

Лр.б.7  
601

ISSN 0136-3549

3134-3823

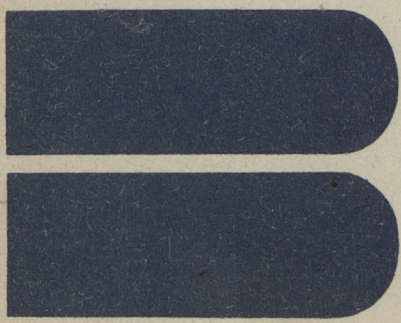
TALLINNA  
POLÜTEHNILISE INSTITUUDI  
TOIMETISED

601

ТРУДЫ ТАЛЛИНСКОГО  
ПОЛИТЕХНИЧЕСКОГО  
ИНСТИТУТА

**ТПИ**  
**'85**

МЕТОДЫ СИНТЕЗА  
И ДИАГНОСТИРОВАНИЯ  
ЦИФРОВЫХ СХЕМ





Гр.6.

601

**ТПИ  
'85**

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED

ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

УДК 62-507+681.142.3+681.32



МЕТОДЫ  
СИНТЕЗА  
И  
ДИАГНОСТИРОВАНИЯ  
ЦИФРОВЫХ  
СХЕМ

Электротехника и автоматика XXУП

Таллин 1985



ТАЛЛИНСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ  
Труды ТПИ № 601

МЕТОДЫ СИНТЕЗА И ДИАГНОСТИРОВАНИЯ  
ЦИФРОВЫХ СХЕМ

Электротехника и автоматика ХХУ11

На русском языке

Редактор О. Пикков

Техн. ред. В. Ранник

Сборник утвержден коллегией Трудов ТПИ 26.03.85

Подписано к печати 20.11.85

МВ-08768

Формат 60x90/16. Печ. л. 6,0 + 0,5. Уч.-изд. л. 5,2

Тираж 300

Зак. № 674

Цена 80 коп.

Таллинский политехнический институт,

200108 Таллин, Эхитаяте теэ, 5

Ротапринт ТПИ, 200006 Таллин, ул. Коскла, 2/9

© Таллинский политехнический институт, 1985

## ДЕКОМПОЗИЦИЯ КОНЕЧНЫХ АВТОМАТОВ НА ОСНОВЕ ИНФОРМАЦИОННОГО КРИТЕРИЯ

1. Введение. Проблема нахождения оптимальной декомпозиции является одной из труднейших в процессе синтеза дискретных управляющих устройств. Наиболее перспективным, с точки зрения решения практических задач декомпозиционного синтеза, является на сегодняшний день безусловно т.н. парный метод декомпозиции [1]. Но несмотря на свои достоинства в рамках этого алгебраического метода декомпозиции конечных автоматов мы лишены возможности заранее оценить альтернативные варианты и тем самым направить процесс синтеза. Особенно относится это к тем случаям, когда результат декомпозиции зависит от выбора элементов из множества несравнимых разбиений [1].

В данной работе предложен способ усовершенствования парного метода декомпозиции путем введения количественных оценок в виде энтропии для разбиений и пар разбиений. Эти оценки используются в качестве целевых функций при синтезе.

2. Основные понятия. Вопросы, связанные с реализацией выходной функции конечного автомата, в данной работе не рассматриваются. Поэтому определим конечный автомат как тройку

$$A = \langle I, S, \delta \rangle,$$

где  $I$  - входной алфавит;

$S$  - множество внутренних состояний;

$\delta: S \times I \rightarrow S$  - функция переходов.

В этом случае целесообразно и понятие сети автоматов [2] определить упрощенно как тройку

$$N = \langle \mathcal{U}, I, \Phi \rangle,$$

где

$$\mathcal{U} = \{ A_i = \langle I_i, S_i, \delta_i \rangle \mid i \in J \}; \quad I_i = I \times I'_i \}$$

- множество компонентных автоматов,

$J = \{1, \dots, n\}$  и  $I'_i$  - внутренний входной алфавит ( $i$ -го компонентного автомата),

$\Phi = \{ \varphi_i : \prod_{j \in J_i} A_j \rightarrow I'_i \mid i \in J; J_j \subset J \}$  - множество функций соединения.

Число  $n$  называется рангом сети  $N$ . Каждой сети  $N$  можно сопоставить функционально эквивалентный ей автомат  $A_N$ . Сеть  $N$  называется декомпозицией автомата  $A$ , если и только если у автомата  $A_N$  имеется подавтомат  $A'_N$ , изоморфный автомату  $A$  [2].

