{13} \cdot b_{23} \cdot a_{34}.$$

Видно отличие полученного результата от предыдущего случая, при котором не все ядра различные. Это отличие обусловлено индексом i , который определяется либо формулой (5а), либо формулой (9б)

На основании вышесказанного можно сформировать алгоритмы расчета синтезируемых матриц, в том числе и фурьеподобных матриц, опираясь при этом лишь на элементы обобщенных ядер.

Один из возможных алгоритмов приводится ниже, и он выражается в следующем.

1. На базе матрицы десятичных индексов строк I и столбцов J матрицы $N \times N$ формируются их двоичные представления, которые заносятся в трехмерный массив $A(I, J, K)$, где с первой половиной заполняется двоичное представление строк $z_n \dots z_1$, а со второй - столбцов $u_n \dots u_1$. При этом $k = \overline{1, 2n}$.

2. Индексация первого и последнего элементов синтезируемой матрицы реализуется следующим образом. Для номера элемента в микроблоке $z_i u_l$ как

$IL = A(I, J, k) \cdot 2^1 + A(I, J, n+1) \cdot 2^0$ для первого элемента матрицы

и

$IL1 = A(I, J, n) \cdot 2^2 + A(I, J, 2n) \cdot 2^0$ для последнего элемента матрицы

Для микроблока l как:

$IS = A(I, J, n) \cdot 2^{n-2} + \dots + A(I, J, 2) \cdot 2^0$ для первого элемента матрицы

и

$IS1 = A(I, J, 2n-1) \cdot 2^{n-2} + \dots + A(I, J, n+1) \cdot 2^0$ для последнего элемента матрицы

3. Остальные элементы синтезируемой матрицы в соответствии с (96), при $i \neq 1$, $i \neq n$ находятся следующим образом.

Для номера элемента в микроблоке $z; u_i$ как:

$ILR(i) = A(I, J, i) \cdot 2^i + A(I, J, i+n) \cdot 2^0$

Для номера микроблока l как:

$ISR(i) = A(I, J, i+1) \cdot 2^0 + \dots + A(I, J, n+i-1) \cdot 2^{n+i-1}.$

Таким образом приведен алгоритм формирования из десятичных индексов I и J индексов-номеров элементов микроблоков $z; u_i$, индексов-номеров самих микроблоков l для каждой ячейки синтезируемой матрицы и индексов матрицы Гуда i . Остается предоставить или сформировать сами ядра-микроблоки с индексацией их и их элементов по тем же алгоритмам, после чего произвести непосредственный синтез матрицы. Для матрицы Фурье такое формирование ядер-микроблоков с их индексацией подробно приведено в [2].

Поскольку, как это ясно из вышеизложенного, можно при других Θ получать и другие ядра, то на базе данной программы можно формировать и фурьеподобные матрицы [4].

Для базы $N = 2^3$, 2^4 и 2^5 на ЭВМ IO46 время работы процессора, формирующего такие матрицы, составляет примерно 1,5, 3, 10 секунд.

Для формирования тех же матриц на базе факторизованных матриц Гуда это время, при прочих равных условиях, составило соответственно 1,6, 4 и 16 секунд.

Таким образом обобщенная форма прямого произведения (ОФП) позволяет сократить время формирования матриц Фурье, причем тем эффективней, чем больше их размерности.

Л и т е р а т у р а

1. Солодовников А.И. и др. Синтез ортогональных базисов на основе обобщенного спектрального ядра // Вопросы теории систем автоматического управления. Вып. 2. Л.: 1976. С. 99-112.

2. Гурьянов Б.Г. и др. Гудовские преобразования сигналов в систолических структурах // Тр. Таллиннск. политехн. ин-та. 1987. № 651. С. 125-131.

3. Good I.J. The interaction algorithm and practical Fourier analysis // J. Roy. Statist. Soc. (London). 1958. В 20. Р. 361-372.

4. Гурьянов Б.Г. Фурье и Хаараподобные матрицы. Тезисы докладов Республиканской научно-технической конференции НТОРЭС. Таллинн, 1988.

5. Солодовников А.И. Синтез полных ортонормированных функций, имеющих алгоритм быстрого преобразования // Вопросы теории систем автоматического управления. Вып. 4. Л.: 1978. С. 94-105.

6. Солодовников А.И., Спиваковский А.М. Основы теории и методы спектральной обработки информации. Л.: Изд-во ЛГУ, 1986.

B. Gurjanov

Fourier' matriksid ja otsekorrutamise üldistatud
kuju

Kokkuvõte

Artiklis vaadeldakse Fourier' matriksite formeerimise algoritmi, mis põhineb otsekorrutamise üldistatud kujul ja üldistatud tuumal.

Antud algoritm on efektiivsem teistest tuntud meetoditest ja seda suurte matriksite puhul.

B. Guryanov

Fourier Matrix and Generalised Form of the Direct
Product

Abstract

This paper describes the algorithm forming Fourier matrix, based on the generalized form of the direct product and on the generalized kernel.

This algorithm is more efficient than the known matrices, especially those for large dimensions.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ СИНТЕЗА КОНТРОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ УПРАВЛЯЮЩИХ АВТОМАТОВ

Введение. В современных условиях резкого возрастания сложности дискретных управляющих устройств (ДУУ) процесс их тестирования становится все более трудоемким и дорогостоящим. Как известно, затраты на тестирование зависят существенно от того, на каком этапе проектирования ДУУ проектировщик начинает заботиться о контролепригодности устройства. Учет контролепригодности уже на этапе логического проектирования, а также регулярное тестирование отдельных шагов синтеза позволяет во многом сэкономить средства на тестирование готовых ДУУ. Очевидно, что такой подход предполагает существование теста, который применим для верификации как промежуточных, так и конечных результатов процесса синтеза ДУУ. Один из наиболее универсальных методов синтеза такого теста является метод Хенни [1] построения контрольных последовательностей (КП) для конечных автоматов. КП, построенная на основе конечно-автоматной модели устройства, дает возможность по выходной реакции автомата отличить исправный автомат от всех других автоматов с такими же входными и выходными алфавитами и с таким же множеством внутренних состояний. Метод Хенни нашел до сих пор довольно узкое практическое применение, поскольку при описании реальных ДУУ в большинстве случаев использование модели конечного абстрактного автомата является нецелесообразным. Основные принципы модификации подхода Хенни для модели микропрограммного автомата (МПА) [2] были изложены в [3]. Построение КП для сетей автоматов, полученных в результате декомпозиции исходного автомата рассматривалось в [4, 5].

В настоящей работе приведены результаты обширного экспериментального исследования методов [3, 4, 5] синтеза КП для МПА и его сетевой реализации. Эффективность КП оценивается длиной КП, временем ее построения и полнотой КП. Результаты эксперимента показано, что предложенные методы синтеза КП позволяют обнаружить большинство одиночных константных неисправностей в схемных реализациях ДУУ на базе МИС. Описана также подсистема синтеза КП, которая в составе автоматизированной системы проектирования ДУУ DILOS внедрена в инженерную практику проектирования.

Основные понятия. В качестве автоматной модели устройства будем использовать МПА, позволяющий компактно описать ДУУ высокой степени сложности. Микропрограммным автоматом называем систему $A = (S, \{0,1\}^L, \{0,1\}^N, \delta, \lambda)$, где S — множество внутренних состояний; $\{0,1\}^L$ — входной алфавит; L — число входных двоичных каналов; $\{0,1\}^N$ — выходной алфавит; N — число выходных двоичных каналов; $\delta: S \times \{0,1\}^L \rightarrow S$ — функция переходов; $\lambda: S \rightarrow \{0,1\}^N$ — функция выходов.

В дальнейшем предполагаем, что рассматриваемый нами МПА типа Мура инициальный, минимальный и сильносвязанный [2]. Как известно, многие задачи в области синтеза ДУУ эффективно решаются на основе метода декомпозиции управляющих автоматов. При этом исходный автомат реализуется в виде сети взаимосвязанных компонентных автоматов [6]. Такая сеть строится на основе полной системы разбиений [6] на множестве внутренних состояний автомата.

Методы построения КП для МПА (для сетей МПА) предполагают, что исходный автомат (сеть) контролируемый. Называем МПА контролируемым, если для него существует диагностическая последовательность (ДП) [1]. Сеть МПА называем контролируемой [5], если для каждого ее компонентного автомата существует ДП.

Существование и длина ДП (а, следовательно, и длина всей КП) во многом определяется выходным разбиением МПА. Разбиение π_μ называем выходным разбиением МПА, если для всех $s, t \in S$ выполняется условие $s \equiv t(\pi_\mu) \Leftrightarrow \lambda(s) = \lambda(t)$, где запись $s \equiv t(\pi_\mu)$ означает, что состояния s и t находятся в одном блоке разбиения π_μ . Очевидно, что при тривиальном случае, если $\pi_\mu = O_\mu$ (разбиение, в каждом блоке

которого одно состояние), все состояния МПА непосредственно наблюдаются по выходным переменным и ДП имеет нулевую длину.

Подсистема ПОСТРЭКС построения КП для МПА и сетей МПА.
По результатам исследований [3, 4, 5] был разработан ряд алгоритмов построения КП для МПА и для его сетевых реализаций. Часть из этих алгоритмов была реализована в виде подсистемы автоматизированной системы проектирования ДУУ DILOS, созданной на кафедре ЭВМ ТПИ. Система DILOS позволяет сетевые реализации сложных ДУУ как на базе малых, так и на базе средних и больших интегральных схем (включая программируемые логические матрицы и микропроцессорные наборы), обладает средствами моделирования на различных уровнях представления ДУУ. Дополнение системы DILOS подсистемой ПОСТРЭКС существенно расширяет область ее практического применения, приближает ее к требованиям, выдвигаемым практикой проектирования. Общая схема работы подсистемы ПОСТРЭКС, которая представляет собой иерархическую структуру макрооперации языка ЛЯПАС-М, приведена на рис. 1.

Опишем основные задачи, решаемые подсистемой ПОСТРЭКС:

1. Построение обхода графа МПА (алгоритм А). Обход графа совпадает с КП для МПА, если выходное разбиение $\pi_p = O_\pi$.

2. Построение КП для МПА, выходное разбиение которого $\pi_p \neq O_\pi$ (алгоритм Б). Данный алгоритм включает задачу нахождения ДП для МПА.

3. Построение КП для сети МПА, построенной по данной системе разбиений, включая построение ДП для компонентных автоматов сети (алгоритм В).

Головной модуль ПОСТРЭКС обращается к модулям ПОСЛМПА (построение последовательностей для МПА) и ПОСЛСЕТ (построение последовательностей для сетей МПА). Модуль ПОСЛМПА реализует алгоритмы А и Б, модуль ПОСЛСЕТ – алгоритм В. Результаты работы обоих модулей (т.е. соответствующие контрольные последовательности) записываются в архив системы. В подсистеме ПОСТРЭКС (как и во всей системе DILOS) имеются следующие ограничения на параметры МПА: число состояний – до 511, число входных каналов – до 127, число выходных каналов – до 255. Экспериментальное исследование проводилось на двух потоках МПА:

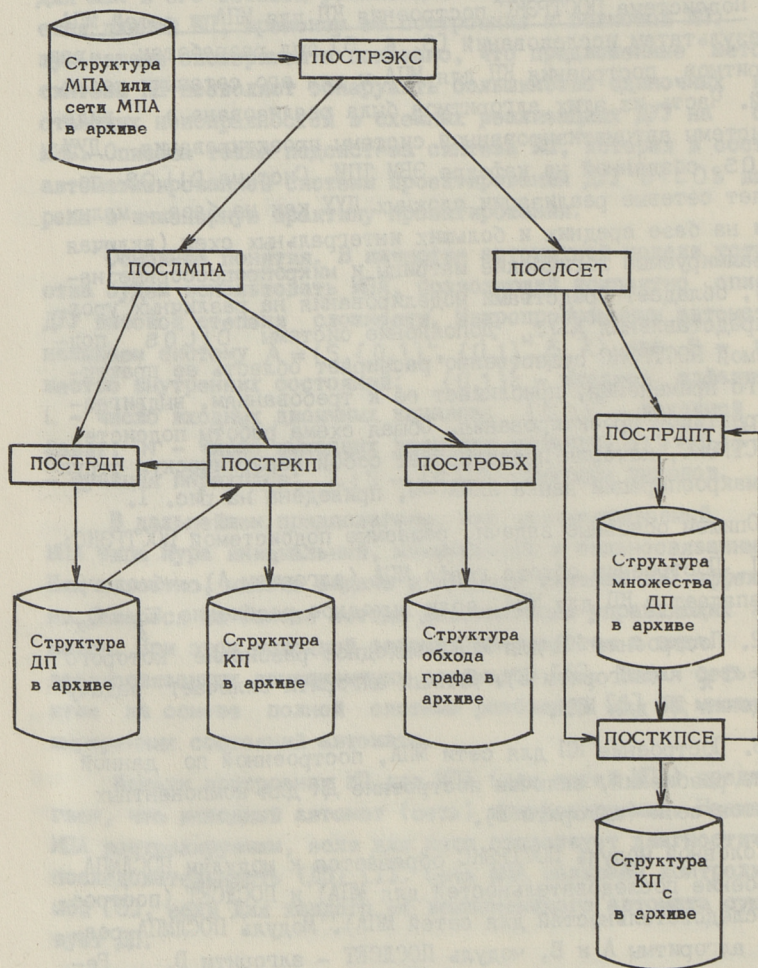


Рис. 1. Общая схема работы сегментов подсистемы ПОСТРЭКС.

1. МПА, получаемые при помощи существующей в системе программы генерации псевдослучайных автоматов с различными, наперед заданными параметрами;

2. МПА, разработанные в различных организациях страны при построении реальных ДУУ (библиотека УАССМА-ЕС). При экспериментальном исследовании алгоритмов и программ построения КП для МПА и сети МПА были сравнены длины последовательностей, полученных в результате работы подсистемы ПОСТРЭКС, а также исследованы временные характеристики разработанных программ. Результаты экспериментов приведены в таблицах 1 и 2, соответственно для реальных и псевдослучайных МПА, где

M – количество состояний МПА;

L – количество двоичных входов;

N – количество двоичных выходов;

H – длина таблицы МПА;

d_1 – длина ДП для МПА;

T_1 – время расчета ДП;

d_2 – длина КП для МПА;

T_2 – время расчета КП;

n – количество компонентных автоматов в контролируемой сети;

d_3 – длина КП для сети МПА;

T_3 – время расчета КП для сети МПА.

Исследование проводилось на ЦВМ ЕС-1055 в операционной системе СВМ. Эксперименты подтвердили предположение [3], что для всех МПА ДП очень короткая и состоит для большинства МПА лишь из одного входного набора. Длина КП превысила примерно 1,5.. 2,0 раза параметра N – количество строк в таблице МПА

КП для сетевой реализации МПА в большинстве случаев несколько длиннее КП для МПА. Это объясняется большим числом связей в сетях. При этом длина КП для отдельного компонентного автомата всегда короче для исходного МПА. В экспериментах участвовали сети, состоящие из 2...4 компонентных автоматов. Все исследованные сети были контролируемые.

Полнота построенных КП. Одной из наиболее важных характеристик оценки реальной эффективности КП является отношение количества обнаруживаемых неисправностей к количеству

всех рассматриваемых неисправностей (полнота КП). Оценивание полноты построенных КП проводилось с помощью разработанной в Институте кибернетики АН ЭССР системы автоматизированного синтеза тестов (САСТ) [7]. Были использованы следующие процедуры этой системы:

- 1) процедура трансляции описания элементов TRANEL;
- 2) процедура трансляции входного описания TRAN;
- 3) процедура трансформации входного описания всей схемы и составления модели схемы COMMOD;
- 4) процедура трансляции и анализа внешних тестов TRANAN;

Принципиальная схема использованной подсистемы САСТ приведена на рис. 2.

Был создан программный интерфейс на языке ЛЯПАС-М, задачей которого являлось составление и описание схемной реализации МПА в соответствии с требованиями входного описания САСТ. В качестве элементной базы были использованы И, ИЛИ, НЕ и D-триггера. Тем же интерфейсом транслировались построенные подсистемой ПОСТРЭКС КП в язык описания тестов САСТ-а.

Процедура TRANAN использовалась в режиме проверки и анализа внешних тестов. Результаты экспериментов показали, что построенные подсистемой ПОСТРЭКС КП имеют полноту в пределах 72-96 %, причем САСТ-м были рассмотрены константные одиночные неисправности в схемной реализации МПА. Например, для МПА с 12 состояниями, 13 входными и 15 выходными каналами была построена КП длиной 64 входных набора. Из общего количества неисправностей в схеме (1180) КП обнаружила 1099, полнота КП равняется 93,1 %.

Заметим, что описанные эксперименты были проведены с КП, построенными по алгоритму Б.

Заключение. Результатами описанной серии экспериментов было показано, что подсистемой ПОСТРЭКС можно решать задачи построения КП для МПА высокой степени сложности за приемлемое для практического применения время. Также подтвердилась высокая полнота построенных КП относительно реализации МПА на базе МИС. Вопросы, связанные с эффективностью системы ПОСТРЭКС относительно других базисов реализа-

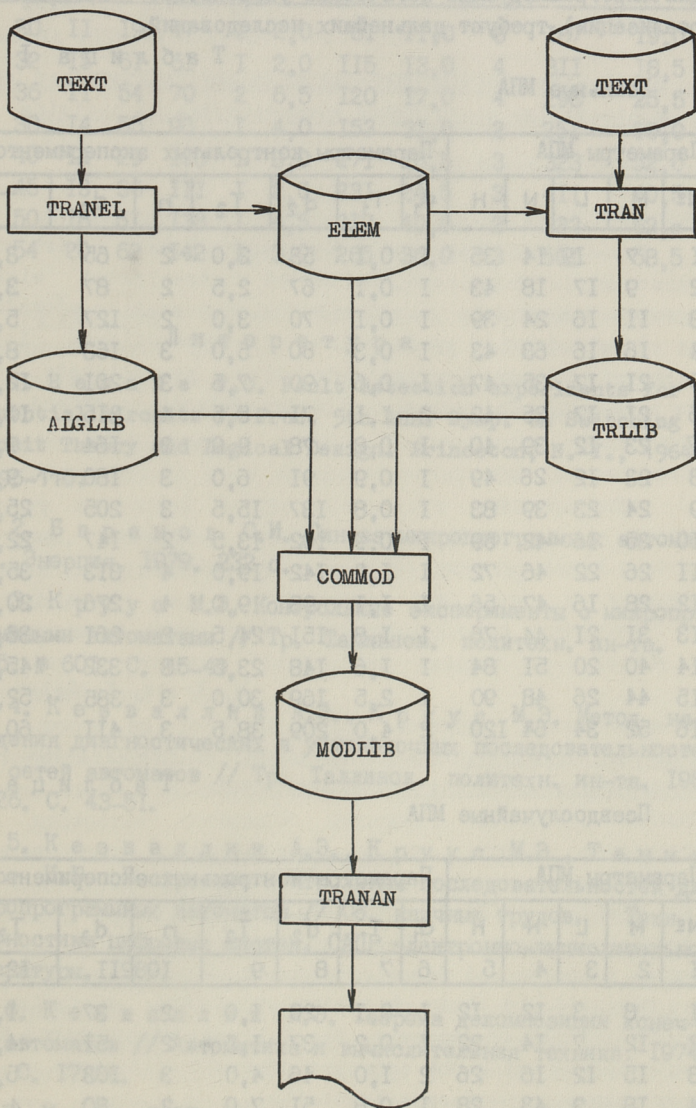


Рис. 2. Подсистема САСТ, использованная для данного эксперимента.

ции (программируемые логические устройства, программные реализации) требуют дальнейших исследований.

Т а б л и ц а 1

Реальные МПА

Параметры МПА					Параметры контрольных экспериментов						
№	M	L	N	H	d ₁	T ₁	d ₂	T ₂	n	d ₃	T ₃
1	7	19	14	35	1	0,1	53	2,0	2	65	3,0
2	9	17	18	43	1	0,1	67	2,5	2	87	3,0
3	11	16	24	39	1	0,1	70	3,0	2	127	5,0
4	16	16	63	43	1	0,3	80	5,0	3	163	8,0
5	21	17	35	47	1	0,6	90	7,5	3	201	12,0
6	21	17	35	42	2	1,1	71	5,5	4	215	11,5
7	23	12	39	40	1	0,8	78	9,0	3	164	9,0
8	23	12	26	49	1	0,9	91	6,0	3	180	9,5
9	24	23	39	83	1	0,8	137	15,5	3	205	25,0
10	25	16	42	59	1	0,9	102	13,5	2	147	22,0
11	26	22	46	72	1	1,2	142	19,0	4	313	35,5
12	28	16	47	56	1	1,1	98	9,0	4	276	20,5
13	31	21	44	76	1	1,8	151	24,5	2	261	36,5
14	40	20	51	84	1	1,6	148	23,5	3	332	45,0
15	44	26	48	90	1	2,5	169	30,0	3	385	52,5
16	52	34	64	120	2	4,0	209	38,5	3	411	60,5

Т а б л и ц а 2

Псевдослучайные МПА

Параметры МПА					Параметры контрольных экспериментов						
№	M	L	N	H	d ₁	T ₁	d ₂	T ₂	n	d ₃	T ₃
1	2	3	4	5	6	7	8	9	10	11	12
1	8	3	12	12	1	0,1	20	1,0	2	37	1,0
2	12	7	14	22	1	0,2	37	1,5	2	51	4,5
3	15	12	16	26	2	1,0	46	4,0	3	103	5,5
4	18	9	43	28	1	0,8	51	7,0	2	80	4,5
5	20	14	18	36	2	1,8	65	6,0	3	138	9,0
6	23	6	41	32	1	1,4	50	8,0	3	127	8,0
7	24	8	37	32	1	1,6	50	5,0	3	103	15,5
8	25	5	32	36	1	5,0	71	17,5	4	211	19,5

I	2	3	4	5	6	7	8	9	10	11	12
9	30	11	19	47	1	2,0	81	11,0	3	187	19,0
10	32	15	61	61	1	2,0	115	13,0	4	311	18,5
11	36	11	54	70	2	6,5	120	17,0	4	298	25,5
12	38	14	55	90	1	4,0	153	31,0	2	251	50,0
13	40	14	29	103	1	3,0	171	28,5	3	323	52,0
14	45	13	36	111	1	2,6	231	24,5	3	411	40,5
15	50	18	51	138	1	4,5	217	40,0	2	333	62,0
16	54	20	62	142	1	3,0	265	36,0	3	501	58,5

Л и т е р а т у р а

1. Н е н н и е F.C. Fault detection experiments for sequential circuits // Proc. 5th Ann. Symp. on Switching Circuit Theory and Logical Design. Princeton, N.-Y., 1964. P. 95-110.