Разбиение произвольного конечного множества  $X = \{x_1, x_2, \dots, x_m\}$  на блоки  $B_i^{(1)}, B_i^{(2)}, \dots, B_i^{(\alpha)}, \dots, B_i^{(m)}$  обозначим через  $\pi_i(X)$ . Нулевое (поэлементное) разбиение обозначим через  $O_X$ , а единичное (одноблочное) через  $1_X$ . Для любого подмножества  $X' \subset X$  определим его вес  $q(X')$  как отношение  $q(X') = \|X'\| / \|X\|$ . Сужением разбиения  $\pi_i(X)$  на  $X' \subset X$  будем называть разбиение

$$\bar{\pi}_i(X') \stackrel{\text{df}}{=} \{ B_i^{(\alpha)} \cap X' \mid B_i^{(\alpha)} \in \pi_i; B_i^{(\alpha)} \cap X' \neq \emptyset \}.$$

Запись  $x_i \equiv x_j(\pi)$  означает, что элементы  $x_i$  и  $x_j$  находятся в одном и том же блоке разбиения  $\pi$ .

Разбиения  $\pi_i(X')$  и  $\pi_j(X'')$  будем называть эквивалентными (обозначаем  $\pi_i(X') \equiv \pi_j(X'')$ ), если и только если существует биекция  $\varphi: \pi_i(X') \rightarrow \pi_j(X'')$  такая, что  $(\forall B_i^{(\alpha)} \in \pi_i) (\|B_i^{(\alpha)}\| / \|\varphi(B_i^{(\alpha)})\| = \text{const})$ . Для любых разбиений  $\pi_i(X)$  и  $\pi_j(X)$  примем, что

$$\pi_i \cdot \pi_j \stackrel{\text{df}}{=} \{ B_i^{(\alpha)} \cap B_j^{(\beta)} \mid B_i^{(\alpha)} \in \pi_i; B_j^{(\beta)} \in \pi_j; B_i^{(\alpha)} \cap B_j^{(\beta)} \neq \emptyset \}$$

и  $\pi_i + \pi_j \stackrel{\text{df}}{=} \prod \{ \pi_k(X) \mid \pi_k \geq \pi_i, \pi_j \}, \text{ где } \pi_i \leq \pi_j \Leftrightarrow \pi_i \cdot \pi_j = \pi_i.$

Всевозможные разбиения на  $X$  образуют относительно операций умножения и сложения полумодулярную решетку [3], которую будем обозначать через  $\mathcal{L}_{\pi}(X)$ . Определим для разбиения  $\pi_i \in \mathcal{L}_{\pi}(X)$  энтропию:

$$H(\pi_i) \stackrel{\text{df}}{=} - \sum_{\alpha=1}^{m_i} q(B_i^{(\alpha)}) \ln q(B_i^{(\alpha)}).$$

Примем, что условная энтропия  $H(\pi_j / \pi_i) \stackrel{\text{df}}{=} H(\pi_i \cdot \pi_j) - H(\pi_i)$ .

Рангом системы разбиений  $P(X) = \{ \pi_i(X) \mid i \in J \}$  называем число  $w \stackrel{\text{df}}{=} \|J\|$ . Произведение всех разбиений из  $P(X)$  обозначаем  $m(P) \stackrel{\text{df}}{=} \prod_{i \in J} \pi_i$ . Систему разбиений  $P(X)$  называем полной, если и только если  $m(P) = O_X$ . Принимаем, что система разбиений

ний  $P'(X)$  покрывает систему  $P(X)$ , если и только если для любой  $\pi_i \in P(X)$  найдется  $\pi'_i \in P'(X)$  такое, что  $\pi'_i \geq \pi_i$ .