2. Б а р а н о в С.И. Синтез микропрограммных автоматов. Энергия, 1979. 232 с.

3. К р у с М.Э. Контрольные эксперименты с микропрограммными автоматами // Тр. Таллинск. политехн. ин-та. 1985. № 601. С. 45-49.

4. К е э в а л л и к А.Э., К р у с М.Э. Метод нахождения диагностических и установочных последовательностей для сетей автоматов // Тр. Таллинск. политехн. ин-та. 1986. № 626. С. 43-51.

5. К е э в а л л и к А.Э., К р у с М.Э., Т а м м е м я з К.Ю. Построение контрольных последовательностей для микропрограммных автоматов // Сб. научных трудов. Техн. диагностика цифровых систем. САПР электронно-вычислительной аппаратуры. 1989.

6. К е э в а л л и к А.Э. Теорема декомпозиции конечных автоматов // Автоматика и вычислительная техника. 1974. № 1. С. 17-24.

7. Изделие САСТ. Описание применения. ДПХ. 00036-01 31 01 - ЛУ. 1985 г.

M. Kruus, K. Salum

Digitaalsetele juhtautomaatidele kontrolljadade
sünteesimise algoritmide eksperimentaalne uurimine

Kokkuvõte

Vaadeldakse diskreetsetele juhtseadmetele testjadade sünteesimise probleeme. On leitud kontrolljadade mikroprogrammautomaatidele ja mikroprogrammautomaatide võrkudele ning võrreldud nende karakteristikuid (kontrolljada leidmiseks kulunud aeg, kontrolljada pikkus, avastatavate vigade protsent). Esitatakse masineksperimendi tulemused.

M. Kruus, K. Salum

Experimental Study of the Algorithms for Synthesizing
Checking Sequences for Control Automata

Abstract

Problems of finding the checking sequences for control automata are discussed. The checking sequences for microprogram automata and for microprogram automata network have been designed and their characteristics (the designing time, the length of sequence, the fullness of covering) have been compared using the computer. The effectiveness of algorithms is proved experimentally.

ОПТИМИЗАЦИЯ ДЕКОМПОЗИЦИОННОГО СИНТЕЗА МИКРОПРОГРАММНЫХ
АВТОМАТОВ НА ОСНОВЕ ИНФОРЕСУРСА

I. Введение. Широкое внедрение в практику проектирования программируемых логических устройств (ПЛУ), таких как ПЛМ, микропроцессоры и др., породило усиленный интерес к этапу логического проектирования дискретных управляющих устройств. Высокие требования к срокам и надежности этапа логического проектирования, а также сложность возникающих на этом этапе задач влекут за собой необходимость его автоматизации. С другой стороны, специфика базиса ПЛУ проявляется в необходимости применения декомпозиционных методов проектирования в этом базисе. Необходимость эта обусловлена тем, что в корпусе одной интегральной схемы может быть реализовано устройство, параметры которого ограничены (количество входных и выходных полюсов, быстродействие и т.д.).

Одним из наиболее общих методов декомпозиции конечных автоматов является парный метод декомпозиции [1], при котором декомпозиция строится на основе т.н. полного множества разбиений на множестве состояний исходного автомата. Парный метод декомпозиции распространен и на модель микропрограммного автомата (МПА), которая позволяет компактно описывать управляющие автоматы большой размерности. В работах [2-3] предлагаются методы выбора разбиений для декомпозиции автоматов с различными ограничениями на структуру и базис получаемых в результате декомпозиции сетей автоматов. Анализ показывает, что наиболее критичным ограничением при реализации автоматов на ПЛУ является ограничение на количество входов. В то же время учет этого и многих других параметров, определяющих сложность получаемых сетей, на этапе выбора разбиений затруднен. Для сравнения между собой разбиений оказывается недостаточно отношения частичного порядка $\leq$, так как обычно приходится выбирать между разбиениями,

которые несравнимы между собой. В настоящей работе для решения задачи оптимизации декомпозиционного синтеза автоматов на ПЛУ и оценки информационных свойств разбиений предлагается использовать понятие инфоресурса [4,5,6] в качестве дополнительной размерности в теории декомпозиции.

2. Основные понятия и постановка задачи

Приведем сначала определения, необходимые для постановки задачи декомпозиции МПА.

Определение 1. Под МПА будем понимать систему $A=(X,S,Y,\delta,\lambda)$, где $X=\{x_1, \dots, x_1\}$ - множество входных двоичных переменных (каналов), из которых МПА получает входные наборы z - элементы булева пространства $\Phi(X)$. Здесь и ниже под булевым пространством $\Phi(X)$, порождаемым множеством двоичных переменных $X=\{x_1, \dots, x_1\}$, будем понимать множество $\{0,1\}^{|X|}$; $|X|$ - мощность множества X .

$S=\{s_1 | 1=1, \dots, m\}$ - множество внутренних состояний;

$Y=\{y_1, \dots, y_n\}$ - множество выходных двоичных переменных (каналов), на которые МПА выдает выходные наборы w - элементы булева пространства $\Phi(Y)$;

$\delta: D(\delta) \rightarrow S$ - функция переходов, где $D(\delta) \subseteq S \times \Phi(X)$ - область определения функции δ ;

$\lambda: D(\lambda) \rightarrow \Phi(Y)$ - функция выходов, где $D(\lambda) \subseteq S$ или $D(\lambda) \subseteq S \times \Phi(X)$ - область определения функции λ для моделей Мура или Мили, соответственно.

В настоящей работе в интересах краткости изложения ограничимся рассмотрением МПА Мура.

Центральным понятием при постановке задач декомпозиции автоматов является понятие сети автоматов [1]. В [1] показано, что рассмотрение полуавтоматов в качестве компонент сети не приводит, как правило, к потере общности модели. Поэтому и в данной работе взято за основу соответствующее определение сети МПА [2].

Определение 2. Сетью МПА называется система $N=(X,B,Y,F,g)$, где

1. $X=\{x_1, \dots, x_1\}$ - множество входных двоичных переменных сети.

2. $B=\{A_1 | 1 \in J=\{1, \dots, k\}\}$ - множество компонентных автоматов сети (базис сети). Здесь $A_1=(X_1, S_1, \delta_1)$ - микропрограммный полуавтомат (МПА), у которого $X_1=X'_1 \cup X''_1$; $X'_1 \subseteq X$. X'_1 и X''_1 называются, соответственно, множествами внешних и внутренних входов A_1 .

3. $Y = \{y_1, \dots, y_n\}$ - множество выходных двоичных переменных сети.

4. $F = \{f_1 | 1 \in J\}$ - множество функций соединения (структура сети):

$$f_1: \times_{j \in J_1} S_j \rightarrow \Phi(X_1^n); J_1 \subseteq J.$$

5. $g: \times_{j \in J_g} S_j \rightarrow \Phi(Y), J_g \subseteq J$ - функция выхода сети.

Графически сеть МПА N изображена на рис. I.

Определение 3. Проекцией элемента $z \in \Phi(X)$ на $\Phi(X')$, $X' \subseteq X$, называется элемент $z' \in \Phi(X')$, значения координат которого в $\Phi(X')$ совпадают с координатами X' из z . Обозначаем $z' = \text{pr}(z, X')$.

Пример. Если $X = \{x_1, x_2, x_3, x_4\}$, а $X' = \{x_2, x_3\}$, то $\text{pr}(0101, X') = 10$.

Определение 4. Результирующим автоматом сети N называется автомат $A_N = (X_N, S_N, Y_N, \delta_N, \lambda_N)$, у которого :

$$1. X_N = X; Y_N = Y.$$

$$2. S_N = \times_{1 \in J} S_1.$$

$$3. \delta_N: S_N \times \Phi(X_N) \rightarrow S_N, \text{ где}$$

$$\delta_N((s^1, \dots, s^k), z) = (\delta_1(s^1, z_1' \times z_1^n), \dots, \delta_k(s^k, z_k' \times z_k^n)),$$

причем $z \in \Phi(X); s^1 \in S_1; z_1' = \text{pr}(z, X_1'); z_1^n = f_1(\{s^j | j \in J_1\}) (1 \in J)$.

$$4. \lambda_N: S_N \rightarrow \Phi(Y), \text{ где}$$

$$\lambda_N(s^1, \dots, s^k) = g(\{s^1 | 1 \in J_g\})$$

Определение 5. МПА $A' = (X', S', Y', \delta', \lambda')$ называется подавтоматом МПА $A = (X, S, Y, \delta, \lambda)$, если и только если $S' \subseteq S; X' \subseteq X; Y' \subseteq Y$ и для всех $s \in S'$ и $z' \in \Phi(X')$ имеет место

$$(z \in \Phi(X) \ \& \ z' = \text{pr}(z, X')) \Rightarrow \delta'(s, z') = \delta(s, z) \ \& \ \lambda'(s) = \text{pr}(\lambda(s), Y').$$

Определение 6. Если имеется биекция $f: X \leftrightarrow X'$, где X и X' множества булевых переменных, то под $f^\square$ будем понимать функцию

$$f^\square: \Phi(X) \rightarrow \Phi(X') \text{ такую, что}$$

$$z' = f^\square(z) \Leftrightarrow \forall x_1 \in X [\text{pr}(z, \{x_1\}) = \text{pr}(z', \{f(x_1)\})]$$

Определение 7. МПА $A = (X, S, Y, \delta, \lambda)$ и $A' = (X', S', Y', \delta', \lambda')$ называются изоморфными, если и только если существуют три биекции:

$$\phi_1: S \leftrightarrow S'; \phi_2: X \leftrightarrow X'; \phi_3: Y \leftrightarrow Y'$$

таких, что для любых $s \in S$ и $z \in \Phi(X)$ имеет место

$$\phi_1(\delta(s, z)) = \delta'(\phi_1(s), \phi_2^\square(z));$$

$$\phi_3^\square(\lambda(s)) = \lambda'(\phi_1(s)).$$

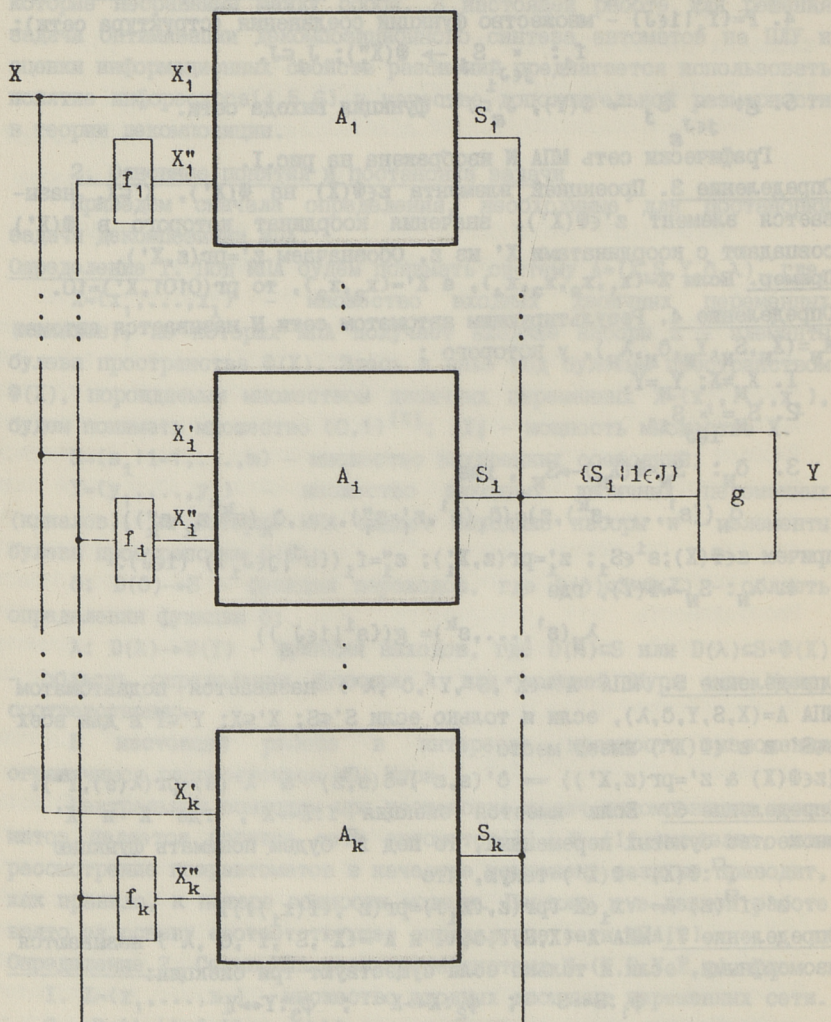


Рис.1. Сеть МПА.

Определение 8. Под задачей декомпозиции МПА А будем понимать построение сети N, результирующий автомат которой имеет подавтомат, изоморфный автомату А.

В [2] приведен конструктивный метод декомпозиции МПА $A=(X,S,Y,\delta,\lambda)$ в сеть МПА N по полному множеству разбиений $\{\pi_i | i \in J\}$ на множестве внутренних состояний S. Полным называется множество $\{\pi_i | i \in J\}$, состоящее из разбиений, произведение которых равно нулевому (поэлементному) разбиению. Основные этапы метода декомпозиции [2], лежащего и в основе нашего подхода, следующие.

Шаг 1. Выбираем множества входов и выходов сети равными множествам входов X и выходов Y автомата А, соответственно.

Шаг 2. Образует множество функций $\{F_i | i \in J\}$:

$$F_i: S \times \Psi(X) \rightarrow \pi_i, \text{ где } F_i(s_p, z) = \pi_i[\delta(s_p, z)1].$$

Здесь и в дальнейшем под $\pi_i[x]$ понимается блок разбиения π_i , содержащий элемент x.

Шаг 3. Образует множество разбиений $\{\phi_i | i \in J\}$ на множестве S в соответствии со следующим выражением:

$$s_p \equiv s_q(\phi_i) \iff \forall z \in \Psi(X) [F_i(s_p, z) = F_i(s_q, z)1], \quad s_p, s_q \in S.$$

Здесь и далее запись $s_p \equiv s_q(\phi_i)$ означает, что элементы s_p и s_q принадлежат одному и тому же блоку разбиения ϕ_i .

Шаг 4. Найдем минимальные множества внешних входов, необходимых для A_i , — $XMIN(A, \pi_i) \subseteq X$ ($i \in J$). При этом $x_j \in X \setminus XMIN(A, \pi_i)$, если и только если

$\forall s_p \in S \forall z \in \Psi(X) [F_i(s_p, z) = F_i(s_p, z'_j)1]$. Здесь z'_j — двоичный вектор, j-я координата которого инверсна j-й координате z, а остальные координаты z и z'_j совпадают. Принимаем $X'_i = XMIN(A, \pi_i)$.

Шаг 5. Для каждого $i \in J$ выбираем множество двухблочных разбиений $Q_i = \{\rho_i^h | h=1, \dots, k_i\}$ на множестве S. Пусть $\{\pi_j | j \in J'_i \subseteq J\}$ — множество разбиений такое, что $\prod_{j \in J'_i} \pi_j \leq \phi_i$. Тогда определим $\{\rho_i^h\}$ так, что

$$\prod_{h=1}^{k_i} \rho_i^h = \prod_{j \in J'_i} \pi_j, \quad \text{где } J'_i = J'_i \setminus \{i\} \text{ определяет множество индексов}$$

компонентных автоматов сети, дающих информацию компонентному автомату A_i . Если $i \in J'_i$, то A_i имеет внутреннюю обратную связь, в противном случае нет. Если закодировать один блок разбиения ρ_i^h нулем и другой единицей, то фактически ρ_i^h , $h=1, \dots, k_i$, определяют

внутренние входные двоичные переменные для A_1 , т.е. $X_1''=Q_1$ и $\Psi(X_1'') = \bigtimes_{h=1}^{k_1} \rho_1^h$;

Шаг 6. Определяем базис сети:

а) Принимаем $S_1 = \pi_1$;

б) Определяем входной алфавит A_1 как $X_1 = X_1' \cup X_1''$;

в) Определяем функцию переходов A_1 :

$$\delta_1: \pi_1 \times (\Psi(X_1') \times (\bigtimes_n \rho_1^h)) \rightarrow \pi_1,$$

$$\delta_1(B^1, (z', \{B_1^h\})) = \begin{cases} \pi_1[\delta(B^1 \cap u_1, z)], & \text{если } B^1 \cap u_1 \neq \emptyset; \\ \text{не определено} & \text{если } B^1 \cap u_1 = \emptyset; \end{cases}$$

Здесь $B^1 \in \pi_1$; $z \in \Psi(X)$; $z' = \text{pr}(z, X_1')$; $u_1 = \bigcap_n B_1^h$; $B_1^h \in \rho_1^h$.

Шаг 7. Определяем структуру сети:

$$f_1: \bigtimes_{j \in J_1} S_j \rightarrow \Psi(X_1''), \text{ где}$$

$$f_1(\{B^j | j \in J_1\}) = \begin{cases} \{\rho_1^h[u_1]\}, & \text{если } u_1 \neq \emptyset; \\ \{\rho_1^h[u_1]\}, & \text{если } u_1 = \emptyset. \end{cases}$$

Здесь $B^j \in S_j$, $u_1 = \bigcap_{j \in J_1} B^j$.

Шаг 8. Определяем функцию выхода сети.

$$g: \bigtimes_{j \in J_g} S_j \rightarrow \Psi(Y); J_g = J;$$

$$g(\{B^j | j \in J_g\}) = \begin{cases} \lambda(u), & \text{если } u \neq \emptyset; \\ \text{не определено} & \text{в противном случае.} \end{cases}$$

Здесь $B^j \in S_j$; $u = \bigcap_{j \in J_g} B^j$.

Приведенный конструктивный метод декомпозиции МПА является общим в том смысле, что в нем не заданы какие-либо ограничения ни на базис, ни на структуру сети. Поэтому он может служить основой для решения широкого круга самых различных задач декомпозиционного синтеза МПА.

При построении сетевых реализаций МПА на базе ПЛУ возникают, однако, специфические проблемы, требующие на наш взгляд как определенной модификации понятия сети МПА, так и четкого

упорядочения приоритетов при решении возникающих многокритериальных оптимизационных задач. Во-первых, как уже говорилось выше, при выборе разбиений для декомпозиции необходим в первую очередь учет ограничения на количество входов ПЛУ. Во-вторых, в случае ПЛУ, особенно – ПЛМ с внутренней памятью или микропроцессоров, представляет интерес решение задачи декомпозиции в модифицированном виде, где компонентами получаемой сети являлись бы не полуавтоматы, а МПА Мура. Это позволило бы более гибко оперировать с ресурсом ПЛУ и получить дополнительные возможности для упрощения функций соединения f_i и функции выхода сети g за счет "перекачки" сложности между ними и функциями выходов λ_i компонентных автоматов (см. рис.2). Фактически речь идет здесь о том, что после удовлетворения основного ограничения по количеству входов ПЛУ, возможно дополнительное упрощение сети за счет свободного ресурса ПЛУ.

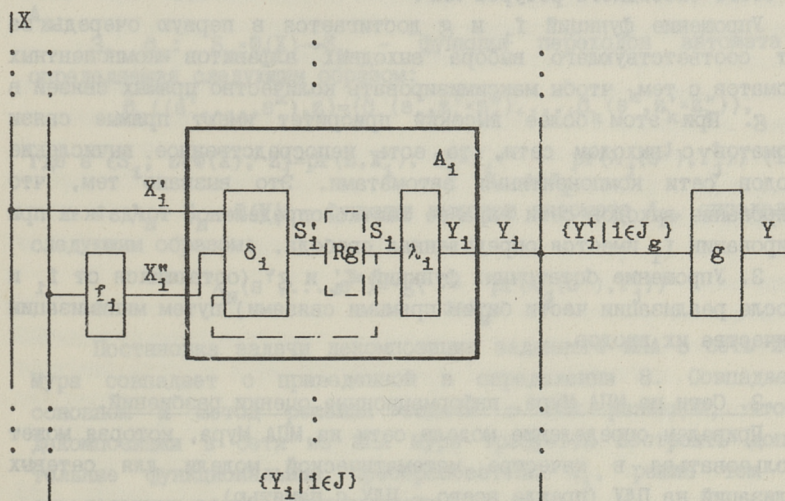


Рис.2. Фрагмент сети из МПА Мура.

При построении сети МПА Мура в ходе декомпозиции заданного МПА возникают, таким образом, сложные задачи оптимизации,

связанные с совместным кодированием выходов функций λ_1 и внутренних входных алфавитов X_1'' (выходов функции f_1). Отметим, что процедура декомпозиции МПА в сеть из МПА Мура совпадает, в общих чертах, с приведенным выше конструктивным методом декомпозиции, основные отличия связаны с реализацией в компонентных автоматах A_1 дополнительных функциональных преобразователей λ_1 .

Постановка задачи оптимизации декомпозиционного синтеза МПА в базисе ПЛУ опирается на учет как технических характеристик ПЛУ, так и своеобразия модели МПА и включает в себя следующую последовательность приоритетов:

1. Учет ограничения на количество входов компонентных автоматов с минимизацией числа компонентных автоматов;

2. Перекачка сложности из f_1 и g в λ_1 с использованием для этого свободного ресурса ПЛУ.

Упрощение функций f_1 и g достигается в первую очередь за счет соответствующего выбора выходных алфавитов компонентных автоматов с тем, чтобы максимизировать количество прямых связей в f_1 и g . При этом более высокий приоритет имеют прямые связи автоматов с выходом сети, то есть непосредственное вычисление выходов сети компонентными автоматами. Это вызвано тем, что кодирование выходов сети заранее жестко определено, тогда как при кодировании f_1 имеется определенная свобода.

3. Упрощение остаточных функций f_1' и g' (остающихся от f_1 и g после реализации части битов прямыми связями) путем минимизации количества их входов.

3. Сети из МПА Мура, информационные оценки разбиений.

Приведем определение модели сети из МПА Мура, которая может использоваться в качестве математической модели для сетевых реализаций на ПЛУ (прежде всего - ПЛУ с памятью).