Любую пару разбиений  $\rho_i(X) \stackrel{\text{Df}}{=} \langle \pi_{i1}(X), \pi_{i2}(X) \rangle$  будем называть каналом на  $X$ . Энтропия канала определяется как  $H(\langle \pi_i, \bar{\pi}_i \rangle) \stackrel{\text{Df}}{=} H(\bar{\pi}_i / \pi_i)$ . Операции умножения и сложения определяются для каналов следующим образом:

$$\langle \pi_h, \bar{\pi}_i \rangle \cdot \langle \pi_j, \bar{\pi}_k \rangle \stackrel{\text{Df}}{=} \langle \pi_h \cdot \pi_j, \bar{\pi}_i \cdot \bar{\pi}_k \rangle,$$

$$\langle \pi_h, \bar{\pi}_i \rangle + \langle \pi_j, \bar{\pi}_k \rangle \stackrel{\text{Df}}{=} \langle \pi_h + \pi_j, \bar{\pi}_i + \bar{\pi}_k \rangle.$$

Отметим, что всевозможные каналы на  $X$  образуют относительно операции умножения и сложения так же как разбиения полумодулярную решетку (обозначим  $\mathcal{L}_\rho(X)$ ) [3].

Для системы каналов  $K(X) = \{\rho_i(X) | i \in J\}$  определим функцию информационной зависимости:

$$J(K) \stackrel{\text{Df}}{=} \sum_{i \in J} H(\rho_i) - H(m(K)),$$

где

$$m(K) = \prod_{i \in J} \rho_i.$$

### 3. Разбиение как алгебраический эквивалент информации.

Разбиение  $\pi(X)$  можно интерпретировать как информацию о базисном множестве  $X$ . Естественно принимать, что если  $\pi_i(X) \leq \pi_j(X)$ , то информация, представляемая разбиением  $\pi_i$  больше информации, представляемой разбиением  $\pi_j$ . Эквивалентность разбиений означает их информационную равноценность. Принимаем, что  $\pi_i(X)$  квазинезависимо относительно  $\pi_j(X)$  (обозначаем  $\pi_i \perp \pi_j$ ), если и только если для любых  $B \in \bar{\pi}_i + \pi_j$  и  $V_j^{(\alpha)} \in \pi_j$  при  $V_j^{(\alpha)} \subset V$  верно  $\bar{\pi}_i(B) \equiv \bar{\pi}_i(V_j^{(\alpha)})$ .

Квазинезависимость представляет собой отношение толерантности, т.е. является рефлексивной и симметричной. Квазинезависимость разбиений  $\pi_i(X)$  и  $\pi_j(X)$  эквивалентна условию  $H(\pi_i) + H(\pi_j) = H(\pi_i \cdot \pi_j) + H(\pi_i + \pi_j)$  [4] и является обобщением известного понятия независимости (именно разбиения являются независимыми, если они квазинезависимы и их сумма равна единичному разбиению).

Пусть  $G(\pi_i)$  – функция с вещественной областью значений, определенная на совокупности всех разбиений  $\pi_i(X)$  на всевозможных конечных множествах  $X$ . Для того, чтобы  $G(\pi_i)$  представляла количественную оценку информационной содержи-

мости разбиения  $\pi_i$ , она должна удовлетворять следующим условиям:

$$(И1) \quad \pi_i(X') \equiv \pi_j(X'') \Rightarrow G(\pi_i) = G(\pi_j);$$

$$(И2) \quad \alpha) \pi_i(X) \geq \pi_j(X) \Rightarrow G(\pi_i) \leq G(\pi_j);$$

$$\delta) G(\pi_i) + G(\pi_j) \geq G(\pi_i \cdot \pi_j) + G(\pi_i + \pi_j);$$

$$(И3) \quad \pi_i \cdot \pi_j \Rightarrow G(\pi_i) + G(\pi_j) = G(\pi_i \cdot \pi_j) + G(\pi_i + \pi_j).$$

Отметим, что условия (И1)–(И3) представляют основные свойства понятия информации.

Теорема I [4]. Если функция  $G(\pi_i)$  удовлетворяет условиям (И1)–(И3), то  $G(\pi_i) = aH(\pi_i) + b$ , где  $a$  и  $b$  – некоторые положительные константы.

Теорема I показывает, что содержание понятия энтропии не ограничивается уже известными интерпретациями Л. Больцмана и К. Шеннона, а скрывает в себе и алгебраические основы. Это позволяет рассматривать энтропию разбиения как количественную оценку его информационной содержимости.