Определение 9. Под сетью из МПА Мура будем понимать систему

$$N = (X, B, Y, F, g), \text{ где}$$

1. X - множество входов сети.

2. $B = \{A_j \mid j \in J = \{1, \dots, k\}\}$ - множество компонентных МПА (базис сети); $A_j = (S_j, X_j' \cup X_j'', Y_j, \delta_j, \lambda_j)$.

Здесь $X_j' \subseteq X$ и X_j'' - соответственно, множества внешних и

внутренних входов МПА A_1 , а Y_1 — множество его выходов.

3. Y — множество выходов сети.

4. $F = \{f_1 | 1 \in J\}$ — множество функций соединения (структура сети):

$$f_1: \times_{j \in J_1} \Psi(Y_j^1) \rightarrow \Psi(X_1); J_1 \subseteq J; Y_j^1 \subseteq Y_j.$$

5. $g: \times_{j \in J_g} \Psi(Y_j^+) \rightarrow \Psi(Y)$ — функция выхода сети; $J_g \subseteq J; Y_j^+ \subseteq Y_j$.

Определение 10. Под результирующим автоматом A_N сети из МПА Мура $N = \{\{A_1\}, X, Y, \{f_1\}, g\}$ понимается микропрограммный автомат

$$A_N = (S_N, X, Y, \delta_N, \lambda_N), \text{ где}$$

1. X, Y — множества входных и выходных переменных автомата A_N , совпадающие с множествами входных и выходных переменных сети N , соответственно.

2. $S_N = \times_{1 \in J} S_1$ — множество состояний результирующего автомата A_N .

3. $\delta_N: S_N \times \Psi(X) \rightarrow S_N$ — функция переходов автомата A_N , определяемая следующим образом:

$$\delta_N((s^1, \dots, s^k), z) = (\delta_1(s^1, z_1' \times z_1''), \dots, \delta_k(s^k, z_k' \times z_k'')),$$

где $s^1 \in S_1; z \in \Psi(X); z_1' = \text{pr}(z, X_1^1); z_1'' = f_1(\times_{j \in J_1} \text{pr}(\lambda_j(s^j), Y_j^1)) (1 \in J)$.

4. $\lambda_N: S_N \rightarrow \Psi(Y)$ — функция выходов автомата A_N , определяемая следующим образом:

$$\lambda_N(s^1, \dots, s^k) = g(\times_{j \in J_g} \text{pr}(\lambda_j(s^j), Y_j^+))$$

Постановка задачи декомпозиции заданного МПА в сеть из МПА Мура совпадает с приведенной в определении 8. Совпадает, в основном и метод решения задачи с той разницей, что при декомпозиции в сети из МПА Мура требуется построить дополнительные функциональные преобразователи λ_1 , решив тем самым определенную задачу факторизации f_1 и g .

Задача декомпозиции в сеть из МПА Мура разбивается также на два основных этапа. На первом этапе по исходному МПА должно быть синтезировано полное множество разбиений, обеспечивающее выполнение основных ограничений, накладываемых на базис и структуру сети (количество входов у компонентных автоматов, число

компонентных автоматов и т.д.). Задачу нахождения полной системы разбиений для декомпозиции можно рассматривать как некоторую задачу глобальной трансформации, которая, как правило, решается приближенно.

После выбора множества декомпозиционных разбиений встают задачи оптимального кодирования, выбора оптимальных входных и выходных алфавитов компонентных автоматов с целью минимизации как компонентных автоматов, так и функций соединения и выхода сети. Эти задачи решаются через последовательность локальных трансформаций – последовательного кодирования отдельных входных и выходных алфавитов компонентных МПА. Оптимизация в ходе этих локальных трансформаций осуществляется по локальным критериям точно или почти точно.

Применяемый нами конструктивный метод декомпозиции сводится, в основном, к различным операциям с разбиениями. При этом все основные параметры декомпозиции определяются путем сравнения разбиений, которые, как известно, образуют решетку [1]. Однако отношение частичного порядка в решетке разбиений осложняет решение многих практических задач декомпозиции, так как именно несравнимые между собой разбиения определяют характер и свойства декомпозиции. Поэтому были предложены количественные информационные оценки для разбиений [3,4], позволяющие осуществить более тонкое сравнение разбиений, вследствие чего улучшается и качество декомпозиции.

В данной работе для оценки и сравнения разбиений с точки зрения содержащейся в них информации мы используем понятие инфоресурса [1]. Инфоресурс $I(\pi_2/\pi_1)$ разбиения π_2 относительно разбиения π_1 равен

$$I(\pi_2/\pi_1) \stackrel{\text{Def}}{=} \log_2 \left(\min_{\pi_1 \cdot \pi_3 \leq \pi_2} |\pi_3| \right).$$

Значение $|I(\pi_2/\pi_1)|$ равно минимальному количеству двоичных каналов, которое требуется для передачи информации, недостающей в разбиении π_1 до разбиения π_2 (здесь и ниже $|x|$ – минимальное целое, большее или равное x). В [4] показано, что

$$I(\pi_2/\pi_1) = \log_2 \left(\max_{B_1 \in \pi_1} k_1 \right),$$

где k_1 - количество блоков π_2 , пересечение которых с блоком B_1 разбиения π_1 не пусто.

4. Построение системы разбиений для декомпозиции на основе информресурса.

При построении полных систем разбиений для декомпозиции будем исходить из того, что задан микропрограммный автомат $A=(S, X, Y, \delta, \lambda)$. Пусть требуется построить декомпозицию МПА A в сеть N таким образом, чтобы количество двоичных входов у ее компонентных автоматов A_1 не превышало L^* , т.е.

$$\|X_1'\| + \|X_1''\| \leq L^* \quad (1 \in J),$$

где X_1' - множество внешних входов сети, от которых зависит работа i -го компонентного МПА; X_1'' - множество внутренних входов i -го компонентного МПА.

В дальнейшем будем различать реализации МПА на ПЛУ, требующих дополнительного ресурса внешних входов для передачи автомату информации о его состоянии (например, реализация на ПЛМ без памяти), и не требующих такого ресурса (например, реализация на ПЛМ с памятью или на однокристалльных микро-ЭВМ). Эти реализации назовем, соответственно, реализациями из компонент без локальной обратной связи и с локальной обратной связью. Необходимость в декомпозиции с ограничением на количество входов компонентных автоматов наступает в случаях, когда $L^* < \|X\|$ (для компонент с локальной обратной связью) или $L^* < \|X\| + \log_2 |S|$ (для компонент без локальной обратной связи).

X_1' минимальной мощности, как уже говорилось выше, равно $X_{\min}(A, \pi_1)$, а минимальная мощность X_1'' различна для реализаций на ПЛУ с локальной обратной связью и без нее. Для ПЛУ с локальной обратной связью $\|X_1'\|$ определяется количеством бит, необходимым для передачи информации, недостающей в разбиении π_1 (передаваемом по локальной обратной связи) до разбиения θ_1 (θ_1 определяется в ходе конструктивного метода). Это количество равно $|I(\theta_1/\pi_1)|$. Для случая ПЛУ без локальной обратной связи разбиение π_1 непосредственно недоступно компонентному автомату и минимальное значение $\|X_1'\|$ равна $|I(\theta_1/1_\pi)| = \lceil \log_2 \|\theta_1\| \rceil$ (здесь 1_π - единичное разбиение). Таким образом, получаем условия для выбора разбиений с ограничением на входы компонентных МПА для случая ПЛУ с

локальной обратной связью и без нее соответственно:

$$\|XMIN(A, \pi_1)\| + lI(\phi_1/\pi_1) \leq L^* \quad (1)$$

и

$$\|XMIN(A, \pi_1)\| + l\log_2 \|\phi_1\| \leq L^*. \quad (2)$$

При построении полных систем разбиений мы будем стремиться получить эти системы из минимально возможного количества разбиений, удовлетворяющих приведенным условиям. Тем самым минимизируется количество компонентных автоматов сети. Имеющаяся свобода при выборе разбиений используется для увеличения количества выходов сети, вычисляемых непосредственно ее компонентными автоматами.

Для уменьшения перебора используется эвристический алгоритм, распадающийся на три этапа.

На первом этапе генерации разбиений находятся первичные α -разбиения. Первичным разбиением α_1 называется минимальное разбиение на множестве S такое, что соответствующий ему компонентный автомат не зависит от i -го двоичного входа. α_1 определяется следующим выражением[5]:

$$s_p = s_q(\alpha_1) \iff \exists s_r \in S \exists z \in \Psi(X) [s_p = \delta(s_r, z) \& s_q = \delta(s_r, z'_1)],$$

где z'_1 - двоичный вектор, совпадающий с z по всем компонентам, кроме i -ой.

На втором этапе генерируются суммы π_1 из первичных α -разбиений, удовлетворяющие условию I (2), образующие полное множество разбиений $P = \{\pi_1 | i \in J\}$. На третьем этапе найденные разбиения укрупняются таким образом, чтобы оставались выполненными условие I (2) и условие полноты множества разбиений и при этом не уменьшалось множество выходов, вычисляемых непосредственно компонентными автоматами сети.

Основные идеи второго этапа, на котором производится генерация полного множества α -разбиений, следующие.

I. На каждом шаге выбирается одно очередное разбиение π_1 , пополняющее вначале пустое множество P . Разбиение π_1 , с одной стороны, должно удовлетворять условию I (2), а с другой - максимально приблизить множество P к полному (для оценки близости

множества P к полному используется значение $I(0_\pi / \bigcap_{\pi_1 \in P} \pi_1)$.

2. Выбор разбиения π_1 , в свою очередь, распадается на ряд шагов. На каждом шаге вначале нулевое разбиение π_1 последовательно увеличивается (сложением с некоторым первичным) с целью удовлетворения условия I5 (I6). Из различных вариантов увеличения π_1 , эквивалентных по количеству входов, выбирается наиболее предпочтительный с точки зрения полноты множества P .

Приведем результаты синтеза разбиений, полученные для МПА, изображенного на рис.3, при $L^*=4$ для компонент с локальной обратной связью следующие результаты:

Первичные α -разбиения:

$$\alpha_1 = \underline{1,3;2,6;4,5;7;8;9};$$

$$\alpha_2 = \underline{1,2,4;3,5,6,7,8,9};$$

$$\alpha_3 = \underline{1,3;2,6;4,5;7;8;9};$$

$$\alpha_4 = \underline{1;2;3,5;4;6;7;8;9};$$

$$\alpha_5 = \underline{1;2,4;3;5,6;7;8;9};$$

Полное множество разбиений $P=\{\pi_1, \pi_2\}$:

$$\pi_1 = \alpha_1 + \alpha_3 = \underline{1,3;2,6;4,5;7;8;9}; Y'_1 = \{y_1, y_2, y_3, y_4\};$$

$$X'_1 = \{x_2, x_4, x_5\};$$

$$\pi_2 = \alpha_4 + \alpha_5 = \underline{1;2,4;3,5,6;7;8;9}; Y'_2 = \{y_1, y_2\};$$

$$X'_2 = \{x_1, x_2, x_3\};$$

$$Y' = Y'_1 \cup Y'_2 = \{y_1, y_2, y_3, y_4\}.$$

Полное множество разбиений $P=\{\pi_1, \pi_2\}$ после укрупнения :

$$\pi_1 = \underline{1,3;2,6;4,5;7,9;8}; Y'_1 = \{y_2, y_3, y_4\};$$

$$\pi_2 = \underline{1;2,4;3,5,6;7,8;9}; Y'_2 = \{y_1\}.$$

$$Y' = Y'_1 \cup Y'_2 = \{y_1, y_2, y_3, y_4\}.$$

$$M(\pi_1) = \underline{1,7,8;2;3;4;5;6;9}; M(\pi_2) = \underline{1;2;3,6,9;4;5;7,8};$$

$$I(\theta_1/\pi_1) = I(\theta_2/\pi_2) = 1.$$

$$\|X'_1\| + I(\theta_1/\pi_1) = 3 + 1 = 4 = L^*.$$

$$\|X'_2\| + I(\theta_2/\pi_2) = 3 + 1 = 4 = L^*.$$

Кодирование функций соединения и выхода, оптимизация сети.

После того, как завершен выбор системы разбиений для декомпозиции, фактически решена задача глобальной трансформации. При синтезе сети из МПА Мура мы опираемся на приведенный выше

s_p	s_q	$X(s_p, s_q)$	$Y(s_p, s_q)$
1	2	$x_1 x_3$	-
	6	$\bar{x}_1 x_3$	
	6	$\bar{x}_3$	
2	2	$x_1 x_2 x_3 x_5$	y_3
	4	$x_1 x_2 x_3 \bar{x}_5$	
	1	$x_1 \bar{x}_2 x_3$	
	3	$\bar{x}_1 \bar{x}_2 x_3$	
	5	$\bar{x}_1 x_2 x_3 \bar{x}_5$	
	6	$\bar{x}_1 x_2 x_3 x_5$	
	6	$x_2 \bar{x}_3 x_5$	
	5	$x_2 \bar{x}_3 \bar{x}_5$	
	3	$\bar{x}_2 \bar{x}_3$	
3	3	$x_2 x_4$	-
	5	$x_2 \bar{x}_4$	
	7	$\bar{x}_2$	
4	1	$x_1 x_3$	$y_3 y_4$
	3	$\bar{x}_1 x_3$	
	3	$\bar{x}_3$	
5	6	x_2	$y_3 y_4$
	9	$\bar{x}_2$	
6	6	$x_2 x_5$	y_3
	5	$x_2 \bar{x}_5$	
	8	$\bar{x}_2$	
7	2	1	y_1
8	2	1	$y_1 y_2$
9	6	$x_2 x_5$	y_3
	5	$x_2 \bar{x}_5$	
	7	$\bar{x}_2$	

Рис.3. Исходный МПА.

конструктивный метод декомпозиции. Однако, при этом шаги 5-8 конструктивного метода принимают следующий вид:

Шаг 5. Для каждого $i \in J$ выбираем множества двухблочных разбиений $P_i = \{\sigma_i^v\}$ и $Q_i = \{\rho_i^h\}$ на множестве S таких, что :

$$(1) \forall \sigma_i^v \sigma_i^v \geq \pi_i$$

$$(2) \begin{cases} \pi_i \cdot \prod_{\rho_i^h \in Q_i} \rho_i^h \leq \theta_i - \text{для компонент с локальной} \\ \text{обратной связью;} \\ \prod_{\rho_i^h \in Q_i} \rho_i^h \leq \theta_i - \text{для компонент без локальной} \\ \text{обратной связи;} \end{cases}$$

$$(3) \prod_{y_w \in Y} \pi \langle y_w \rangle \cdot \left[\prod_{i \in J} \prod_{\rho_i^h \in Q_i} \rho_i^h \right] \leq \prod_{i \in J} \prod_{\sigma_i^v \in P_i} \sigma_i^v,$$

где $\pi \langle y_w \rangle$ - двухблочное разбиение, первый блок которого включает в себя те состояния $s_p \in S$, для которых $\text{pr}(\lambda(s_p), \{y_w\}) = 0$, а второй блок - те состояния, для которых бит y_w выходной функции равен 1.

Шаг 6. Базис сети определяется аналогично шагу 6 приведенного выше конструктивного метода. Добавляются следующие действия:

г). Принимаем множество выходных двоичных переменных компонентного автомата Y_i равным множеству двухблочных разбиений P_i ($\Psi(Y_i)$ соответствует $\times \sigma_i^v$).

д). Определяем выходную функцию МПА A_i :

$$\lambda_i: \pi_i \rightarrow \times \sigma_i^v, \quad \lambda_i(B^1) = \{B_i^v\},$$

где B_i^v - блок σ_i^v , для которого $B^1 \subseteq B_i^v$.

Шаг 7. Определяем структуру сети:

$$f_i: \times_{j \in J_i} \sigma_j^v \times_{j \in P_j^1} \sigma_j^v \rightarrow \times_n \rho_i^h, \text{ где } J_i \subseteq J; P_j^1 \subseteq P_j.$$

$$f_i(\{B_j^v \in \sigma_j^v\}) = \begin{cases} \{\rho_i^h[u_1]\}, & \text{если } u_1 \neq \emptyset; \\ \text{не определено,} & \text{если } u_1 = \emptyset; \end{cases}$$

Здесь $u_1 = \cap B_j^v$.

Шаг 8. Вычисляем функцию выхода сети.

$$g: \times_{j \in J_g} \sigma_j^v \times_{j \in P_j^+} \sigma_j^v \rightarrow \Psi(Y); J_g = J; P_j^+ \subseteq P_j.$$

$$g(\{B_j^v \in \sigma_j^v\}) = \begin{cases} \lambda(u), & \text{если } u \neq \emptyset; \\ \text{не определено} & - \text{ в противном случае.} \end{cases}$$

Здесь $u = \cap B_j^v$.

На шаге 4 решается задача локальной трансформации - кодирование функций выхода и соединения компонентных автоматов. Это кодирование сводится, по существу, к выбору множеств двухблочных разбиений $P_1 = \{\sigma_1^v\}$ и $Q_1 = \{\rho_1^h\}$ ($i \in J$), где каждое $\sigma_1^v \in P_1$ соответствует двоичному выходу $y_v \in Y_1$ i -го компонентного МПА и каждое $\rho_1^h \in Q_1$ соответствует двоичному входу $x_h \in X_1^n$ этого компонентного автомата. При выборе этих разбиений одновременно решаются задачи выбора множеств $J_g \subseteq J$ и $J_f \subseteq J$ номеров компонентных автоматов, выходы которых используются соответственно для вычисления выходной функции g и функций соединения f_i ($i \in J$). Выбираются также P_j^+ и P_j^- - множества двухблочных разбиений, соответствующие выходам, поступающим от j -го автомата на g и f_i . Основные идеи алгоритма локальных трансформаций следующие. В соответствии с выбранным порядком приоритетов вначале выбираются такие выходы сети $y_w \in Y$, которые могут быть непосредственно вычислены компонентными автоматами (для которых $\pi(y_w) \geq \pi_j$ для некоторого $j \in J$). Остальные биты образуют остаточную выходную функцию g' .

Следующим этапом является генерация прямых связей между выходами компонентных автоматов и их внутренними входами. При этом необходимо учитывать, что суммарное количество двоичных входов у компонентного автомата не должно превышать L^* . Для учета этого фактора заметим следующее. Пусть i -й компонентный автомат имеет $|X_1'|$ внешних входов и в качестве прямых связей для него уже выбрано $q-1$ двухблочных разбиений $\{\rho_1^h | h=1, \dots, q-1\}$. Тогда при выборе для связи с автоматом A_1 разбиения $\sigma_j^v \geq \pi_j$, это разбиение может быть использовано для прямой связи только в том случае, если число бит, требуемое для передачи недостающей до ϕ_1 информации, не превышает свободного ресурса входов. В терминах инфоресурса это условие записывается в виде:

$$I_q = I(\pi''/\pi' \cdot \sigma_j^v) \leq L^* - |X_1'| - q, \quad (3)$$

где $\pi'' = \sigma_1$; $\pi' = \begin{cases} \pi_1 \cdot \prod_{h=1}^{q-1} \rho_1^h & \text{для компонент с локальной} \\ & \text{обратной связью;} \\ \prod_{h=1}^{q-1} \rho_1^h & \text{для компонент без локальной обратной связи.} \end{cases}$

При этом должно выбираться $\sigma_j^v \geq \pi_j$, минимизирующее значение $I_q = I(\pi''/\pi' \cdot \sigma_j^v)$. Эвристический алгоритм выбора такого двухблочного разбиения $\sigma_j^v \geq \pi_j$, минимизирующего $I(\pi''/\pi' \cdot \sigma_j^v)$ при заданных π' , π'' и π_j описан в [1]. Таким образом, для каждого A_1 производится последовательность локальных трансформаций, при которых выбираются двухблочные разбиения, пополняющие как множество Q_1 , так и множества F_j для автоматов, связываемых с A_1 прямыми связями. Эта процедура для каждого A_1 заканчивается на некотором шаге одним из двух событий: или $I_q = 0$, или условие (3) не выполнено. При первом исходе построение внутреннего алфавита для A_1 закончено, а при втором исходе должна быть реализована остаточная функция f'_1 .

Генерация двухблочных разбиений - аргументов остаточной функции выхода g' и остаточных функций соединения f'_1 проходит в основном аналогично генерации разбиений для прямых связей f_1 . Минимизация числа аргументов остаточных функций производится эвристически путем выбора на каждом шаге двухблочного разбиения для связи σ_j^v , минимизирующего остаточный инфоресурс. После того, как аргументы функций f' выбраны, кодируются значения этих функций..

Ниже приводится алгоритм, производящий локальные трансформации по кодированию выходов и внутренних входов компонентных автоматов.

Шаг 1. $P_1 := \emptyset$; $Q_1 := \emptyset$; $Q_1^0 := \emptyset$; $P_j^1 := \emptyset$; $P_j^+ := \emptyset$; $(1, j \in J)$.

$P_g := \{\pi \langle u_w \rangle \mid u_w \in Y\}$. $w := 0$.

Шаг 2. $w := w + 1$. Если $w > |Y|$, то идти к 5. $i := 0$.

Шаг 3. $i := i + 1$. Если $i > k$, то идти к 2. Если $\pi \langle u_w \rangle > \pi_1$, то идти к 4. Идти к 3.

Шаг 4. $v := |P_1| + 1$; $\sigma_1^v := \pi \langle u_w \rangle$; $P_g := P_g \setminus \{\pi \langle u_w \rangle\}$; $P_1 := P_1 \cup \{\sigma_1^v\}$. Идти к 2.

Шаг 5. $i := 0$. Если $P_g = \emptyset$, то идти к 20. $\mu_0 := \prod \{\pi \langle u_w \rangle \in P_g\}$ и $v_0 = 1 \cdot \pi$.

Шаг 6. Для каждого π_u ($u \in J$) выбираем такое двухблочное разбиение

$\delta_u \geq \pi_u$, для которого $I_u = \min_{\delta_u \geq \pi_u} I(\mu_1/\nu_1 \cdot \delta_u)$

Шаг 7. Среди δ_u ($u \in J$) выбираем такое δ_u , для которого $I_u = \min_{u \in J} I_u$.

Шаг 8. Среди $\sigma_j^v \in \bigcup_{i \in J} P_i$ выбираем такое σ_j^v , для которого

$$I^0 = I(\mu_1/\nu_1 \cdot \sigma_j^v) = \min_{\sigma_j^v} I(\mu_1/\nu_1 \cdot \sigma_j^v).$$

Шаг 9. Если $i > 0$, то идти к 13.