Функции энтропии присущи следующие свойства:

(i) неотрицательность: для любого  $\pi_i(X)$  всегда  $H(\pi_i) \geq 0$ ;

(ii) инвариантность: если  $\pi_i(X') \equiv \pi_j(X'')$ , то  $H(\pi_i) = H(\pi_j)$ ;

(iii) монотонность: из неравенств  $\pi_i(X) \leq \pi_k(X)$  и  $\pi_h(X) \geq \pi_j(X)$  следует  $H(\pi_i/\pi_h) \geq H(\pi_k/\pi_j)$ ;

(iv) рекурсивность: для любых  $\pi_i(X)$  и  $\pi_j(X)$  верно

$$H(\pi_i \cdot \pi_j) = H(\pi_i) + \sum_{\alpha=1}^{m_i} q_{\alpha}(B_i^{(\alpha)}) H(\pi_j(B_i^{(\alpha)}));$$

(v) субаддитивность: для любых  $\pi_h(X)$ ,  $\pi_i(X)$ ,  $\pi_j(X)$  и  $\pi_k(X)$  справедливо

$$(a) \quad H(\pi_i/\pi_h) + H(\pi_k/\pi_j) \geq H(\pi_i \cdot \pi_k/\pi_h \cdot \pi_j);$$

$$(б) \quad H(\pi_i) + H(\pi_j) \geq H(\pi_i \cdot \pi_j) + H(\pi_i + \pi_j).$$

Если разбиение интерпретировалось как информация о базисном множестве, то понятие канала служит информационной характеристикой функционального соотношения на базисном множестве. Из свойств монотонности и субаддитивности энтропии вытекает, что для любых каналов  $\rho_i(X)$  и  $\rho_j(X)$  справедливо  $\rho_i(X) \geq \rho_j(X) \Rightarrow H(\rho_i) \geq H(\rho_j)$  и  $H(\rho_i) + H(\rho_j) \geq H(\rho_i \cdot \rho_j)$ .



4. Информационный критерий при построении сети автоматов. По общей теореме декомпозиции конечных автоматов [2] любая полная система разбиений  $P(S)$  определяет декомпозицию автомата  $A = \langle I, S, \delta \rangle$ . При этом каждому разбиению  $\pi_i \in P(S)$  соответствует в сети  $N$  компонентный автомат  $A_i$ . От выбора  $P(S)$  зависят как свойства, так и сложность сети  $N$ . Во многих случаях требуется найти сеть  $N$  с минимальным числом связей между компонентными автоматами. Именно этой проблеме уделяется в данной работе главное внимание. Предлагаемый нами подход опирается на предположение, что сложность сети  $N$  характеризуется функцией информационной зависимости системы каналов, представляющих  $P(S)$ .

Называем канал  $\rho_i = \langle \pi_i(S), \tau_i(S) \rangle$  на множестве состояний  $S$  автомата  $A = \langle I, S, \delta \rangle$   $h$ -каналом, если и только если  $(\forall x \in I) (s' \equiv s''(\tau_i) \Rightarrow \delta(s', x) \equiv \delta(s'', x)(\pi_i))$ .  $h$ -каналы представляют гомоморфные образы  $\delta_h: I \times \tau(S) \rightarrow \pi(S)$  функционального соответствия  $\delta: I \times S \rightarrow S$  [1]. Количество информации, необходимое для функционирования построенного по  $h$ -каналу  $\rho_i$  компонентного автомата  $A_i$ , определяется как  $H(\tau_i(S)/\pi_i(S)) = H(\rho_i)$ .

Введем на элементах решетки  $\mathcal{L}_\rho(X)$  отношение порядка, принимая, что  $\langle \pi_h, \pi_i \rangle \leq \langle \pi_j, \pi_k \rangle \Leftrightarrow \pi_h \leq \pi_j \wedge \pi_i \geq \pi_k$ . Подрешетку  $\mathcal{L}'_\rho(X) \subset \mathcal{L}_\rho(X)$  называем  $\alpha$ -решеткой, если и только если для любых  $\rho_i \in \mathcal{L}'_\rho(X)$  и  $\rho_j \in \mathcal{L}_\rho(X)$  из  $\rho_j \leq \rho_i$  следует  $\rho_j \in \mathcal{L}'_\rho(X)$ . Из свойств алгебры пар разбиений следует:

Теорема 2. Для любого автомата  $A$  множество всех его  $h$ -каналов образует  $\alpha$ -решетку  $\mathcal{L}_\rho^{(h)}(S)$ .

Решетку  $\mathcal{L}_\rho^{(h)}(S)$  можно рассматривать как определенное информационное представление автомата  $A$ , позволяющее упорядочить все его потенциально возможные информационные каналы и судить о структуре его реализации [1].