Шаг 10. Если $I_u \geq I^0$, то идти к 12.

Шаг 11. $j := u$; $v := \|P_j\| + 1$; $\sigma_j^v := \delta_j$; $P_j := P_j \cup \{\sigma_j^v\}$.

Шаг 12. $P_j^+ := P_j^+ \cup \{\sigma_j^v\}$. Идти к 18.

Шаг 13. $t := t + 1$. Если $I_u \geq I^0$, то идти к 15.

Шаг 14. $I^0 := I_u$; $j := u$; $v := \|P_j\| + 1$; $\sigma_j^v := \delta_j$; $P_j := P_j \cup \{\sigma_j^v\}$.

Шаг 15. Если $I^0 > L^* - \|X_1'\| - t$, то идти к 17.

Шаг 16. $h := \|Q_1^0\| + 1$; $\rho_1^h := \sigma_j^v$; $Q_1^0 := Q_1^0 \cup \{\rho_1^h\}$.

Шаг 17. $P_j^1 := P_j^1 \cup \{\sigma_j^v\}$.

Шаг 18. $\nu_1 := \nu_1 \cdot \sigma_j^v$. Если $I(\mu_1/\nu_1) \neq 0$, то идти к 6.

Шаг 19. Образует разбиение $\theta_1 \geq \nu_1$, для которого

$$\|\theta_1\| = \begin{cases} I(\mu_1 / \prod_{\rho_1^h \in Q_1^0} \rho_1^h) - \text{для компонент без локальной} \\ \text{обратной связи;} \\ I(\mu_1 / \pi_1 \cdot \prod_{\rho_1^h \in Q_1^0} \rho_1^h) - \text{для компонент с локальной} \\ \text{обратной связью;} \end{cases}$$

Кодируя θ_1 двухблочными разбиениями, дополняем этими кодовыми разбиениями множество Q_1 .

Шаг 20. $i := i + 1$. Если $i > k$, то идти к 22.

Шаг 21. $t := 0$; $\mu_1 := \theta_1$;

$$\nu_1 = \begin{cases} 1 \pi - \text{для компонент без локальной} \\ \text{обратной связи;} \\ \pi_1 - \text{для компонент с локальной} \\ \text{обратной связью;} \end{cases}$$

Идти к 6.

Шаг 22. Конец.

На рис. 4 и 5 показаны ход и результаты декомпозиции МПА,

$$F_1: S \times \Psi(X'_1) \rightarrow \pi_1$$

s_p	B_1	$X(s_p, B_1)$
I	B_2	1
2	B_1	$\bar{x}_2$
	B_2	$x_2 x_5$
	B_3	$x_2 \bar{x}_5$
3	B_1	$x_2 x_4$
	B_3	$x_2 \bar{x}_4$
	B_4	$\bar{x}_2$
4	B_1	1
5	B_2	x_2
	B_6	$\bar{x}_2$
6	B_2	$x_2 x_5$
	B_3	$x_2 \bar{x}_5$
	B_5	$\bar{x}_2$
7	B_2	1
8	B_2	1
9	B_2	$x_2 x_5$
	B_3	$x_2 \bar{x}_5$
	B_4	$\bar{x}_2$

$$\pi_1 = \{B_1, B_2, B_3, B_4, B_5, B_6\} = \overline{1, 3; 2, 6; 4, 5; 7; 8; 9};$$

$$\theta_1 = \overline{1; 2; 3; 4; 5; 6; 7, 8; 9};$$

$$I(\theta_1/\pi_1) = 1.$$

$$F_2: S \times \Psi(X'_2) \rightarrow \pi_2$$

s_p	C_1	$X(s_p, C_1)$
I	C_2	$x_1 x_3$
	C_3	$\bar{x}_1 x_3$
	C_3	$\bar{x}_3$
2	C_1	$x_1 \bar{x}_2 x_3$
	C_2	$x_1 x_2 x_3$
	C_3	$\bar{x}_1$
	C_3	$\bar{x}_3$
3	C_3	x_2
	C_4	$\bar{x}_2$
4	C_1	$x_1 x_3$
	C_3	$\bar{x}_1 x_3$
	C_3	$\bar{x}_3$
5	C_3	x_2
	C_5	$\bar{x}_2$
6	C_3	x_2
	C_4	$\bar{x}_2$
7	C_2	1
8	C_2	1
9	C_3	x_2
	C_4	$\bar{x}_2$

$$\pi_2 = \{C_1, C_2, C_3, C_4, C_5\} = \overline{1; 2, 4; 3, 5, 6; 7, 8; 9};$$

$$\theta_2 = \overline{1; 2; 3, 6, 9; 4; 5; 7, 8};$$

$$I(\theta_2/\pi_2) = 1.$$

Рис. 4. Вспомогательные функции.

$A_1:$

B_1	B_j	$X(B_1, B_j)$	$Y(B_1)$
B_1	B_2	x_6	-
	B_1	$\bar{x}_6 x_2 x_4$	
	B_3	$\bar{x}_6 x_2 \bar{x}_4$	
	B_4	$\bar{x}_6 \bar{x}_2$	
B_2	B_1	$x_6 \bar{x}_2$	y_3
	B_2	$x_2 x_5$	
	B_3	$x_2 \bar{x}_5$	
	B_5	$\bar{x}_6 \bar{x}_2$	
B_3	B_1	x_6	$y_3 y_4$
	B_2	$\bar{x}_6 x_2$	
	B_5	$\bar{x}_6 \bar{x}_2$	
B_4	B_2	1	-
B_5	B_2	1	y_2
B_6	B_2	$x_2 x_5$	y_3
	B_3	$x_2 \bar{x}_5$	
	B_4	$\bar{x}_2$	

 $A_2:$

C_1	C_j	$X(C_1, C_j)$	$Y(C_1)$
C_1	C_2	$x_1 x_3$	y_5
	C_3	$\bar{x}_1 x_3$	
	C_3	$\bar{x}_3$	
C_2	C_1	$x_1 x_3 (\bar{x}_2 \vee x_7)$	y_5
	C_2	$x_1 x_2 x_3 \bar{x}_7$	
	C_3	$\bar{x}_1 \vee \bar{x}_3$	
C_3	C_3	x_2	-
	C_4	$\bar{x}_2 \bar{x}_7$	
	C_5	$\bar{x}_2 x_7$	
C_4	C_2	1	y_1
C_5	C_3	x_2	y_5
	C_4	$\bar{x}_2$	

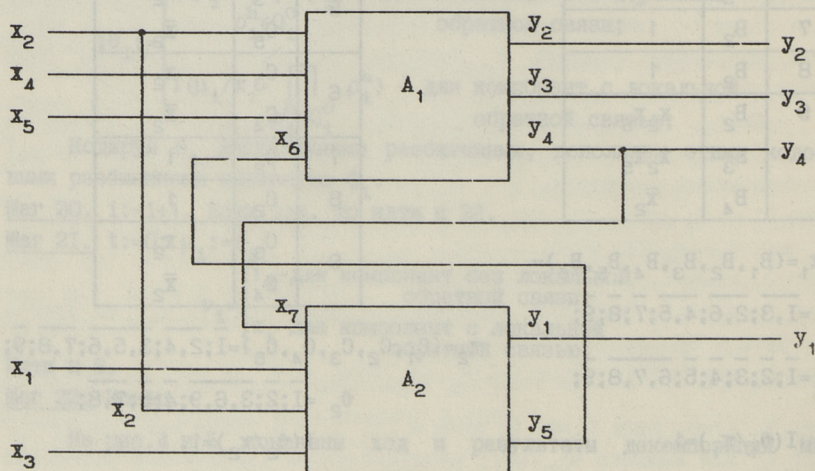
 $\pi\langle x_6 \rangle = \pi\langle y_5 \rangle = 3, 5, 6, 7, 8; 1, 2, 4, 9;$
 $\pi\langle x_7 \rangle = \pi\langle y_4 \rangle = 1, 2, 3, 6, 7, 8, 9; 4, 5.$


Рис.5. Результирующая сеть МПА.

изображенного на рис.2, по полученному для него ранее множеству разбиений. Результатом декомпозиции является изображенная на рис.5 сеть из двух компонентных автоматов, имеющих по 4 входа каждый и выходов – соответственно 3 и 2. Эта сеть может быть реализована, например, на двух ПЛМ с памятью с параметрами (4 входа, 3 выхода, 16 термов, 3 элемента памяти).

Экспериментальное исследование оптимизационного метода декомпозиции, выводы.

Описанные в настоящей работе методы оптимизации декомпозиционного синтеза реализованы программно и внедрены в практику проектирования. Эксперименты по исследованию предложенных методов проводились в системе DILOS на потоках МПА из библиотеки УАССМА-ЕС и на потоках сгенерированных программно МПА. Сравнения результатов и времени, затраченного на генерацию разбиений и на синтез сети МПА, производились между базовым вариантом системы DILOS и новым вариантом, использующим разработанные оптимизационные методы. Эксперименты проводились на ЭВМ ЕС-1055 в операционной системе СВМ при средней загрузке машины.

Основные параметры автоматов и полученных на их основе сетей для базового и нового алгоритмов представлены в таблицах 1 и 2. В таблице 1 приведены результаты обработки реальных управляющих автоматов из библиотеки УАССМА ЕС, а в таблице 2 – те же параметры для псевдослучайно сгенерированных МПА.

Обозначения : n – число состояний исходного МПА;

l – число входов исходного МПА;

k – число компонентных автоматов;

$n_{\max}$ – максимальное число состояний
компонентного автомата;

$l_{\max}$ – максимальное количество входов
компонентного автомата;

$l_{\text{ср}}$ – среднее количество входов
компонентного автомата.

Параметры, отмеченные штрихом, относятся к новому алгоритму, без штрихов – к старому.

Таблица 1

n	l	k	$l_{\max}$	$l'_{\max}$	l_{cp}	l'_{cp}	$n_{\max}$	$n'_{\max}$
41	9	5	10	6	9	5.4	18	14
30	13	4	12	12	8	7.5	23	9
43	14	8	10	7	8.9	6.0	29	11
50	12	6	10	8	9.2	7.2	14	9
36	19	4	18	16	15	13.5	11	6
45	17	7	16	12	15.6	11.4	29	12
84	23	12	20	15	15.4	11.8	23	7
66	18	9	17	15	13.1	10.9	17	14
70	30	13	32	29	27	24.7	20	16
32	11	3	10	8	7.3	7	19	13

Таблица 2

n	l	k	$l_{\max}$	$l'_{\max}$	l_{cp}	l'_{cp}	$n_{\max}$	$n'_{\max}$
32	11	4	9	7	6.5	5	20	16
44	20	7	18	16	12.6	12.4	18	17
37	18	7	11	9	8.3	7.1	13	13
57	12	5	10	9	7.2	5.4	7	5
41	11	8	11	8	8.5	8	12	9
24	10	3	10	7	5.7	5.7	26	22
53	25	10	26	22	18.2	18.1	14	10
68	15	8	16	13	16.9	15.6	8	6
88	21	8	23	18	23.9	23.5	9	9
100	30	10	30	25	26.9	26.1	7	3

Как видно из таблиц, внесение оптимизации в процедуры декомпозиции позволяет получить дополнительный выигрыш по основному ограничению - количеству входов - в среднем 10-20%.

Благодаря выигрышу в числе входов компонентов сети мы имеем возможность: во-первых, декомпонировать даже те устройства, компоненты которых ранее не удавалось разместить в элементах заданного базиса, а во-вторых существенно минимизировать те сети, которые получались ранее, за счёт сокращения числа компонент (что

становится возможным из-за лучшего использования их входов).

По количеству термов, которые должны иметь ПЛМ для реализации компонентных автоматов, также получено преимущество нового способа в сравнении со старым (выигрыш до 50%).

Из приведенных данных видно, что новые алгоритмы обеспечивают реализацию ДУУ в базисе ПЛУ при меньших затратах обору-дования по сравнению с использованными до настоящего времени.

Анализ затраченного на решение машинного времени показал, что новые алгоритмы медленнее старых (иногда в несколько раз), что является естественной платой за более тонкую оптимизацию.

Отметим, что перевод системы DILOS на более современные ЭВМ, в частности - на персональные ЭВМ типа IBM PC/AT, IBM PS/2, предполагаемый в ближайшем будущем, существенно повысит её возможности, в том числе и по быстродействию.

Л и т е р а т у р а :

1. H a r t m a n i s J., S t e a r n s R. E. Algebraic Structure Theory of Sequential Machines. N.-Y.: Prentice-Hall Inc., 1966. 209 p.
2. J a k o b s o n G., K e e v a l l i k A., L e i s P. Some Aspects of the Construction of the Microprogram Automata Networks // Proceedings of the Second International Symposium on Discrete Systems. Dresden. 1977. Vol. 1. P. 120-128.
3. К е э в а л л и к А. Э., Л а у с м а а Т. И. Декомпозиция конечных автоматов на основе информационного критерия // Труды ТПИ. 1985. №601, С. 3-II.
4. Б е р к м а н Б. Е. Кодирование при параллельной передаче данных и его информационный критерий // Труды ТПИ. 1986. №626.
5. К е э в а л л и к А., Б е р к м а н Б., Алгебра пар-неполностью определенных разбиений и критерий инфоресурса при декомпозиции автоматов // Труды ТПИ. 1988. №674. С. 12-20.
6. Б е р к м а н Б. Е. Декомпозиция микропрограммных автоматов в классе гомоморфизмов с использованием инфоресурса // Труды ТПИ. 1988. №674. С. 21-34.

B. Berkman, A. Keevallik

Mikroprogrammautomaatide dekomponeerimise
optimisatsioon inforessursi alusel

Kokkuvõte

Artiklis on välja töötatud uus mikroprogrammautomaatide dekompositsiooni meetod inforessursi alusel. Dekompositsiooni optimeerimiseks on sünteesi protseduur jagatud globaalsete ja lokaalsete transformatsioonide jadaks. On esitatud eksperimentaaluuringute tulemused, mis tõendavad uue meetodi efektiivsust.

66

B. Berkman, A. Keevallik

Microprogram Automata Decomposition Optimization
Using Inforesource Criterion

Abstract

A new method for microprogram automata decomposition divided into a sequence of global and local transformations is presented. It is shown that the partition pair algebra approach for finite automata decomposition supplemented with quantitative informational complexity measure leads to high efficiency decomposition algorithms.

ДЕКОМПОЗИЦИЯ МИКРОПРОГРАММНЫХ АВТОМАТОВ
В СЕТИ ИЗ АВТОМАТОВ МИЛИ

При декомпозиции автоматов проблема информационных связей между компонентными автоматами является одной из узловых. В рамках наиболее распространенного парного метода [1,2,3] декомпозиция строится по полному множеству разбиений без ограничений на базис и структуру сети. Известно, что в сетях из автоматов Мили могут возникать некорректные связи, приводящие к рекуррентной зависимости выходов автоматов друг от друга ("timing problem" [1]), и не всегда определен результирующий автомат сети. Поэтому при декомпозиции парным методом в качестве компонентных автоматов обычно используются автоматы Мура (в частности - полуавтоматы), для которых проблем некорректных связей не возникает. Нетрудно видеть, что при таком подходе передаваемая между компонентными автоматами информация относится исключительно к внутреннему состоянию реализуемого сетью автомата.

Другой подход был использован для декомпозиции в работе [4]. При предложенном в [4] подходе производится декомпозиция в сеть из автоматов Мили, и между автоматами передается таким образом информация не только с состояния реализуемого автомата, но и о текущих входах. Проблемы некорректных связей не возникают благодаря серьезному ограничению: компонентные автоматы сети работают последовательно во времени. Для определенного класса автоматов, допускающих разрез на слабо связанные компоненты [4], эта декомпозиция имеет по сравнению с декомпозицией на автоматы Мура следующие преимущества. Во-первых, благодаря характеру связей, может достигаться дополнительное ослабление зависимости компонентных автоматов от внешних входов, а во-вторых - имеется возможность распределить вычисление выходной функции между компонентными автоматами сети. Основным недостатком этого метода декомпозиции следует, по-видимому, считать его частный характер, невозможность построить сеть по заданному множеству разбиений в

общем случае и, в конечном итоге, сужение класса декомпозируемых автоматов.

Для устранения указанного недостатка предлагается рассматривать сети из работающих параллельно во времени автоматов Мили. Для решения проблемы некорректности связей у компонентных автоматов запрещается зависимость выходов, используемых для связи с другими компонентными автоматами, от части входов, поступающих от других компонентных автоматов. Тем самым устраняется возможность возникновения рекурсивной функциональной зависимости выходов компонентных автоматов друг от друга и решается проблема некорректных связей. В то же время, как легко показать, сети работающих последовательно во времени автоматов Мили, введенные в [4], сети с автоматом связи [5] и иерархические сети, предложенные в [6], представляют из себя частный случай вводимых нами сетей.

Определение 1. Под булевым пространством $\Phi(X)$, порождаемым множеством двоичных переменных $X=\{x_1, \dots, x_n\}$, будем понимать множество $\{0,1\}^n$, элементы которого – всевозможные двоичные вектора длины $n=|X|$, – соответствуют наборам значений переменных из X (здесь и ниже под $|X|$ будем понимать мощность множества X).

Для упрощения записи булевы вектора будем записывать без запятых и скобок. Например, $(0,1,0,1) \equiv 0101$.

Определение 2. Под микропрограммным автоматом (МПА) будем понимать систему $A=(S, X, Y, \delta, \lambda)$, где

$S=\{s_i | i=1, \dots, m\}$ – множество внутренних состояний;

$X=\{x_1, \dots, x_n\}$ – множество входных двоичных переменных (каналов), из которых МПА получает входные наборы z – элементы булева пространства $\Phi(X)$;

$Y=\{y_1, \dots, y_n\}$ – множество выходных двоичных переменных (каналов), на которые МПА выдает выходные наборы w – элементы булева пространства $\Phi(Y)$;

$\delta: D(\delta) \rightarrow S$ – функция переходов, где $D(\delta) \subseteq S \times \Phi(X)$ – область определения функции δ ;

$\lambda: D(\lambda) \rightarrow \Phi(Y)$ – функция выходов, где $D(\lambda) \subseteq S$ или $D(\lambda) \subseteq S \times \Phi(X)$ – область определения функции λ для моделей Мура или Мили, соответственно.

Пример прямой таблицы переходов МПА Мили приведен на рис.1.

Определение 3. Под сетью из МПА (Мили) будем понимать систему $N=(X,Y,B,f)$, где

1. X - множество входов сети.

2. Y - множество выходов сети.

3. $B=\{A_i | i \in J=\{1, \dots, k\}\}$ - множество компонентных МПА (базис сети); $A_i=(S_i, X_i, Y_i, \delta_i, \lambda_i)$ - i -й компонентный МПА Мили.

4. $f: (\bigcup_{i \in J} X_i \setminus X) \rightarrow \bigcup_{i \in J} Y_i$ - структура сети, где $y_t = f(x_p)$ ($y_t \in Y_i$; $x_p \in X_j$), если и только если вход x_p j -го компонентного автомата соединен с выходом y_t i -го компонентного автомата.

Будем называть сеть МПА Мили допустимой, если и только если множества входов X_i и выходов Y_i каждого компонентного автомата могут быть разбиты на три группы каждое (X'_i, X''_i, X'''_i и Y'_i, Y''_i, Y'''_i , соответственно) так, чтобы имело место :

$$1. X = \bigcup_{i \in J} X'_i;$$

$$2. Y \subseteq \bigcup_{i \in J} Y'_i;$$

$$3. \text{Существуют функции } \lambda'_i: S_i \times \Psi(X_i) \rightarrow \Psi(Y'_i);$$

$$\lambda''_i: S_i \times \Psi(X'_i) \times \Psi(X'''_i) \rightarrow \Psi(Y''_i)$$

$$\text{и } \lambda'''_i: S_i \rightarrow \Psi(Y'''_i),$$

где для любых $s_p \in S_i$ и $z \in \Psi(X_i)$ имеет место

$$\lambda'_i(s_p, z) = \lambda'_i(s_p, z) \times \lambda''_i(s_p, z') \times \lambda'''_i(s_p), \text{ причем } z' = pr(z, X'_i \cup X'''_i).$$

4. Имеются два сужения функции f :

$$f_1: \bigcup_{i \in J} X''_i \rightarrow \bigcup_{i \in J} Y''_i$$

и

$$f_2: \bigcup_{i \in J} X'''_i \rightarrow \bigcup_{i \in J} Y'''_i.$$

5. Множества $Y' = \bigcup_{i \in J} Y'_i$, Y''_i ($i \in J$) и Y'''_i ($i \in J$) попарно не пересекаются. Множества Y'_i и Y'_j могут пересекаться между собой, то есть допускается контактное "ИЛИ" на выходе сети.

Для допустимой сети МПА Мили может быть определен результирующий автомат сети A_N .

Определение 4. Под результирующим автоматом A_N допустимой сети МПА Мили $N=(X,Y,B,f)$ понимается микропрограммный автомат Мили $A_N=(S_N, X, Y, \delta_N, \lambda_N)$, где

1. X, Y - множества входных и выходных каналов автомата A_N , совпадающие с множествами входных и выходных каналов сети N , соответственно.

2. $S_N = \times_{i \in J} S_i$ - множество состояний результирующего автомата A_N .

3. $\delta_N: S_N \times \Phi(X) \rightarrow S_N$ - функция переходов автомата A_N , определяемая следующим образом:

$$\delta_N((s^1, \dots, s^k), z) = (\delta_1(s^1, z_1^1 \times z_1^2), \dots, \delta_k(s^k, z_k^1 \times z_k^2)),$$

где $z \in \Phi(X)$; $s^i \in S_i$; $z_i^j = \text{pr}(z, \Phi(X_i^j))$;

$$\text{pr}(z_1^m, \{x_t\}) = \text{pr}(\times_{i \in J} \lambda_i^m(s^i), \{f_2(x_t)\}) \text{ для любых } z_1^m \in \Phi(X_1^m) \text{ и } x_t \in X_1^m;$$

$$\text{pr}(z_1^n, \{x_t\}) = \text{pr}(\times_{i \in J} \lambda_i^n(s^i, z_i^1 \times z_i^2), \{f_2(x_t)\})$$

для любых $z_1^n \in \Phi(X_1^n)$ и $x_t \in X_1^n$; $(i \in J)$.