Из теоремы 2 следует, что для любого  $\pi_i \in P(S)$  существует максимальное разбиение  $M(\pi_i) = \sum \tau_i, \langle \pi_i, \tau_i \rangle \in \mathcal{L}_\rho^{(h)}(S)$ , образующее  $h$ -канал с разбиением  $\pi_i$ . Нетрудно видеть, что  $H(\langle \pi_i(S), M(\pi_i(S)) \rangle)$  определяет минимальное количество информации, необходимое для  $A_i$  от компонентов сети  $N$ .

Из теоремы 2 также следует, что в решетке  $\mathcal{L}_\rho^{(h)}(S)$  существуют элементы, которые являются минимальными по отношению частичного порядка " $\leq$ " (обозначаем эти элементы через  $\rho_k^{\min}$ ). Каналы  $\rho_k^{\min} = \langle \tau_k(S), \pi_k(S) \rangle$  имеют особую роль при декомпозиции автомата, поскольку построенные на их основе компонентные автоматы имеют максимальные эффективности преобразования информации, т.е.

$$(\forall \rho_i \in \mathcal{L}_\rho^{(h)}(S)) (\rho_i \geq \rho_k^{\min} \Rightarrow H(\rho_i) \geq H(\rho_k^{\min})).$$

Введем для любой полной системы разбиений  $P(S)$  на множестве состояний автомата  $A$  обозначение

$$K_P(S) \stackrel{\text{df}}{=} \{ \langle \pi_i(S), \tau_i(S) \rangle \mid \pi_i \in P(S), \tau_i = M(\pi_i) \}.$$

Теорема 3. Для любой полной системы разбиений  $P(S)$  найдется покрытая ей полная система разбиений  $P'(S)$  того же ранга, при  $K_{P'}(S) \subset \{ \rho_i^{\min} \}$  такая, что  $\mathcal{F}(K_{P'}(S)) \leq \mathcal{F}(K_P(S))$ .

Доказательство. Для любой  $\rho_i \in K_P(S)$  всегда найдется  $\rho_i^{\min}$  так, что  $\rho_i \geq \rho_i^{\min}$ . Отсюда следует, что система

$$P'(S) \stackrel{\text{df}}{=} \{ \pi'_i \mid \langle \pi'_i, \tau'_i \rangle = \rho_i^{\min} \leq \langle \pi_i, \tau_i \rangle = \rho_i \}$$

также полная. Так как  $P(S)$  и  $P'(S)$  полные, то

$$\begin{aligned} m(K_P) = m(K_{P'}) = 0 \quad \text{и} \quad (\forall i \in J) (\rho_i^{\min} \leq \rho_i) &\Rightarrow \\ \Rightarrow (\forall i \in J) (H(\rho_i^{\min}) \leq H(\rho_i)) &\Rightarrow \sum_{i \in J} H(\rho_i^{\min}) \leq \\ \leq \sum_{i \in J} H(\rho_i) &\Rightarrow \mathcal{F}(K_{P'}(S)) \leq \mathcal{F}(K_P(S)). \end{aligned}$$

Из теоремы 3 следует, что для получения декомпозиции автомата  $A$  с минимальной информационной зависимостью целесообразно при выборе исходной полной системы разбиений  $P(S)$  учитывать отношение " $\leq$ " на множестве  $h$ -каналов. Наилучшая реализация ( $\mathcal{F}(K_P(S)) = \min$ ) получается в случае, если элементами  $K_P(S)$  являются  $\rho_k^{\min}$ . В случае, если декомпозиция не может быть построена только на основе каналов  $\rho_k^{\min}$ , следует выбирать дополнительные каналы, которые как можно ближе по отношению " $\leq$ " к максимальным каналам. Отметим, что в случае параллельной декомпозиции  $\mathcal{F}(K_P(S)) = 0$ , что свидетельствует об отсутствии информационных связей между компонентными автоматами.

Рассмотрим теперь задачу минимизации информационной зависимости в сети  $N$  в случае, если полная система разбиений  $P(S)$  уже выбрана. Для решения этой задачи используем

понятие структурной матрицы [2]. Структурной матрицей сети  $N$  называется матрица  $\|m_{ij}\|$  при  $i, j \in J$ , определенная следующим образом:

$$m_{ij} = \begin{cases} 1, & \text{если } j \in J_i; \\ 0, & \text{если } j \notin J_i. \end{cases}$$

Содержательно  $m_{ij} = 1$  интерпретируется как соединение между компонентными автоматами  $A_i$  и  $A_j$ .