4. $\lambda_N: S_N \times \Phi(X) \rightarrow \Phi(Y)$ - функция выходов результирующего МПА A_N , причем для любого $y_j \in Y$ имеет место

$$\text{pr}(\lambda_N((s^1, \dots, s^k), z), \{y_j\}) = \begin{cases} \bigvee_i \text{pr}(\lambda_i^1(s^i, z_i^1 \times z_i^2 \times z_i^3), \{y_j\}), & \text{если} \\ & y_j \in Y'; \text{ (дизъюнкция берется по} \\ & \text{всем } i \text{ таким, что } y_j \in Y'_i); \\ \text{pr}(\lambda_i^1(s^i, z_i^1 \times z_i^2), \{y_j\}), & \text{если } y_j \in Y''_i; \\ \text{pr}(\lambda_i^1(s^i), \{y_j\}), & \text{если } y_j \in Y'''_i. \end{cases}$$

Здесь значения битов z_i^1 , z_i^2 , z_i^3 определяются также, как и выше.

Декомпозиция в сеть из автоматов Мили может осуществляться по полному множеству разбиений или покрытий и ее ход в основном аналогичен ходу декомпозиции в сети из автоматов Мура [3]. Задача построения строго формализованных процедур декомпозиции в сети из МПА Мили выходит за рамки настоящей работы. Поэтому отметим лишь основные идеи этой декомпозиции.

1. На этапе построения вспомогательных функций наряду с функциями вида $F_i: S \times \Phi(X_i) \rightarrow S_i$, ищем функции $G_i: S \times \Phi(X_i) \rightarrow \Phi(Y'_i)$, определяемые функцией выхода исходного МПА.

2. Множества Y''_1 , Y'''_1 , X''_1 , X'''_1 дополняются в ходе декомпозиции, причем при выборе выходных алфавитов компонентных автоматов сети учитываются потребности других компонентных автоматов и условие допустимости сети.

На рис.2-4 приведен пример декомпозиции МПА Мили, изображенного на рис.1 и его реализации сетью из ПЛМП (программируемых логических матриц с памятью) с параметрами (7 входов, 8 выходов, 32 термина, 6 элементов памяти).

s_p	s_q	$X(s_p, s_q)$	$Y(s_p, s_q)$
I	2	$\bar{x}_2 \bar{x}_4$	$y_1 y_2 y_3 y_4 y_5$
	2	$x_1 \bar{x}_2$	$y_1 y_2 y_3 y_4 y_5$
	3	$\bar{x}_1 \bar{x}_2 x_4$	$y_2 y_3 y_9 y_{10}$
	3	$x_2 \bar{x}_3 x_4$	$y_2 y_3 y_9 y_{10}$
	4	$x_2 x_3$	$y_{14} y_{15}$
	5	$x_2 \bar{x}_3 \bar{x}_4$	$y_1 y_4$
2	6	$x_1 x_3$	$y_{19} y_{20}$
	7	$\bar{x}_3$	$y_8 y_{19}$
	7	$\bar{x}_1$	$y_8 y_{19}$
3	4	I	$y_{14} y_{15}$
4	5	$\bar{x}_5 \bar{x}_6 x_7$	$y_1 y_4$
	6	$x_5 x_6$	$y_1 y_4 y_{15}$
	7	$\bar{x}_5 x_6$	$y_8 y_{19}$
	8	$x_5 \bar{x}_6$	$y_2 y_{10} y_{13}$
	8	$\bar{x}_5 \bar{x}_6 \bar{x}_7$	$y_2 y_{10} y_{13}$
5	6	$\bar{x}_7 \bar{x}_2 x_3$	$y_1 y_4 y_{15}$
	7	$x_7 x_2$	$y_8 y_{19}$
	8	$\bar{x}_7 x_2$	$y_2 y_{10} y_{13}$
	9	$x_7 \bar{x}_2$	$y_{10} y_{13} y_{15}$
	10	$\bar{x}_2 \bar{x}_3 \bar{x}_7$	$y_8 y_{10} y_{20}$

s_p	s_q	$X(s_p, s_q)$	$Y(s_p, s_q)$
6	2	$x_3 x_4 \bar{x}_8$	$y_6 y_7 y_8$
	3	$\bar{x}_3 x_4$	$y_{11} y_{12} y_{13}$
	6	$x_3 x_4 x_5 \bar{x}_6 x_7 x_8$	$y_1 y_4 y_{15}$
	7	$\bar{x}_4 \bar{x}_5 \bar{x}_7$	$y_8 y_{19}$
	9	$\bar{x}_4 x_5 x_6$	$y_{10} y_{13} y_{15}$
	10	$\bar{x}_4 x_5 \bar{x}_6$	$y_8 y_{19} y_{20}$
	10	$\bar{x}_4 \bar{x}_6 x_7$	$y_8 y_{19} y_{20}$
	8	$\bar{x}_4 \bar{x}_5 x_6 x_7$	$y_6 y_7 y_{12}$
	8	$x_3 x_4 x_6 x_8$	$y_6 y_7 y_{12}$
	8	$x_3 x_4 \bar{x}_7 x_8$	$y_6 y_7 y_{12}$
	8	$x_3 x_4 \bar{x}_5 x_8$	$y_6 y_7 y_{12}$
7	8	I	$y_2 y_{10} y_{13}$
8	8	$\bar{x}_1$	—
	9	x_1	$y_8 y_9 y_{16}$
9	5	x_2	$y_{11} y_{14} y_{15}$
	6	$\bar{x}_2$	$y_3 y_6 y_{15}$
10	8	x_1	$y_8 y_9 y_{16}$
	9	$\bar{x}_1$	$y_9 y_{16}$

Рис. 1. МПА Мяли.

$F_1, G_1:$

s_p	B_1	$X(s_p, B_1)$	Y'_1	Y''_1
I	B_2	$\bar{x}_2 \bar{x}_4$	$y_2 y_3$	$v_1 v_2$
	B_2	$\bar{x}_2 x_4$	$y_2 y_3$	v_1
	B_2	$x_2 \bar{x}_3 x_4$	$y_2 y_3$	—
	B_5	$x_2 x_3$	y_{14}	—
	B_6	$x_2 \bar{x}_3 \bar{x}_4$	—	v_2
	B_7	$\bar{x}_3$	y_{19}	v_1
2	B_7	$\bar{x}_3$	y_{19}	—
3	B_5	1	y_{14}	—
4	B_6	$\bar{x}_5 \bar{x}_6 x_7$	—	v_1
	B_7	$x_5 x_6$	—	v_2
	B_7	$\bar{x}_5 x_6$	y_{19}	$v_1 v_2$
	B_8	$x_5 \bar{x}_6$	$y_2 y_{13}$	—
	B_8	$\bar{x}_6 \bar{x}_7$	$y_2 y_{13}$	—
	B_9	$\bar{x}_2 x_3 \bar{x}_7$	$y_1 y_4 y_{15}$	—
5	B_7	$x_2 x_7$	$y_8 y_{19}$	v_1
	B_8	$x_2 \bar{x}_7$	$y_2 y_{10} y_{13}$	—
	B_8	$\bar{x}_2 x_7$	$y_{10} y_{13} y_{15}$	v_1
	B_9	$\bar{x}_2 \bar{x}_3 \bar{x}_7$	$y_8 y_{10} y_{20}$	—
	B_4	$x_3 x_4 x_6$	—	—
	B_4	$x_3 x_4 \bar{x}_7$	—	—
6	B_4	$x_3 x_4 \bar{x}_5$	—	—
	B_3	$x_3 x_4 x_5 \bar{x}_6 x_7$	—	$v_1 v_2$
	B_2	$\bar{x}_3 x_4$	$y_{11} y_{12} y_{13}$	v_1
	B_7	$\bar{x}_4 \bar{x}_5 \bar{x}_7$	$y_8 y_{19}$	v_1
	B_8	$\bar{x}_4 x_5 x_6$	$y_{10} y_{13} y_{15}$	v_1
	B_9	$\bar{x}_4 x_5 \bar{x}_6$	$y_8 y_{19} y_{20}$	v_1
	B_9	$\bar{x}_4 \bar{x}_6 x_7$	$y_8 y_{19} y_{20}$	v_1
	B_4	$\bar{x}_4 \bar{x}_5 x_6 x_7$	—	v_2
	B_8	1	$y_2 y_{13}$	—
	B_8	1	—	—
9	B_6	x_2	$y_{11} y_{14}$	v_1
	B_7	$\bar{x}_2$	y_3	—
10	B_8	1	—	—

 $F_2, G_2:$

s_p	C_1	$X(s_p, C_1)$	Y'_1
I	C_1	$v_1 v_2$	$y_1 y_4 y_5$
	C_1	$x_1 v_1$	$y_1 y_4 y_5$
	C_2	$\bar{x}_1 v_1 \bar{v}_2$	$y_9 y_{10}$
	C_5	$\bar{v}_1 \bar{v}_2$	y_{15}
	C_2	$\bar{v}_1 v_2$	$y_1 y_4$
	C_2	$\bar{x}_1$	y_{20}
2	C_2	$\bar{v}_1$	y_8
	C_2	$\bar{x}_1$	y_8
3	C_2	1	y_{15}
4	C_1	$v_1 v_2$	y_8
	C_2	$v_1 \bar{v}_2$	$y_1 y_4$
	C_2	$\bar{v}_1 v_2$	$y_1 y_4 y_{15}$
	C_2	$\bar{v}_1 \bar{v}_2$	y_{10}
	C_1	v_1	—
	C_2	$\bar{v}_1$	—
6	C_1	$\bar{x}_8 \bar{v}_1 \bar{v}_2$	$y_6 y_7 y_8$
	C_1	$v_1 \bar{v}_2$	—
	C_2	$v_1 v_2$	$y_1 y_4 y_{15}$
	C_2	$\bar{v}_1 v_2$	$y_6 y_7 y_{12}$
	C_2	$x_8 \bar{v}_1 \bar{v}_2$	$y_6 y_7 y_{12}$
	C_2	1	y_{10}
7	C_2	1	y_{10}
8	C_1	x_1	$y_8 y_9 y_{16}$
	C_2	$\bar{x}_1$	—
9	C_2	v_1	y_{15}
	C_2	$\bar{v}_1$	$y_6 y_{15}$
10	C_1	$\bar{x}_1$	$y_8 y_9 y_{16}$
	C_2	x_1	$y_9 y_{16}$

$$\tau_2 = \overline{1, 2, 7, 9; 3, 4, 5, 6, 8, 10} = \{C_1, C_2\}$$

$$\tau_1 = \overline{1; 2, 3; 2, 6; 2, 8; 4; 5; 6, 7; 8, 9; 10} = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9\}$$

Рис. 2. Вспомогательные функции.

S_1	$\Phi(Y_1^m)$	$\Phi(Z_1)$
B_1	101	0000
B_2	001	0001
B_3	011	0011
B_4	010	01-1
B_5	101	1000
B_6	100	1001
B_7	111	1011
B_8	110	11-1
B_9	000	1010

S_2	$\Phi(Y_2^m)$
C_1	0
C_2	1

$\Phi(Z_2) \times \Phi(X_2^m)$	S
0 101	1
0 0--	2
1 011	3
1 101	4
1 100	5
1 -11	6
0 111	7
1 -10	8
0 110	9
1 000	10

$\Phi(Z_1) \times \Phi(X_1^m)$	S
0000 -	1
0--1 0	2
0001 1	3
1000 -	4
1001 -	5
-011 1	6
1011 0	7
-1-1 1	8
11-1 0	9
1010 -	10

$$X_1^m = Z_2 = Y_2^m = \{u_1\};$$

$$X_2^m = Y_1^m = \{u_2, u_3, u_4\};$$

$$\tau_1 = \overline{1; 2, 3; 2, 6; 2, 8; 4; 5; 6; 7; 8, 9; 10} = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9\};$$

$$\tau\langle u_2 \rangle = \overline{2, 3, 6, 8, 10; 1, 4, 5, 6, 7, 8, 9} \geq \tau_1;$$

$$\tau\langle u_3 \rangle = \overline{1, 2, 3, 4, 5, 10; 2, 6, 7, 8, 9} \geq \tau_1;$$

$$\tau\langle u_4 \rangle = \overline{2, 3, 6, 8, 10; 1, 4, 5, 6, 7, 8, 9} \geq \tau_1;$$

$$\tau\langle u_1 \rangle = \tau_2 = \overline{1, 2, 7, 9; 3, 4, 5, 6, 8, 10}.$$

Здесь $Z_1 = \{z_1, z_2, z_3, z_4\}$ и $Z_2 = \{u_1\}$ - множества кодовых переменных для кодирования внутренних состояний компонентных автоматов A_1 и A_2 , соответственно.

Рис. 3. Функции выхода и соединения. Коды состояний.

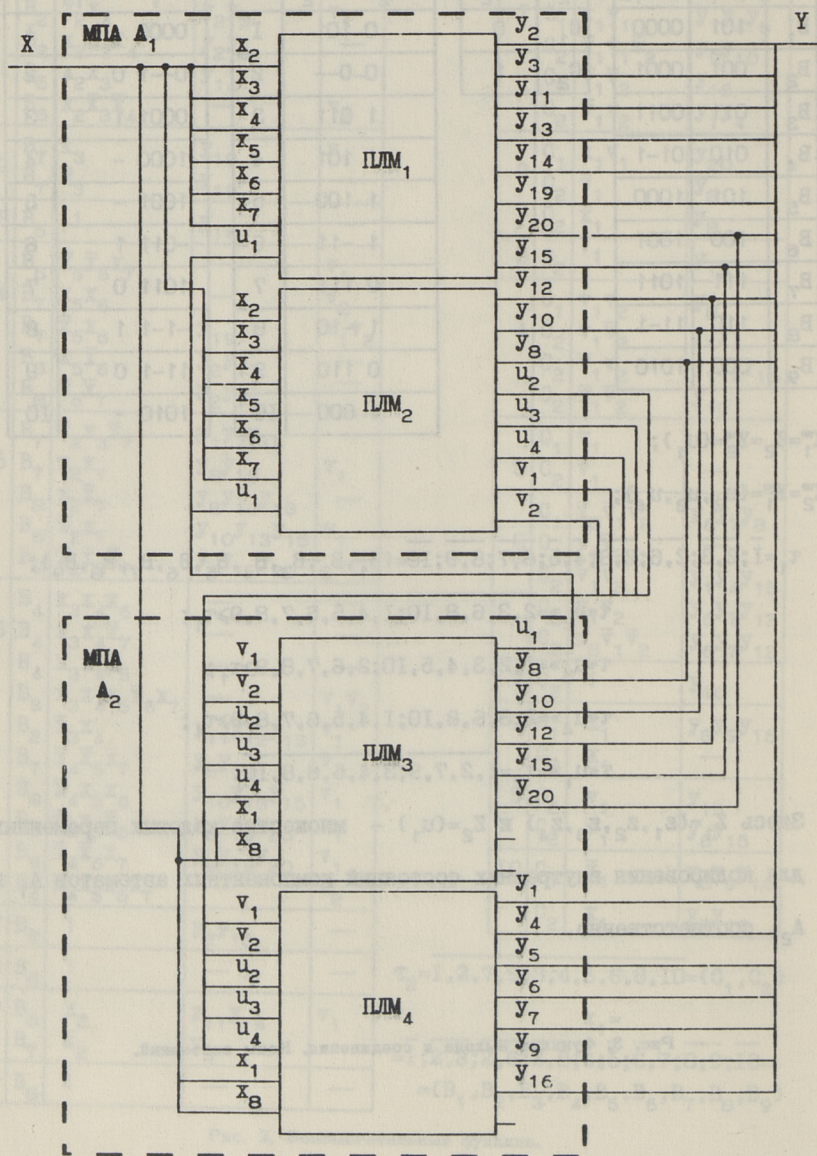


Рис. 4. Результирующая сеть ПЛМ.

Заметим, что в результате декомпозиции в сеть из автоматов Мили, модель которой была предложена в настоящей работе, нами получена реализация исходного МПА Мили сетью из 4-х ПЛМП(7,8,32,6), тогда как в работе [5] для реализации этого же МПА потребовалось 7 ПЛМП. Добавим также, что для МПА из работы [6] при использовании предложенной модели сети требуется 3 ПЛМП (8,8,48,6), тогда как в работе [6] потребовалось 4 таких ПЛМП.

В заключение отметим, что выигрыш в аппаратных затратах на реализацию автоматов в указанных примерах получен за счет использования более общей модели сети из МПА Мили. Возможны также случаи, когда за счет более общей модели сети могут быть получены реализации для автоматов, в принципе нереализуемых при заданных ограничениях БИС в классе сетей, описанных в [5,6].

Л и т е р а т у р а :

1. H a r t m a n i s J., S t e a r n s R. E. Algebraic Structure Theory of Sequential Machines. N.-Y.: Prentice-Hall Inc., 1966. 209 p.
2. J a k o b s o n G., K e e v a l l i k A., L e i s P. Some Aspects of the Construction of the Microprogram Automata Networks // Proceedings of the Second International Symposium on Discrete Systems. Dresden. 1977. Vol. 1. P. 120-128.
3. Б е р к м а н Б. Е. Декомпозиция микропрограммных автоматов в классе гомоморфизмов с использованием инфоресурса // Тр. Таллиннск. политехн. ин-та. 1988. №674. С. 21-34.
4. Б а р а н о в С. И., К и л л е н б е р г Х. Декомпозиция микропрограммных автоматов // Автоматика и вычислительная техника. 1978. №6. С. 5-II.
5. Л е в и н И. С. Декомпозиционный синтез автоматов на ПЛМ с памятью // Автоматика и вычислительная техника. 1986. №2. С. 66-73.
6. Л е в и н И. С. Иерархическая модель взаимодействия микропрограммных автоматов// Автоматика и вычислительная техника. 1987. №3. С. 77-83.

Mikroprogrammautomaatide dekomponeerimine
Mealy automaatide võrguks

Kokkuvõte

Artiklis käsitletavat probleemi on seotud mikroprogrammautomaatide dekompositsiooniga paaride meetodil, kasutades Mealy automaatide võrku. Välja on töötatud Mealy automaatide võrgu originaalne mudel, mille jaoks on esitatud võrgu resulteeriv automaat. Kasutades antud võrgu mudelit, on võrreldes teiste meetoditega saadud väiksema PIM-de arvuga automaatide realiseerimise.

B. Berkman

Microprogram Automata Decomposition into
Networks of Mealy Automata

Abstract

A problem of microprogram automata decomposition into Mealy automata networks is considered in terms of pair decomposition method. The original model of Mealy automata network is proposed, for which resulted automaton of network is presented. Thanks to the proposed network model automata realizations are obtained on a fewer number of PLA as compared to other methods.

Л.Ф. Каширова, М.Э. Круус,
П.С. Эллервее

ПОСТРОЕНИЕ КОНТРОЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛЯ УПРАВЛЯЮЩИХ АВТОМАТОВ, РЕАЛИЗОВАННЫХ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ МАТРИЦАХ

Введение. В настоящее время существует несколько различных подходов к проблеме тестирования дискретных управляющих устройств (ДУУ). Среди них следует выделить метод, предложенный впервые Хенни [1], в котором проверяющий тест (контрольная последовательность (КП)) синтезируется на основе конечно-автоматной модели исследуемого устройства. Несмотря на свою универсальность (построение КП не зависит от конкретной схемной реализации устройства), метод Хенни пока не нашел широкого практического применения, поскольку он ориентирован на модель абстрактного конечного автомата, с помощью которой описать реальные ДУУ трудоемко и часто невозможно. Практическое значение метода Хенни значительно увеличивается при использовании модели микропрограммного автомата (МПА), как это предложено в [2]. Поскольку КП синтезируется на основе модели устройства, то возникает вопрос об эффективности получаемой КП, охарактеризовать которую можно количеством обнаруживаемых неисправностей на структурном (схемном) уровне. Очевидно, что эффективность КП зависит от использованного элементного базиса.

В настоящей работе в качестве элементного базиса при реализации комбинационной части ДУУ использованы программируемые логические матрицы (ПЛМ), которые широко распространены на сегодняшний день. Приведена классификация неисправностей и описаны свойства КП, обнаруживающей эти неисправности. Показано, что КП, построенная по методу [2], позволяет обнаружить подавляющее большинство неисправностей в ДУУ, реализованном на ПЛМ.

Основные понятия. Рассмотрим конечный автомат $A = (I, S, O, \delta_A, \lambda_A)$, где I – входной алфавит, S – множество состояний, O – выходной алфавит, $\delta_A: S \times I \rightarrow S, \lambda_A: S \times I \rightarrow O$ – функции переходов и выходов соответственно. В дальнейшем будем предполагать, что A – минимальный, сильносвязный автомат, имеющий диагностическую последовательность, т.е. A – контролируемый автомат [1].

Обозначим через I^* множество слов в алфавите I , включая пустое слово ϵ . Расширим функции δ_A, λ_A на множество I^* согласно [3], расширенные функции δ_A, λ_A будем обозначать через $\bar{\delta}_A, \bar{\lambda}_A$. Пусть $p = i_1 \dots i_k \in I^*$, число $d(p) = k$ назовем длиной последовательности p . Функция $\delta_A(S, p)$ определяет состояние, в которое перейдет автомат A из состояния S под действием последовательности p , $\lambda_A(S, p)$ – определяет ту выходную последовательность, которую выдает автомат, осуществляя переход из состояния S под действием последовательности p .

Определим микропрограммный автомат B , имеющий L двоичных входов и N двоичных выходов как систему $B = (S, \{0,1\}^L, \{0,1\}^N, \delta_B, \lambda_B)$, где $S = \{S_i / i = 1, 2, \dots, M\}$ – множество состояний, $\{0,1\}^L$ – входной алфавит, $\{0,1\}^N$ – выходной алфавит, $\delta_B: D(\delta) \rightarrow S$ – функция переходов, где $D(\delta) \subseteq S \times \{0,1\}^L$ – область определения функции δ_B , $\lambda_B: D(\lambda) \rightarrow \{0,1\}^N$ – функция выходов, $D(\lambda) \subseteq S \times \{0,1\}^L$ – область определения функции λ_B .

Пару (S, t) , где $s \in S, t \in \{0,1\}^L$ назовем переходом автомата B , если $\exists s' \in S (s' = \delta_B(S, t))$.

Характерным свойством микропрограммного автомата является то, что каждый его переход определен только несколькими переменными из множества $\{x_1, x_2, \dots, x_k\}$, которые назовем существенными. Сопоставим множеству переходов микропрограммного автомата множество интервалов $U = \{u_1, u_2, \dots, u_r\}$ в пространстве $\{0,1\}^L$, где $u_i, i = 1, \dots, r$ содержит только существенные переменные.

Пусть $U_s \subseteq U$ определяет множество всех переходов из состояния S . В дальнейшем интервалу $u_i \in U$ будем давать теоретико-множественную интерпретацию и использовать теоретико-множественные операции при работе с интервалами.

Обозначим через F множество неисправностей, которые могут возникать в автомате B , пусть B' – автомат B , в котором присутствует некоторое множество неисправностей из F , $B' = (S', \{0,1\}^L, \{0,1\}^N, \delta'_B, \lambda'_B)$.

Рассмотрим автоматы A и B , связанные между собой отношением гомоморфизма, таким образом, что существуют отображения $h_1: U \rightarrow I, h_2: \{0,1\}^N \rightarrow O, h_3: S \rightarrow S$, где h_3 – тождественное отображение. Пусть отображения h_1 и h_2 удовлетворяют следующим условиям:

$$\forall u_i \in U \forall t, t' \in U; \exists s \in S (h_1(t) = h_1(t') = i \in I \& h_2(\lambda_B(S, t)) = h_2(\lambda_B(S, t')) = \lambda_A(S, i) \& \delta_B(S, t) = \delta_B(S, t') = \delta_A(S, i)). \quad (I)$$

Из выражения (I) следует, что h_1 – взаимно однозначное отображение между множествами U и I .

Последовательность $\beta^0 \in \{0,1\}^{L*}$ назовем проверяющей последовательностью для автомата B , если β^0 отличает автомат B от B' , т.е. выполняется следующее:

$$\forall s \in S \forall s' \in S' (\bar{\lambda}_B(S, \beta^0) \neq \bar{\lambda}_{B'}(S', \beta^0).$$

Построим контрольную последовательность $\alpha = \alpha_1 \dots \alpha_k \in I^*$ для автомата A методом, изложенным в [1]. Сопоставим ей последовательность $\beta = h_1^{-1}(\alpha_1) h_1^{-1}(\alpha_2) \dots h_1^{-1}(\alpha_k)$. В [2] предложен метод построения последовательности β , использующий описание автомата B .

Последовательность $\beta^0 = \beta_1^0 \beta_2^0 \dots \beta_k^0$ назовем доопределением последовательности β (обозначение $\beta^0 \leq \beta$) если $\beta_j^0 \in \beta_j$, где $\beta_j \in U, j = 1, 2, \dots, k$.

В данной работе будем исследовать свойства последовательности β , чтобы найти такое доопределение $\beta^0 \leq \beta$, которое является проверяющей последовательностью для автомата B , относительно заданного множества неисправностей из F .

Рассмотрим микропрограммный автомат B , реализованный с помощью двухуровневой ПЛМ (L, N, q, V) с памятью, где L – число входов, N – число выходов, q – число термов, V – число разрядов регистра в цепи обратной связи (рис. 1). На рисунке 1 входные линии ПЛМ помечены символами $x_1, \dots, x_L$, выходные – $y_1, \dots, y_N$, они являются входными и выход-

ными переменными автомата В соответственно. $D_1, \dots, D_V$ - функции возбуждения триггеров, их значения определяют текущее состояние автомата. $Q_1, \dots, Q_V$ - выходные функции триггеров, определяющие значения на выходах элементов памяти, эти значения соответствуют последующему состоянию автоматов.

Будем рассматривать неисправности в точках пересечения линий ПЛМ (короткое замыкание между линиями, неисправности типа 0,1 на линии сводятся к неисправностям этого класса).

Появление соединения в точке пересечения, где его не должно быть, назовем неисправностью типа ПС.

Нарушение соединения в точке пересечения, если оно там было, назовем неисправностью НС.

Множество неисправностей F разобьем на следующие подмножества:

1. F_1 - множество неисправностей типа НС, ПС в схеме или.

2. F_2 - неисправности типа ПС на линиях, соответствующих инверсным значениям существенных переменных терма.

3. F_3 - неисправности типа НС на линиях, соответствующих существенным входным переменным автомата В.

4. F_4 - неисправности типа НС на линиях, соответствующих существенным переменным текущего состояния.

5. F_5 - неисправности типа ПС на линиях, соответствующих несущественным входным переменным.

Пусть h_4 - взаимно однозначное отображение $S \leftrightarrow \{0,1\}^V$, задающее кодировку состояний автомата В.

Сформируем условия, при выполнении которых последовательность β^0 является проверяющей последовательностью для автомата В относительно неисправностей из множества F .

Утверждение I. Последовательность $\beta_0 \leq \beta$ является проверяющей последовательностью для автомата В относительно неисправностей из F_1 .

Доказательство. Пусть $\beta \leq \beta$ - некоторое доопределение последовательности β . Используя отображение h_1 , сопоставим β^0 контрольную последовательность $\alpha = \alpha_1 \dots \alpha_k$. Из [1]

следует, что α проверяет множество переходов в A и идентифицирует множество состояний S . Поскольку S является также множеством состояний автомата B , то из условия (I) следует, что β^0 идентифицирует состояния автомата B и для каждого $u_i \in U$ проверяет один переход (S, t) , где $s \in S$, $t \in u_i$. В присутствии неисправностей из F_1 меняется либо код последующего состояния автомата, т.е. $h_4(\delta_B(S, t)) \neq h_4(\delta'_B(S, t))$, либо выходная функция $\lambda_B(S, t) \neq \lambda'_B(S, t)$. Поскольку h_4 — взаимно однозначное отображение, то $\delta_B(S, t) \neq \delta'_B(S, t)$. Из способа построения последовательности β^0 [1] следует, что утверждение доказано.

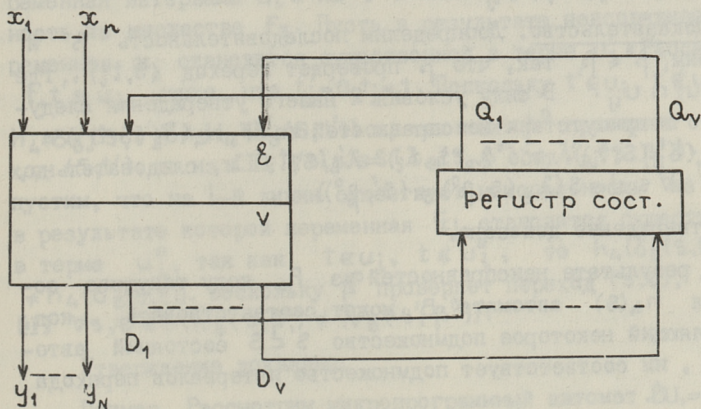


Рис. 1. Реализация МПА на ПЛМ.

Утверждение 2. Последовательность β^0 является проверяющей последовательностью для автомата B относительно множества неисправностей из F_2 .

Доказательство. Пусть i -му терму в ПЛМ соответствует интервал перехода u_i , но в результате неисправностей из F_2 неисправному терму соответствует интервал $u'_i = 0$, в который хотя бы одна переменная входит как со знаком инверсии, так и без него. И в силу этого либо $h_4(\delta_B(S, t)) \neq h_4(\delta'_B(S, t))$, либо выходная функция $\lambda_B(S, t) \neq \lambda'_B(S, t)$, так как $u'_i = 0 = u_i$. Здесь $h_4(S)$ — текущее состояние автомата, соответствующее i -му терму. Вследствие этого из [1] $\forall s, s' \in S (\lambda_B(S, \beta^0) \neq \lambda'_B(S, t))$. Утверждение доказано.

Допустим, что в ПЛМ существуют неисправности из множества F_3 . Рассмотрим множество интервалов U , найдем в нем подмножество $U_s \subset U$, определяющее множество переходов из состояния S в состояние множества $\{s_1, s_2, \dots, s_k\}$.

Утверждение 3. Если

$$1. \exists u_{ij} \in U_s (u_{ij} \cap u_i \neq \emptyset \& u_{ij} \neq u_i)$$

$$2. \forall t \in u_i \forall t' \in u_{ij} (h_4(\delta_B(s, t)) \vee h_4(\delta_B(s, t')) \neq h_4(\delta_B(s, t'))),$$

либо $\lambda_B(s, t) \vee \lambda_B(s, t') \neq \lambda_B(s, t')$, то последовательность β^0 является проверяющей последовательностью для автомата B относительно неисправностей из F_3 , если β^0 проверяет переход (s, t_i) , где $t_i \in u_i \cap u_{ij}$.

Доказательство. Доопределим последовательность β и построим $\beta^0 \leq \beta$ так, что β^0 проверяет переход (s, t_i) , где $t_i \in u_i \cap u_{ij}$. В силу условия 2 нашего утверждения следует, что в присутствии неисправностей из F_3 $h_4(\delta_B(s, t_i)) \neq h_4(\delta'_B(s, t_i))$, либо $\lambda_B(s, t_i) = \lambda'_B(s, t_i)$ и, следовательно, из [I] $\forall s, s' \in S (\bar{\lambda}_B(s, \beta^0) \neq (s', \beta^0))$.

Утверждение доказано.

В результате неисправностей из F_4 коду текущего состояния $h_4(s)$ автомата B может соответствовать код, определяющий некоторое подмножество $S' \subset S$ состояний автомата B , им соответствует подмножество интервалов перехода $U_{S'} \subset U_S = U'$.

Утверждение 4. Если

$$1. \exists u_j \in U' (u_j \neq u_i \& u_j \cap u_i \neq \emptyset)$$

$$2. \forall t \in u_i \forall t' \in u_j (h_4(\delta_B(s, t)) \vee h_4(\delta_B(s_j, t')) \neq$$

$$\neq h_4(\delta_B(s_j, t')), \text{ либо } \lambda_B(s, t) \vee \lambda_B(s_j, t') \neq \lambda_B(s_j, t'),$$

то последовательность β^0 является проверяющей последовательностью для автомата B относительно неисправностей из F_3 , если она проверяет переход (s_j, t') автомата B , где $s_j \in S'$.

Доказательство. Построим $\beta^0 \leq \beta$ так, что β^0 проверяет переход (s_j, t') , тогда из условия 2 утверждения следует, что $h_4(\delta_B(s_j, t')) \neq h_4(\delta'_B(s_j, t'))$, либо $\lambda_B(s_j, t') \neq \lambda'_B(s_j, t')$ и из [I], $\forall s, s' \in S (\bar{\lambda}_B(s, \beta^0) \neq \bar{\lambda}'_B(s', \beta^0))$.

Утверждение доказано.

Допустим, что в схеме присутствуют неисправности из множества F_5 . На множестве переходов $\{S\} \times u_i$ рассмотрим пару переходов (s, t) и (s, t') , где $t = t_1 \dots t_L$, $t' = t_1 \dots t_L$ и $t_j \neq t'_j$, если переменная x_j является несущественной для интервала u_i .

Утверждение 5. Последовательность β^0 является проверяющей последовательностью относительно неисправностей из множества F_5 , если β^0 проверяет как переход (s_i, t) , так и переход (s_i, t') автомата В.

Доказательство. Допустим, что x_j — несущественная переменная интервала u_i и на i -й линии существует неисправность из множества F_5 . Пусть в результате неисправности переменная x_j становится существенной в терме u'_i . Рассмотрим $t, t' \in u_i$ такие, что $t'_j = 0, t_j = 1$. Поскольку $t' \in u_i$, $t' \notin u'_i$, то $h_4(\delta_B(s, t')) \neq h_4(\delta'_B(s, t'))$ и, так как β^0 проверяет переход (s, t') , то из [I] $\forall s, s' \in S(\lambda_B(s, \beta^0) \neq \lambda'_B(s', \beta^0))$. Допустим, что на i -й линии существует неисправность из F_5 , в результате которой переменная $\bar{x}_j$ становится существенной в терме u''_i , так как $t \in u_i$, $t \notin u''_i$, то $h_4(\delta_B(s, t)) \neq h_4(\delta'_B(s, t))$, поскольку β^0 проверяет переход (s, t) , то из [I] $\forall s, s' \in S(\bar{\lambda}_B(s, \beta^0) \neq \bar{\lambda}'_B(s', \beta^0))$.

Утверждение доказано.

Пример. Рассмотрим микропрограммный автомат В, заданный таблицей, изображенной на рис. 2. Сопоставим ему автомат А, связанный с В отношением гомоморфизма. Таблица переходов, выходов автомата А изображена на рис. 3. Отображение $h_1: \bar{x}_1 x_2 \rightarrow i, x_1 \rightarrow i_2, \bar{x}_1 \bar{x}_2 \rightarrow i_3, h_2, h_3$ — тождественные отображения.

Автомат В реализован с помощью 2-уровневой ПЛМ (2, 3, 7, 2) с памятью, комбинационная часть которой изображена на рис. 4. Диагностическая последовательность $p_A = i_2^2$ автомата А, $p_B = x_1^2$ — автомата В.

Рассмотрим множество неисправностей, возникающих на 4-й линии. 0 — обозначение неисправностей типа ПС, $\odot$ — неисправностей типа НС. Каждая неисправность снабжена номером.

$F = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ — множество неисправно-

S_H	S_K	U	y	$\delta_4(S_H)$	$\delta_4(S_K)$
1	4	$\bar{x}_1 \ x_2$	—	0 0	1 1
	2	x_1	—		0 1
2	1	$\bar{x}_1 \ x_2$	y_1	0 1	0 0
	3	x_1	—		1 0
3	2	$\bar{x}_1 \ \bar{x}_2$	y_1	1 0	0 1
	1	x_1	y_1		0 0
4	2	x_1	y_1	1 1	0 1

Рис. 2. Прямая таблица переходов МПА.

$S \backslash I$	i_1	i_2	i_3
1	4,0	2,0	—
2	1,1	3,0	—
3	—	1,1	2,1
4	—	2,1	—

Рис. 3. Таблица автомата.

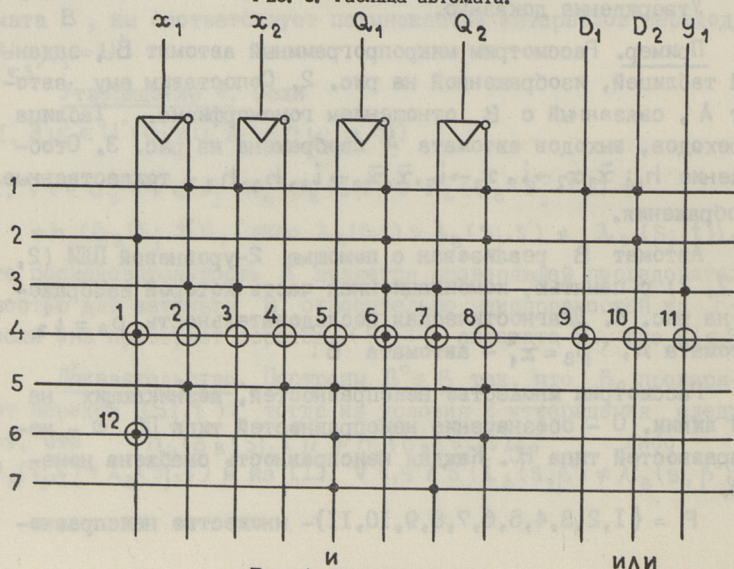


Рис. 4. Содержание ПЛН.

стей на 4-й линии, где $F_1 = \{9, 10, 11\}$, $F_2 = \{2, 5, 8\}$, $F_3 = \{1\}$, $F_4 = \{6, 7\}$, $F_5 = \{3, 4\}$. Рассмотрим для простоты одиночные неисправности в ПЛМ из множества F . В таблице (рис. 5) в строках 1-3 изображена контрольная последовательность автомата A : 2-я строка - контрольная последовательность α , 3-я - соответствующая α , выходная последовательность, 1-я - состояния, через которые проходит автомат A при подаче контрольной последовательности α , 4-я строка таблицы задает последовательность β , соответствующую последовательности α .

Далее, группы по 3 строки задают поведение автомата B в присутствии неисправности из $F_i, i = 1, 2, 3, 4, 5$. Слева, около фигурных скобок стоит номер неисправности, присутствующей в B . Первая из 3-х строк задает состояния автомата B' , вторая - отрезок последовательности β , обнаруживающей эту неисправность, 3-я - соответствующая β выходная последовательность автомата B' .

Но в схеме существуют и необнаруженные неисправности. Например, неисправность № 12 (рис. 4) необнаружена последовательностью β , для нее не выполняются условия, сформулированные в утверждении 3.

Отметим, что выбор кодировки h_4 состояний МПА может существенно повлиять на обнаружение неисправностей в ПЛМах, а следовательно, и на эффективность контрольной последовательности.

Заключение. Результаты данной работы подтвердили эффективность применения КП, построенной по методике [3], к реализациям МПА, комбинационная часть которых размещена на одной ПЛМ. Заметим, что при реализации сложных ДУУ становится необходимым использовать несколько ПЛМ (сетевая реализация исходного автомата). Развивая методику [2, 4], можно построить КП и для такой реализации. Эффективность ее применения требует дальнейших исследований.

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		
1	1	2	3	1	2	3	1	4	2	3	1	2	3	2	3	1	2	1	2	3	
2		i_2	i_2	i_2	i_2	i_2	i_1	i_2	i_2	i_2	i_2	i_2	i_3	i_2	i_2	i_2	i_1	i_2	i_2		
3		0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0	
4		x_1	x_1	x_1	x_1	x_1	$\bar{x}_1 x_2$	x_1	x_1	x_2	x_1	x_1	$\bar{x}_1 x_2$	x_1	x_1	x_1	$\bar{x}_1 x_2$	x_1	x_1		
9	5	00	01	00	01																
	6		x_1	x_1	x_1																
	7		0	0	0																
2	8	00	01	00	01																
	9		x_1	x_1	x_1																
	10		0	0	0																
1	11	00	01	10	00	01	10	00	11	01	10	00	01	10	01	10	00	01	10	00	01
	12		x_1	x_1	x_1	x_1	x_1	$\bar{x}_1 x_2$	x_1	x_1	x_1	x_1	$\bar{x}_1 \bar{x}_2$	x_1	x_1	x_1	$\bar{x}_1$	x_1	x_1		
	13		0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	1	0
6	14	00	01	10	00	01	10	00	11	11	11	11									
	15		x_1	x_1	x_1	x_1	x_1	$\bar{x}_1 x_2$	x_1	x_1	x_1										
	16		0	0	1	0	0	1	0	1	1	1									
3	17	00	01	00	11	01															
	18		x_1	$\bar{x}_1 \bar{x}_2$	x_1	x_1															
	19		0	0	0	1															
4	20	00	01	10	00	01	00	01													
	21		x_1	$\bar{x}_1 \bar{x}_2$	x_1	x_1	$\bar{x}_1 x_2$	x_1													
	22		0	0	1	0	0	0													

Рис. 5. Реакция МПА на КП.

Л и т е р а т у р а

1. H e n n i e F.C. Fault detecting experiments for sequential circuits // Proc. 5th Ann. Symp. Switching Circuit Theory and Logical Design. Princeton., 1964. P. 95-100.
2. К р у у с М. Контрольные эксперименты с микропрограммными автоматами. Тр. Таллиннск. политехн. ин-та. № 60I. 1985. С. 45-50.
3. Б е г о м е л о в А.М., Б а р а ш к о А.С., Г р у н с к и й И.С. Эксперименты с автоматами. Киев: Наукова Думка, 1974.
4. К е з з а л л и к А., К р у у с М. Метод нахождения диагностических и установочных последовательностей для сетей автоматов. Тр. Таллиннск. политехн. ин-та. № 626. 1986. С. 43-51.

L. Kaširova, M. Kruus, P. Ellervee

Kontrolljada süntees programmeeritavatel
loogikamaatriksitel realiseeritud juhtautomaatidele

Kokkuvõte

Käesolevas artiklis käsitletakse kontrolljadade sünteesi probleeme mikroprogrammautomaatidele, mis on realiseeritud programmeeritavatel loogikamaatriksitel. Esitatakse PLM-s esinevate rikete klassifikatsioon. Tõestatakse rida väiteid, millest järeldub, et informatsioon mikroprogramm-automaadi struktuuri kohta võimaldab lühendada automaadi kontrolljada.

L. Kashirova, M. Kruus, P. Ellervee

Derivation of Checking Sequence for Control
Automaton Being Realized as a Programmable Logic Array

Abstract

In this article we study the problem of creating the checking sequence for microprogram automaton realized as a programmable logic array (PLAs). The classes of faults for PLAs are introduced. It is shown that the knowledge of the structure of microprogram automaton can be utilized to design a simpler checking sequence. This sequence detects the faults of given classes.

ВЕРОЯТНОСТНОЕ ТЕСТИРОВАНИЕ ЦИФРОВЫХ СХЕМ
И АЛЬТЕРНАТИВНЫЕ ГРАФЫ

Сущность вероятностного тестирования цифровых схем заключается в подаче последовательностей случайных независимых двоичных сигналов на входы проверяемого объекта. Преимуществом вероятностного тестирования являются простота реализации и достаточная универсальность, т.е. применимость для широкого класса объектов. При таком подходе к тестированию не требуется предварительного проектирования тестов, что обычно связано с большими временными затратами, не требуется также больших объемов памяти для хранения предварительно сгенерированных тестовых последовательностей, длина которых для сложных цифровых схем может оказаться очень большой. Ввиду вышесказанного вероятностное, в частности, псевдослучайное тестирование широко используется как при внешнем тестировании, так и в самотестируемых цифровых устройствах.

Одним из существенных параметров систем тестирования цифровых схем является время, необходимое для выполнения процедур тестирования, которое однозначно определяется длиной теста. В случае детерминированных методов тестирования длина теста определяется требуемой полнотой теста, т.е. покрытием возможных неисправностей схемы. Определенные трудности возникают при определении необходимой длины теста при вероятностном тестировании. Наиболее точные оценки полноты теста в случае псевдослучайного тестирования могут быть получены при помощи моделирования, которое, однако, приводит к большим вычислительным затратам. Более подходящим и общим при вероятностном тестировании являются статистические методы оценки качества тестов, а также статистические методы определения длины теста.

Зависимость полноты покрытия неисправностей от количества тестовых наборов, представляющих собой случайные числа, может быть выражена экспонентой вида [1]

$$T = (1 - e^{-\lambda \log_{10} N}) \cdot 100\%, \quad (1)$$

где λ — постоянная величина, принимающая конкретное значение для каждого определенного типа цифровой схемы.

Аналогичная зависимость получена в [2]:

$$N = \frac{\ln[1 - p_d(N)]}{\ln[1 - p_d]}, \quad (2)$$

где N — число независимых случайных наборов, составляющих тест,

$p_d(N)$ — вероятность обнаружения неисправности при подаче N тестовых наборов и

p_d — вероятность необнаружения указанной неисправности.

Итак, если задана требуемая достоверность теста в виде вероятности $p_d(N)$, то по формуле (2) можно вычислить необходимую длину N вероятностного теста. При этом должна быть задана также вероятность p_d обнаружения заданной неисправности. Учитывая, что вероятности p_d для различных неисправностей различны, необходимо, обобщая выражение (2), вычислить длину теста по следующей формуле

$$N = m q x \left\{ \frac{\ln[1 - p_d(N)]}{\ln[1 - p_d]} \right\}, \quad (3)$$

где максимум берется по всему множеству проверяемых неисправностей.

Рассмотрим возможности оценки и вычисления вероятностей p_d обнаружения заданных неисправностей. Вероятность $p_d(z \equiv e)$, $e \in \{0, 1\}$, обнаружения константной неисправности $z \equiv e$ полюса z некоторой цифровой схемы представляет собой меру оценки эффективности вероятностного тестирования и характеризует сложность обнаружения данной неисправности. Условием обнаружения неисправности $z \equiv e$ на некотором выходе схемы y является [3]

$$(z \equiv \bar{e}) \frac{\partial y}{\partial z} = 1. \quad (4)$$

Из этого следует

$$p_d(z \equiv e) = p(z = \bar{e}) p\left(\frac{\partial y}{\partial z} = 1\right), \quad (5)$$

где $p(a=b)$ – вероятность выполнения условия $a=b$.

В работе [4] рассмотрены возможности вычисления вероятностей выходных сигналов и вероятностей чувствительности выходных сигналов к изменениям входных сигналов цифровой схемы, используя аппарат бинарных диаграмм решения. Но предложенные методы действительны лишь для случая независимых событий. Аналитическое вычисление значений вероятностей $p(z = \bar{e})$ и $p\left(\frac{\partial y}{\partial z} = 1\right)$ для цифровых схем общего вида значительно усложняется зависимостью событий появления неисправности и ее активизации и транспортировки на один из выходов схемы. Поэтому иногда пытаются вычислять лишь некоторые оценки вероятностей обнаружения неисправностей. Так, в [5] задача определения указанной вероятности сводится к задаче вычисления сигнальных вероятностей на эквивалентной цифровой схеме с учетом транспортировки влияния неисправности по некоторому одному активизированному пути. В случае, когда существует множество возможных путей транспортировки неисправности на выходы цифровой схемы, вычисленная по эквивалентной схеме вероятность будет служить нижней оценкой искомой вероятности обнаружения неисправности [5].

Рассмотрим здесь метод вычисления значений вероятностей обнаружения неисправностей, использующий способ представлений цифровых схем системой альтернативных графов [3]. При этом покажем, что имеется возможность путем незначительной модификации процедур вычисления добиться различной точности вероятностей – от вычисления нижних оценок до получения точных значений.

Представим цифровую схему с функцией $y = f(z)$ в виде альтернативного графа [3]

$$G = (M, \Gamma, Z),$$

где M – множество вершин графа,

Γ – отображение M в M ,

Z – множество булевых переменных (аргументов функции f).

Каждая вершина $m \in M$ взвешена некоторой переменной $z(m) \in Z$ (с инверсией или без нее). Согласно функции f

на графе определено отображение $\Gamma(m, z(m))$ по которому каждому значению $z(m)$ однозначно соответствует или некоторый последователь вершины m , $\Gamma(m, z(m)) \in \Gamma(m)$, или $\Gamma(m, z(m)) = \phi$. Любой набор $\underline{z}$ порождает транзитивное замыкание $\Gamma(m, z(m))$ в виде некоторого пути $\ell(\underline{z}) = \hat{\Gamma}(m^H, \underline{z})$ из начальной вершины графа m^H в некоторую терминальную вершину $m^T \in \ell(\underline{z})$, где $\Gamma(m^T, z(m^T)) = \phi$. Будем говорить, что $A\Gamma \subset G$ представляет функцию $y = f(\underline{z})$, если для любого набора $\underline{z}$, порождающего на G путь $\ell(\underline{z}) = \hat{\Gamma}(m^H, \underline{z})$ с терминальной вершиной $m^T \in \ell(\underline{z})$, имеет место $f(\underline{z}) = e(m^T)$.

Согласно [3] построение теста для вершины m , взвешенной переменной $z(m)$, сводится к активизации в $A\Gamma$ трех путей: начального пути $\ell^H(m^H, m)$ из m^H в m и двух конечных путей соответственно, $\ell^1(m^1, m^{T,1})$ из m^1 в $m^{T,1}$ и $\ell^0(m^0, m^{T,0})$ из m^0 в $m^{T,0}$. Здесь m^0 и m^1 , соответственно, нижний и правый последователи вершины m , а $m^{T,0}$ и $m^{T,1}$, терминальные вершины, где, соответственно, $z(m^{T,0}) = 0$ и $z(m^{T,1}) = 1$. Такой активизации соответствует выполнение условия (4). Так как в структурных $A\Gamma$ [3] каждой вершине соответствует один и только один путь в исходной схеме, то в результате этой процедуры получается тест также для некоторого конкретного пути в исходной схеме.

Из вышесказанного и из формулы (5) следует

$$p_d(z \equiv e) = p(z = \bar{e}) p(\ell^H) p(\ell^1) p(\ell^0), \quad (6)$$

где $z \equiv z(m)$, $p(\ell)$.

$\ell = \ell(m_1, m_2)$ — вероятность активизации одного из путей между вершинами m_1 и m_2 .

Вычисление вероятностей $p(\ell)$ активизации путей ℓ происходит по следующей формуле

$$p(\ell) = \sum_i \left[\prod_{m \in M_i^1} p(z(m)) \prod_{m \in M_i^0} (1 - p(z(m))) \right], \quad (7)$$

где

$$M_i^1 = \{m \mid m \in \ell_i, z(m) = 1\},$$

$$M_i^0 = \{m \mid m \in \ell_i, z(m) = 0\},$$

а ℓ_i - один из путей между вершинами m_1 и m_2 .

В формуле (7) происходит суммирование по всем возможным путям ℓ_i между m_1 и m_2 .

При раскрытии скобок в выражении (7) разрешены упрощения вида

$$p_i p_i = p_i. \quad (8)$$

Справедливость преобразования (8) следует из зависимости двух событий - появления двух сигналов, имеющих один и тот же источник. Здесь $p_i = p(z_i = 1)$.

В случае, когда вероятности p_i появления единичного значения для всех входных сигналов $z_i \in Z$ одинаковы, вычисление оценок вероятностей $p(\ell)$ по формуле (7) без учета зависимости сигналов достаточно просто - допускаются любые упрощения типа

$$p_i p_j = p^2 \quad (9)$$

(так как $p_i = p_j = p$) уже до раскрытия скобок. При этом, естественно, теряется информация для последующих замен типа (8), в результате чего точное вычисление вероятности $p(\ell)$ уже невозможно. Вычисление же точных значений $p(\ell)$ заметно усложняется, так как любые упрощения типа (8) или (9) разрешаются лишь после раскрытия скобок.

Трудность точного вычисления $p(\ell)$ по формуле (7) следует из того, что при генерации на АГ всех возможных путей

ℓ_i между заданными вершинами m_1 и m_2 не учитываются возможные противоречивости, в результате чего выражение для $p(\ell)$ получается громоздким. С другой стороны, на модели АГ легко строить алгоритм, который учитывал бы возможные противоречивости путей уже на стадии их перебора. Такой алгоритм приводит к следующей формуле

$$p\left(\frac{\partial y}{\partial z(m)} = 1\right) = \sum_i \left[\prod_{m \in M_i^1} p(z(m)) \prod_{m \in M_i^0} (1 - p(z(m))) \right], \quad (10)$$

где

$$M_i^1 = \{m \mid m \in \ell_i^H \cup \ell_i^0 \cup \ell_i^1, z(m) = 1\},$$

$$M_i^0 = \{m \mid m \in \ell_i^H \cup \ell_i^0 \cup \ell_i^1, z(m) = 0\}.$$

В формуле (10) происходит суммирование по всем возможным непротиворечивым комплектам путей $\ell_i^H, \ell_i^0, \ell_i^1$, соот-

ветственно, между вершинами m^H и m , m^0 и $m^{T,0}$, m^1 и $m^{T,1}$.

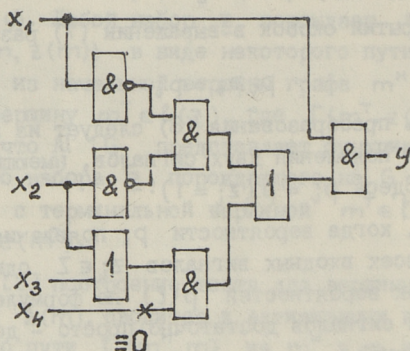


Рис. 1.

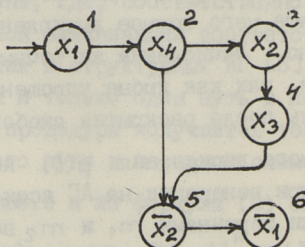


Рис. 2.

Пример. Рассмотрим схему на рис. 1 и соответствующий ей альтернативный граф на рис. 2. Вычисляем оценку и точное значение вероятности обнаружения неисправности $z_4 \equiv 0$ при условии, что вероятности появления единичных сигналов для всех входов схемы равны $z_i = 0,5$, $i = \overline{1,4}$.

1. Вычисляем нижнюю оценку вероятности $p_d(z_4 \equiv 0)$ по формуле (7). Получим

$$l^H = (1, 2): \quad p(l^H) = p_1,$$

$$l^1 = (3; 3, 4): \quad p(l^1) = p_2 + (1 - p_2) p_3,$$

$$l^0 = (5; 5, 6): \quad p(l^0) = p_2 + (1 - p_2) p_1,$$

$$\begin{aligned}
 p_d(z_4 \equiv 0) &= p_4 p_1 (p_2 + (1 - p_2) p_3) (p_2 + (1 - p_2) p_1) = \\
 &= p^2 (2p - p^2) (2p - p^2) = \\
 &= 4p^4 - 4p^5 + p^6 = 9/64.
 \end{aligned}$$

2. Вычисляем точное значение вероятности $p_d(z_4 \equiv 0)$ по формуле (7). Раскрывая сначала скобки и произведя затем упрощения вида (8) и (9), получим

$$\begin{aligned}
 p_d(z_4 \equiv 0) &= p_1 p_2 p_4 + p_1 p_3 p_4 - p_1 p_2 p_3 p_4 = \\
 &= 2p^3 - p^4 = 3/16.
 \end{aligned}$$

3. Вычисляем точное значение вероятности $p_d(z_4 \equiv 0)$ по формуле (10). Получим

$$l_1 = (l_1^H \cup l_1^1 \cup l_1^0) = (1, 2 \cup 3 \cup 5): p(l_1) = p_1 p_2.$$

$$l_2 = (l_2^H \cup l_2^1 \cup l_2^0) = (1, 2 \cup 3, 4 \cup 5, 6): p(l_2) = p_1 (1 - p_2) p_3.$$

$$\begin{aligned}
 p_d(z_4 \equiv 0) &= p_4 (p_1 p_2 + p_1 (1 - p_2) p_3) = \\
 &= p_1 p_2 p_4 + p_1 p_3 p_4 - p_1 p_2 p_3 p_4 = \\
 &= 2p^3 - p^4 = 3/16.
 \end{aligned}$$

Л и т е р а т у р а

1. Williams T.W. Test length in a self-testing environment // IEEE Design & Test of Computers, 1985, April. P. 59-63.

2. Я р м о л и к В.Н. Контроль и диагностика цифровых узлов ЭЕМ. Минск: Наука и техника, 1988. 240 с.

3. У б а р Р.Р. Альтернативные графы и техническая диагностика дискретных объектов // Электронная техника. Сер. 8, вып. 5. 1988.

4. A k e r s S.B. Probabilistic techniques for test generation from functional descriptions // IEEE Test Conference, 1979. P. 113-116.

5. S a v i r J., D i t l o w G.S., B a r d e l l P.H. Random pattern testability // IEEE Trans. on Computers. 1984. N 1. P. 79-90.

Numbriliste skeemide tõenäosuslik testimine ja
alternatiivsed graafid

Kokkuvõte

On välja töötatud uus meetod rikete avastamise tõenäosuste arvutamiseks numbrilistes skeemides, mida testitakse pseudojuhuslike testidega. Meetod põhineb numbriliste skeemide esitamisel alternatiivsete graafidena. Uus meetod võimaldab arvutusprotseduuride lihtsa modifitseerimise abil saavutada erinevaid tõenäosuste täpsusi - alates nende alumiste piiride hinnangutest kuni täpsete väärtusteni.

R. Ubar

Random Testing of Digital Circuits and
Alternative Graphs

Abstract

A new method for probabilistic calculation of the fault detectability in digital devices tested by pseudo-random test vectors is developed. The method is based on describing digital circuits by alternative graphs. It is possible to achieve different estimations for fault detection probabilities - from estimations of their lower bounds up to their exact values, by simple modification of the calculation procedures.

К. Григорьева

АЛГОРИТМ ОЦЕНКИ КАЧЕСТВА ВЕРОЯТНОСТНОГО
ТЕСТИРОВАНИЯ ЦИФРОВЫХ СХЕМ

Предлагаемый в [I] метод вычисления значений вероятностей обнаружения неисправностей в цифровых схемах при тестировании их случайными (или псевдослучайными) тестовыми наборами позволяет при помощи несложных алгебраических операций получать оценки такой вероятности с различной точностью в зависимости от допускаемых упрощений.

Простота и однородность вычислений при использовании этого метода вызывает естественное желание создания единого алгоритма, позволяющего совместить возможность получения точной оценки вероятности с упрощенным способом ее расчета, причем таким, чтобы упрощения учитывались непосредственно в процессе вычисления. Преимуществом такого алгоритма является также и возможность прервать тестирование по истечении отведенного ему лимита времени.

Для достижения поставленной таким образом цели предлагается генерировать псевдослучайные тесты для объекта диагноза посредством активизации для каждой тестируемой вершины трех путей $\ell^H(m^H, m)$, $\ell^1(m, m^{T,1})$ и $\ell^0(m, m^{T,0})$, определенных в [I]. Заметим, однако, что в общем случае терминальные вершины $m^{T,1}$ и $m^{T,0}$ должны удовлетворять условию

$$z(m^{T,1}) \beta(m^{T,1}) \neq z(m^{T,0}) \beta(m^{T,0}), \quad (I)$$

где $\beta(m^{T,i})$ — признак инвертирования сигнала $z(m^{T,i})$ в вершине $m^{T,i}$.

Активизацию этих путей будем осуществлять путем сканирования их на модели альтернативных графов (АГ) до тех пор,

пока путь ℓ^n не окончится вершиной m , ℓ^1 — некоторой терминальной вершиной $m^{T,1}$ и ℓ^0 — терминальной вершиной $m^{T,0}$, направление выхода из которой противоположно направлению выхода из $m^{T,1}$. При этом для каждого пути ℓ^n сканируются все возможные пути ℓ^1 , а для них, в свою очередь, все возможные пути ℓ^0 .

Таким образом, при реализации каждого тестового набора осуществляется разбиение множества всех входных переменных Z на взаимно-ортогональные подмножества

$$Z = Z^n \cup Z^1 \cup Z^0 \cup Z^x, \quad (2)$$

где Z^n, Z^1 и Z^0 — множества переменных, значения которых определяются, соответственно, при генерировании путей ℓ^n, ℓ^1 и ℓ^0 .

Z^x — множество переменных, значения которых на данном тестовом наборе не определены.

При этом во множества Z^n, Z^1 и Z^0 заносятся переменные, получающие значение "1" в процессе движения по графу, и исключаются из них при освобождении их от участия в тестовом наборе. При изменении же значений с "1" на "0" переменные сохраняются в соответствующих множествах. Таким образом, множества Z^n, Z^1 и Z^0 упорядочены по принципу стека.

Каждому текущему j -му тестовому набору соответствует текущее значение вероятности обнаружения им неисправности $z(m) \equiv e$ в вершине m вида

$$p_{d,j}(z(m) \equiv e) = p(z(m) = \bar{e}) p(\ell_j^n) p(\ell_j^1) p(\ell_j^0), \quad (3)$$

где

$$p(\ell_j^i) = \prod_{k: z_k \in Z^i} p_k^{\alpha_k}, \quad i = n, 1, 0, \alpha_k \in \{0, 1\} \quad (4)$$

и

$$p_k^{\alpha_k}: \begin{cases} p_k^0 = p(z_k = 0) = p_k, \\ p_k^1 = p(z_k = 1) = 1 - p_k. \end{cases} \quad (5)$$

Такое введение второго индекса при вероятности p_k обусловлено тем, что мы должны учитывать отсутствие (или наличие) значений как типа 0, так и 1.

Результирующая оценка вероятности

$$p_d(z(m)=e) = \sum_j p_{d,j}(z(m) \equiv e). \quad (6)$$

В соответствии с формулой (6) целесообразно накапливать искомую оценку вероятности обнаружения неисправности в заданной вершине графа пошагово, прибавляя к ней каждое очередное текущее значение p_j и обрывая этот процесс по выполнении заданного условия (истечение лимита времени).

В свою очередь текущее значение искомой вероятности также целесообразно накапливать в процессе сканирования:

- умножая ее на p_k^1 при установлении значения $z_k = 1$ и включении z_k в одно из множеств Z^i ($i = 0, 1, n$);
- умножая ее на p_k^0 / p_k^1 при изменении значения z_k с "1" на "0";
- умножая ее на $1/p_k^0$ при изменении значения z_k с "0" на неопределенное и исключении z_k из соответствующего множества Z^i ($i = 0, 1, n$).

В результате сокращается количество выполняемых арифметических операций и исключается повторный учет вероятностей оценки тех же событий.

Исходя из этих соображений, получен приведенный ниже алгоритм вычисления точной оценки вероятности обнаружения одиночной неисправности типа $\equiv 0$ или $\equiv 1$ в заданной вершине графа. При описании алгоритма приняты следующие обозначения:

- m^h - начальная вершина графа;
- m^t - текущая вершина графа;
- m - тестируемая вершина графа;
- $\bar{e}$ - неисправность в вершине m , вероятность обнаружения которой вычисляется;
- i - индекс множества Z^i , в которое включаются/исключаются переменные в процессе сканирования одноименного пути ℓ^i (для удобства организации рекурсии при сканировании путей индекс n заменен на 2);
- $\text{succ}(m^k) \mid \alpha$ - последователь вершины z^k на графе в направлении α ;
- V^i - направление выхода из пути ℓ^i ;

$\beta(m^K)$ - признак инвертирования сигнала $z(m^K)$ в вершине m^K ;

p_d - суммарная оценка вероятности обнаружения заданной неисправности в вершине m ;

p_j - вероятность обнаружения заданной неисправности в вершине m текущим тестовым набором;

$\text{least}(Z^i)$ - последний элемент множества Z^i .

Алгоритм

- 0°. $Z^2, Z^1, Z^0 := \emptyset, Z^X := Z, p_d := 0, p_j := p^e(z(m)), m^t := m^X, i := 2$.
- 1°. Если $m^t = m$, то идти к п. 15°.
- 2°. Если $z(m^t) = z(m)$, то: если $i = 2$, то $s := \text{succ}(m^t) | e$; иначе $s := \text{succ}(m^t) | i$; идти к п. 5°.
- 3°. Если $z(m^t) = x$, то: $Z^i := Z^i \cup z(m^t), z(m^t) := 1, p_j := p_j^f(z(m^t))$.
- 4°. $s := \text{succ}(m^t) | z(m^t)$.
- 5°. Если s - вершина графа, то: $m^t := s$, идти к п. 1°.
- 6°. Если $i = 2$, то идти к п. 11°.
- 7°. $V^i := z(m^t) \beta(m^t)$.
- 8°. Если $i = 1$, то: $i := 0, s := \text{succ}(m) | 0$, идти к п. 5°.
- 9°. Если $V^1 \neq V^0$, то $p_d := p_d + p_j$.
- 10°. Если условие длительности выполнено, то идти к п. 16°.
- 11°. Если $z(m^t) = 1$, то: $z(m^t) := 0, p_j := p_j p^u(z(m^t)) / p^1(z(m^t))$, идти к п. 2°.
- 12°. $Z^i := Z^i \setminus z(m^t), z(m^t) := x, p_j := p_j / p^0(z(m^t))$.
- 13°. Если $Z^i = \emptyset$, то: если $i = 2$, то идти к п. 16°; иначе: $i := -1$, идти к п. 13°.
- 14°. $m^t := \text{least}(Z^i)$, идти к п. 11°.
- 15°. $i := 1, s := \text{succ}(m) | 1$, идти к п. 5°.
- 16°. Конец.

П. 2° алгоритма определяет направление движения по графу в тех случаях, когда вершины, взвешенные той же перемен-

ной, что и тестируемая вершина m , встречаются, соответственно, на пути l^n или на путях l^1 и l^0 . При этом, в первом случае оно определено типом неисправности, вероятность обнаружения которой вычисляется, а в двух последних - индексом пути.

Для иллюстрации работы алгоритма рассмотрим схему и соответствующую ей модель АГ, приведенные на рис. 1 и 2,

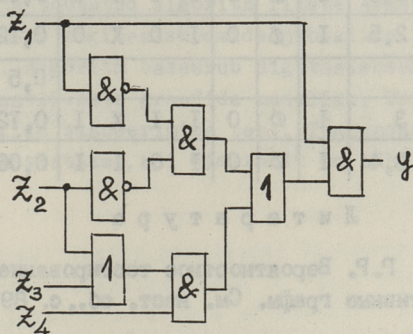


Рис. 1.

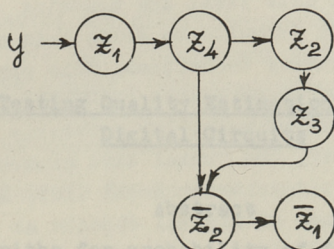


Рис. 2.

предполагая также, что все вероятности одинаковы и равны $p = 0,5$. Результаты изображены в виде таблицы I, где отразим только те из сканируемых путей, которые соответствуют генерируемым наборам (т.е. непротиворечивые пути)

Т а б л и ц а I

m	$\equiv \bar{e}$	Z^H	Z^I	V^I	Z^0	V^0	z_1	z_2	z_3	z_4	p_j	p_d
I	$\equiv 0$										0,5	0
		$\emptyset$	2,3	I	$\emptyset$	0	I	I	X	I	0,125	0,125
		$\emptyset$	2,3	I	$\emptyset$	0	I	0	I	I	0,0625	0,1875
		$\emptyset$	2,3	I	$\emptyset$	0	I	0	0	I	0,0625	0,25
		$\emptyset$	2,5	I	$\emptyset$	0	I	0	X	0	0,125	0,375
2	$\equiv 0$										0,5	0
		I	3	I	$\emptyset$	0	I	I	X	I	0,125	0,125
		I	3,4	I	$\emptyset$	0	I	0	I	I	0,0625	0,1875

Л и т е р а т у р а

И. У б а р Р.Р. Вероятностное тестирование цифровых схем и альтернативные графы. См. наст. сб., с. 89.

Digitaalskeemide juhusliku testimise
kvaliteedi hinnangu algoritm

Kokkuvõte

On välja töötatud algoritm rikete avastamise tõenäosuse hindamiseks digitaalskeemide juhuslike testidega testimise puhul. Algoritm baseerub digitaalskeemide kirjeldamisel alternatiivsete graafide mudeliga. Testid genereeritakse graafide skaneerimise teel. Tõenäosuse arvutamise protsess katkestatakse, kui ajalimiit on läbi.

K. Grigoryeva

A Random Testing Quality Estimation Algorithm for
Digital Circuits

Abstract

An algorithm for probability of fault detection in digital circuits by random testing is developed. The algorithm is based on describing digital circuits by alternative graphs and test generation by the scanning of them. It is possible to break the probability calculation process when the time limit has been spent.

AN AUTOMATIC TEST GENERATION SYSTEM
FOR MICROPROCESSOR VLSI

1. Introduction

The development of integrated circuit technology has resulted in a wide range of applications for microprocessors. Testing of microprocessors is a difficult problem because of object complexity. The problem is more serious for users due to lack of information on internal implementation of microprocessors and other VLSI chips. The classical techniques which specify the actual implementation of the digital system on the gate level are becoming practically unfeasible. A possible solution to solve this problem is to use the functional testing techniques. This paper presents an automatic test generation system (PROTEST) for microprocessor VLSI. PROTEST is meant for automatic test synthesis on the functional level. The formal test generation process is based on the Alternative Graph Model (AG), which permits to describe a wide class of digital objects at the common manner and to use united test generating strategy [1].

AG serves as a digital object's functional specification for automatic test generating purposes. It can also facilitate the expression of structural assumption about the system which is vital for practical test generation.

PROTEST is written in LISP and C programming languages. The first version of the test generation system was developed on the CM-1420 minicomputer under UNIX operating system. Currently its new version is implemented on the IBM PC XT, IBM PC AT and IBM PS/2 personal computers.

Section 2 briefly describes the key features of PROTEST system. The AG model of microprocessor VLSI for practical test generation is given in Section 3. Section 4 describes the test generating process.

2. Key features of PROTEST

The test generation system has the following capabilities:

- High-level object description language:

The predicate expressions language fits well for introducing functional descriptions like those typically found in microprocessor manufacturer's data-sheets. There are powerful macro facilities together with preprocessor options, which make language self-documentary and easily tunable to a wide set of possible objects.

- Automatic test generation on the basis of object's functional description:

The test generation is based on mathematical model, which is automatically generated from digital object's (DO) functional description. The DO like microprocessor is modelled by a set of AG [2]. Each AG describes the behaviour of some functional component of object under test at the programming and microprogramming levels. The first stage is abstract test program and test data patterns generation. The second stage transforms the abstract test programs to the tester-specific language. The test programs can be synthesized for different testers.

- Test generation on the basis of user defined test information:

The test data patterns (local tests for functional components) used during test generating can be defined automatically or manually. In this case, manually defined local tests will be used.

3. DO's model

The object under test is represented in the form of AG model

$$S_{AG} = (Z, G),$$

where G is a set of AG and Z is a set of DO's variables. Each AG is an oriented graph, that represents the behaviour of some functional component (module or part) at the programming and microprogramming levels. There are three types of nodes in AG model: X-node, I-node, and F-node.

a) X-node: This type of node is labeled with boolean variables and models logical conditions (flags). X-node has two output paths.

b) I-node: This type of node is labeled with n -bit words and is interpreted as instruction or microinstruction register with decoder. I-node has n output paths.

c) F-node: This type of node is labeled with function and represents DO's operation. The function's arguments can be register variables and constants.

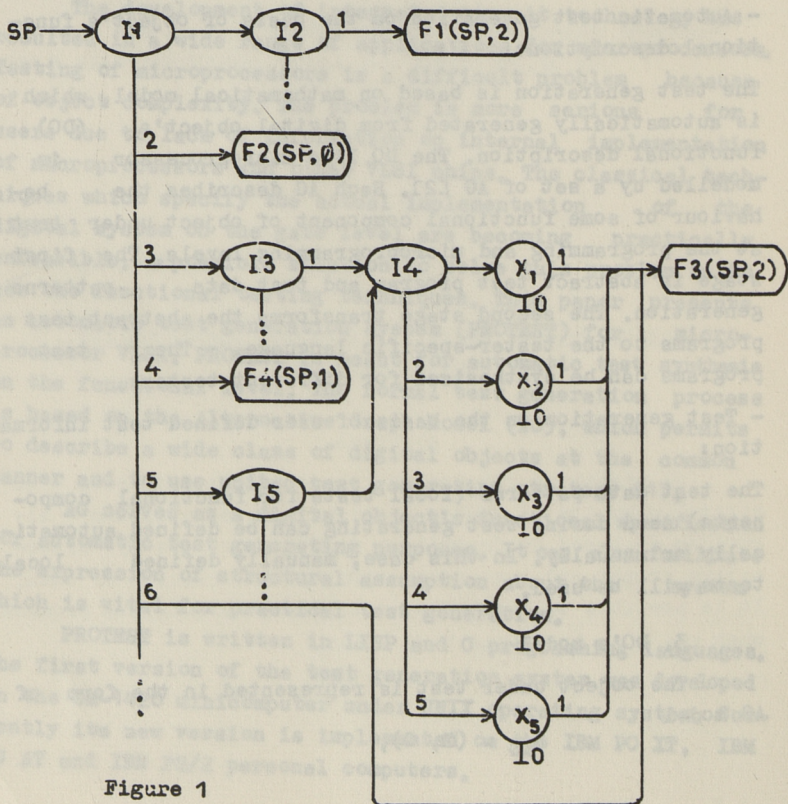


Figure 1

NODETAB (Nodes Table)

NUM	TYPE	...	DESCRIPTOR
10	F		FUN
100	X	...	
101	X	...	BIN
		...	
135	I	...	MUL
150	I	...	

(F-node Descriptor)

ADDR	OP1	OP2
------	-----	-----

Function's address in memory

VARTAB

VARTAB

VARX SUCC1 SUCC2 (X-node Descriptor)

VARI	COUNT	POINT
------	-------	-------

(I-node Descriptor)

SUBDTAB (Sub-descriptions Table)

VAL	V-COUNT	SUCC

VARTAB (Variables Table)

TYPE	FORM	...	INDX	...
INT	BYTE	...	135	...
INT	BIT	...	101	...

VALTAB (Values Table)

MVI, MOV, LXI, DAA,
..., XRA, ANI, ORI

Figure 2

The object under test is viewed as made up of two parts: an operational unit (OU) and a control unit (CU). On the AG model OU is described by the set of I- and X-nodes. Figure 1 presents AG fragment for stack pointer SP (microprocessor INTEL 8080), where I_i represent instruction code fields and X_i describe flags associated with instruction execution. F_i represent ALU operations. Figure 2 presents a simplified table structure of DO's AG model, with following basic tables:

- 1) NODETAB (Nodes Table),
- 2) VARTAB (Variables Table),
- 3) SUBDTAB (Subdescriptors Table) and
- 4) VALTAB (Values Table).

Each node has a record in NODETAB. For different node types (I, X and F) there are different descriptor structures in NODETAB. FUN (F-node Descriptor) includes information about node's function (address in operational memory and operands addresses in VARTAB). BIN (X-node Descriptor) includes information about X-node's variable and successors of this node. MUL (I-node Descriptor) has a hierarchical structure and it includes information about node's variable and output paths. There is a record for I-node's each output path in SUBTAB. Each AG variable has a record in VARTAB.

4. The test generation process

The test generation process is illustrated in Figure 3.

The test generation task is held up on AG model of DO [3]. There are also used the library of emulation functions and test program segments files. The emulation functions are used for F-nodes modelling. They describe functional behaviour of a procedures coded in C programming language and are organized in the object code library. Figure 4 presents the emulation function for a instruction DAA (microprocessor INTEL 8080).

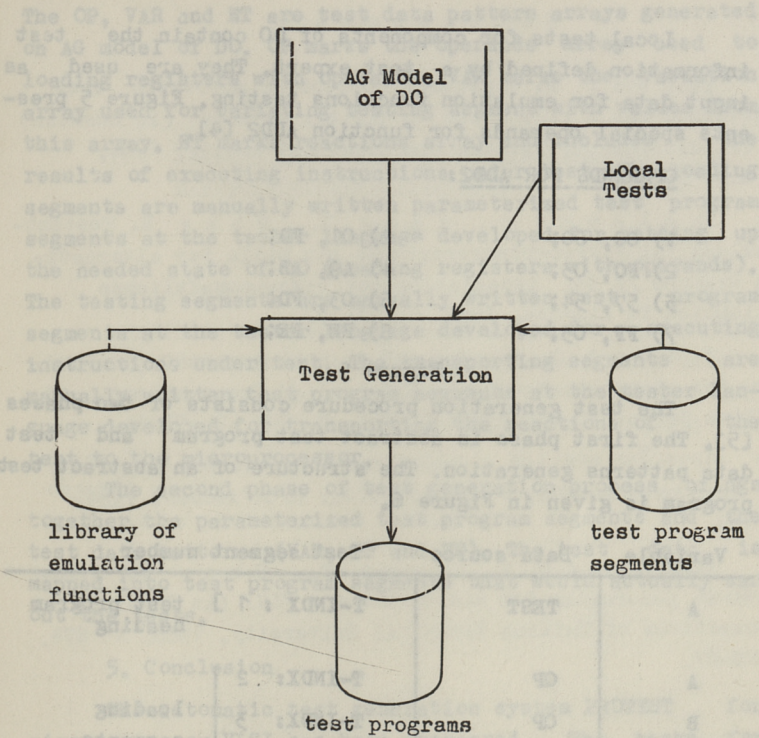


Figure 3

```
unsigned DAA (A, AC, CY)
```

```
{
```

```
  if (AC || (A & OXOF) > 9)
```

```
    a += 6;
```

```
  return (A);
```

```
}
```

Figure 4

Local tests for components of DO contain the test information defined by a test expert. They are used as input data for emulation functions testing. Figure 5 presents special operands for function ADD2 [4].

OPERANDS FOR ADD2:

- | | |
|------------|------------|
| 1) 00, 00; | 2) 01, FD; |
| 2) FC, 03; | 4) A9, AB; |
| 5) 57, 54; | 6) 03, FD; |
| 7) FF, 03; | 8) FE, FE; |

Figure 5

The test generation procedure consists of two phases [5]. The first phase is abstract test program and test data patterns generation. The structure of an abstract test program is given in Figure 6.

Variable	Data source	Test segment number
A	TEST	T-INDX : 1] test program healing
A	OP	T-INDX: 2]
B	OP	T-INDX: 3] loading segments
	⋮	
F-cy	OP	T-INDX: 23]
IN1	VAR	T-INDX: 35] testing segment
A	ET	T-INDX: 47] transporting segment

REGISTERS:

OTK	=		] the tester control register's names and values
?			
PT	=		

Figure 6

The OP, VAR and ET are test data pattern arrays generated on AG model of DO. OP marks the operands array used to loading registers with operands. VAR marks the variation array used for varying testing segment with values from this array. ET marks reactions array and includes the results of executing instructions under test. The loading segments are manually written parameterized test program segments at the tester language developed for setting up the needed state of DO (loading registers with operands). The testing segments are manually written test program segments at the tester language developed for executing instructions under test. The transporting segments are manually written test program segments at the tester language developed for transporting the reactions of the test to the microprocessor.

The second phase of test generation process brings together the parameterized test program segments and the test data patterns (VAR, OP and ET). The test data is mapped into test program segments that would actually carry out the tests.

5. Conclusion

The automatic test generation system PROTEST for microprocessor VLSI has been presented. The tests for digital system are generated on the functional level and test generation process is based on Alternative Graph Model. Special features of microprocessors, like the possibility of the straightforward solving of the controllability and observability tasks, are exploited to find the suitable correlation of manual and automatic parts in test generation process. The test generation is performed in two stages and test programs can be synthesized for different testers.

The first experiments with PROTEST were carried out. Test generation for accumulator of the microprocessor INTEL 8080 took one minute of CPU time running on a IMB PS/2 Model 50. Large scale experiments are currently being planned and the improvement of the system will continue.

References

1. U b a r R. A general concept of automatic test generation for a wide class of digital objects (in Russian) // Transactions of Tallinn Technical University. 1986. N 626. P. 70-92.
2. A l a n g o V., K o n t T., U b a r R. Testentwicklung für Mikroprozessorsysteme mit Hilfe der alternativen Graphen // 33. Internationales Wissenschaftliches Kolloquium TH Ilmenau. 1988. S. 11-14.
3. K o n t T. An approach to test generation automation for microprocessors (in Russian) // Transactions of Tallinn Technical University. 1988. N 674. P. 55-64.
4. G ä r t n e r A. Optimierte Selbsttest Programme für Mikroprozessoren. - Dissertation. RWTH Aachen, 1984. S. 138.
5. A l a n g o V., K o n t T., U b a r R. Test-program generating for microprocessor VLSI (in Russian) // Transactions of Tallinn Technical University. 1988. N 674. P. 78-87.

T. Kont

Testide automaatgenereerimise süsteem
mikroprotsessorlülitustele

Kokkuvõte

Artiklis tutvustatakse testide genereerimise süsteemi PROTEST. PROTEST on ette nähtud funktsionaalsete testide automaatseks sünteesiks mikroprotsessorlülitustele. Testide automaatne genereerimine toimub diskreetset objekti kirjeldavatel alternatiivsetel graafidel. Testide sünteesil saab objekti teatud osade kontrollimiseks kasutada emuleerivaid funktsioone ja lokaalseid teste.

T. Kont

An Automatic Test Generation System
for Microprocessor VLSI

Abstract

Test generation system for microprocessor VLSI PROTEST has been presented in this paper. PROTEST is meant for automatic test synthesis on functional level. An outline of test generating with PROTEST has been presented. The formal test synthesis is held up on alternative graph model of digital object. There is also a possibility to use emulation functions and local tests for digital object's specific parts testing.

Содержание

1.	Э.Э. Велмре, М.С. Пикков. Измерение коэффициентов ударной ионизации в рп-переходах при очень низкой интенсивности размножения.....	3
2.	С. Орро, Т. Ранг. Специализированные базы данных для программных средств моделирования и анализа элементов твердотельной электроники...	20
3.	Б.Г. Гурьянов. Матрицы Фурье и обобщенная форма прямого произведения.....	25
4.	М.Э. Круус, К. Салум. Экспериментальное исследование алгоритмов синтеза контрольных последовательностей для управляющих автоматов..	33
5.	Б.Е. Беркман, А.Э. Казгаллик. Оптимизация декомпозиционного синтеза микропрограммных автоматов на основе инфоресурса.....	43
6.	Б.Е. Беркман. Декомпозиция микропрограммных автоматов в сети из автоматов Мили	67
7.	Л.Ф. Каширова, М.Э. Круус, П.С. Элларвез. Построение контрольной последовательности для управляющих автоматов, реализованных на программируемых логических матрицах.....	77
8.	Р.Р. Убар. Вероятностное тестирование цифровых схем и альтернативные графы.....	89
9.	К. Григорьева. Алгоритм оценки качества вероятностного тестирования цифровых схем.....	97
10.	An Automatic Test Generation System for Microprocessor VLSI.....	104

МАШИННОЕ ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ УСТРОЙСТВ
И СИСТЕМ

Электротехника и автоматика XXУП

УДК 621.382.2

Измерение коэффициентов ударной ионизации в рп-переходах при очень низкой интенсивности размножения. Велмре Э., Пикков М. - Труды Таллиннского технического университета. 1989, № 696, с. 3-19.

Предложен метод измерения коэффициентов ударной ионизации электронов и дырок в полупроводниках в режиме очень слабого размножения, когда $M \gg 1,002$. Метод основывается на снятии зависимости тока базы от напряжения на коллекторном переходе биполярной транзисторной структуры, работающей в нормальном активном режиме. Измерены коэффициенты ударной ионизации электронов и дырок в германии в кристаллографическом направлении $\langle 111 \rangle$ при $T = 300 \text{ K}$ в диапазоне напряженностей электрического поля от $9 \cdot 10^4$ до $1,6 \cdot 10^5 \text{ В} \cdot \text{см}^{-1}$.

Рисунков - 3, библи. наименований - 15.

УДК 621.382

Специализированные базы данных для программных средств моделирования и анализа элементов твердотельной электроники. Орро С., Ранг Т. - Труды Таллиннского технического университета. 1989, № 696, с. 20-24.

Описываются требования к специализированным базам данных (СБД). Выведены глобальные и специфические требования СБД. Описана СБД для ППП анализа биполярных полупроводниковых приборов и структур Шоттки.

Библи. наименований - 8.

УДК 512.8-681.3

Матрицы Фурье и обобщенная форма прямого произведения. Гурьянов Б. – Труды Таллиннского технического университета, 1989, № 696, с. 25-32.

В статье рассматривается алгоритм образования матриц Фурье на базе обобщенной формы прямого произведения и обобщенного ядра.

Такой алгоритм эффективней известных, в особенности при больших размерах матриц.

Библ. наименований – 6.

УДК 62-507

Экспериментальное исследование алгоритмов синтеза контрольных последовательностей для управляющих автоматов. Круус М., Салум К. – Труды Таллиннского технического университета. 1989, № 696, с. 33-42.

Рассмотрены проблемы построения контрольных последовательностей для дискретных управляющих устройств. Сравнены характеристики контрольных последовательностей (время их построения, полнота, длина), найденных на основе различных уровней представления управляющего устройства (микропрограммный автомат, сеть автоматов). Приведены результаты машинного эксперимента, подтверждающие эффективность используемых алгоритмов.

Библ. наименований – 7, таблиц 2, рисунков – 2.

УДК 62.507

Оптимизация декомпозиционного синтеза микропрограммных автоматов на основе инфоресурса. Беркман Б., Кэваллик А. – Труды Таллиннского технического университета, 1989, № 696, с. 43-66.

В статье разрабатываются процедуры декомпозиции микропрограммных автоматов на основе инфоресурса. Предложены алгоритмы глобальных и локальных трансформаций, в ходе которых производится оптимизация получаемых сетей автоматов. Для оценки промежуточных решений в ходе оптимиза-

ции используется информационная оценка – инфоресурс. Приведены результаты экспериментального исследования предложенных алгоритмов.

Рисунков – 5, таблиц – 2, библиографических наименований – 6

УДК 62.507

Декомпозиция микропрограммных автоматов в сети из автоматов Мили. Беркман Б. – Труды Таллиннского технического университета, 1989, № 696, с. 67–76.

В статье рассматривается проблема декомпозиции микропрограммных автоматов в сети из автоматов Мили по ортогональному множеству разбиений. Разработана оригинальная модель сети из микропрограммных автоматов Мили, для которой представлен результирующий автомат сети. Благодаря использованию предложенной модели сети получены реализации автоматов меньшим количеством ПЛМ по сравнению с другими методами.

Рисунков – 4, библиографических наименований – 6.

УДК 62.507

Построение контрольной последовательности для управляющих автоматов, реализованных на программируемых логических матрицах. Каширова Л., Круус М., Эллервез П. – Труды Таллиннского технического университета, 1989, № 696, с. 77–88.

В данной статье рассматривается проблема построения проверяющей последовательности для микропрограммного автомата, реализованного на программируемых логических матрицах (ПЛМ). Рассматриваются неисправности, встречающиеся в ПЛМ, приводится их классификация. Доказываются утверждения, из которых следует, что знание структуры микропрограммного автомата позволяет построить более простую контрольную последовательность

Рисунков – 5, библиографических наименований – 4.

УДК 681.32

Вероятностное тестирование цифровых схем и альтернативные графы. Убар Р. - Труды Таллиннского технического университета. 1989, № 696, с. 89-96.

Разработан новый подход к вычислению значений вероятностей обнаружения неисправностей в цифровых схемах псевдослучайными тестами, основывающийся на представлении цифровых схем альтернативными графами. Разработанный метод позволяет путем незначительной модификации процедур вычисления добиться различной точности вероятностей от вычисления нижних границ оценок до получения точных значений.

Рисунков - 2, библиографических наименований - 5.

УДК 681.32

Алгоритм оценки качества вероятностного тестирования цифровых схем. Григорьева К. - Труды Таллиннского технического университета. 1989, № 696, с. 97-103.

Разработан алгоритм оценки вероятности обнаружения неисправностей цифровых схем при тестировании их случайными тестами. Алгоритм базируется на описании цифровых схем альтернативными графами и генерировании тестов путем сканирования их. Процесс вычисления вероятности может быть прерван, когда достигнуто желаемое значение вероятности или когда исчерпан лимит времени.

Рисунков - 2, таблиц - 1, библиографических наименований - 1.

УДК 681.32

Система автоматизированного синтеза тестов для микропроцессорных БИС. Коньт Т. - Труды Таллиннского технического университета. 1989, № 696, с. 104-113.

В данной статье рассматривается система автоматизированного синтеза тестов для микропроцессорных БИС ПРОТЕСТ. Система ПРОТЕСТ предназначена для генерации тестов на функциональном уровне. Автоматическое генерирование тестов производится на основе модели дискретного объекта в виде альтернативных графов. При этом могут быть использованы файлы эмулирующих функций и локальных тестов для определенных частей объекта.

Рисунков - 6, библиографических наименований - 6.

Цена руб. 1.20

EESTI AKADEEMILINE RAAMATUKOGU



1 0200 00082472 6