Матрице  $\|m_{ij}\|$  с минимальным числом единиц соответствует сеть  $N$  с минимальным числом связей [2]. Однако минимальное заполнение структурной матрицы не гарантирует в общем случае минимальной информационной зависимости в сети  $N$ . Следующий алгоритм заполнения  $\|m_{ij}\|$  приводит к сокращению информационной зависимости при декомпозиции:

A1. Для каждого разбиения  $\pi_k(S) \in \Pi_1(K_p(S))$  выясняем, не окажется ли оно меньше некоторого разбиения из системы  $\{\tau_j | j \in J\} = \Pi_2(K_p(S))$ . Из множества  $\{\pi_k | \pi_k \leq \tau_j\}$  найдем разбиение  $\pi_i$ , для которого разность  $\Delta_j(K) \stackrel{\text{def}}{=} H(\pi_k) - H(\tau_j)$  является минимальной. Принимаем, что элемент  $m_{ij} = 1$ , а столбец с индексом  $j$  из матрицы вычеркиваем.

A2. Находим всевозможные двухкомпонентные произведения  $\{\pi_h \cdot \pi_i | h, i \in J; h < i\}$  и найдем такие столбцы  $p$ , где  $\pi_h \cdot \pi_i \leq \tau_p$ . Из всех произведений  $\pi_{h_p} \cdot \pi_{i_p} \leq \tau_p$  выбираем то  $\pi_j \cdot \pi_k$ , для которого разность  $\Delta_p(j, k) \stackrel{\text{def}}{=} H(\pi_j) + H(\pi_k) - H(\tau_p)$  является минимальной. Элементы матрицы  $m_{jp}$  и  $m_{kp}$  принимаем предварительно равными значению  $\Delta_p(j, k)$ .

A3. Находим всевозможные трехкомпонентные произведения  $\{\pi_h \cdot \pi_i \cdot \pi_j | h, i, j \in J; h < i < j\}$  и вычисляем для них соответствующие разности  $\Delta_p(h, i, j) \stackrel{\text{def}}{=} H(\pi_h) + H(\pi_i) + H(\pi_j) - H(\tau_p)$ . Из системы  $\{\Delta_p(h, i, j) | h, i, j \in J; h < i < j\}$  найдем минимальное  $\Delta_{\min}^{(3)}$ . Если столбец  $p$  имеет ненулевые элементы, которые меньше  $\Delta_{\min}^{(3)}$ , то вместо значений  $\Delta_{\min}^{(2)}$  для парных произведений принимаем эти элементы окончательно равными единице и вычеркиваем из матрицы столбец  $p$ .

Соответствующие элементы столбца  $[\tau_p]$ , для которых найдется произведение  $\pi_k \cdot \pi_l \cdot \pi_m \leq \tau_p$  с минимальной  $\Delta_p(k, l, m)$  среди всех таких произведений, опять принимаем предварительно равными  $\Delta_p(k, l, m)$  при условии, что все элементы столбца  $[\tau_p]$  раньше имели нулевое значение или их значение было

меньше, чем  $\Delta_r(k, l, m)$ . Остальные элементы столбца  $[\tau_r]$  заменяем опять нулями.

А4. Затем процедура повторяется для всевозможных четырехкомпонентных произведений  $\pi_i \cdot \pi_j \cdot \pi_k$ , потом пятикомпонентных произведений и т.д. до тех пор, пока не окажется, что в матрице все столбцы вычеркнуты.

Заключение. Предложенная в работе энтропийная оценка для пар разбиений позволяет количественно моделировать процесс обработки информации в сетях автоматов. По информационной зависимости системы каналов можно оценить и функциональную зависимость между компонентными автоматами в сети автоматов. Тем самым можно начиная уже с этапа выбора полной системы разбиений направить процесс декомпозиционного синтеза, сократить перебор вариантов. Существенным является и то, что по энтропийным оценкам можно оценить и те разбиения или пары разбиений, которые в соответствующих решетках являются несравнимыми.

#### Л и т е р а т у р а

1. Hartmanis J., Stearns R.E. Algebraic structure theory of sequential machines. Prentice-Hall, Inc. Englewood Cliffs, New York, 1966.
2. Ке э в а л л и к А.Э. Теорема декомпозиции конечных автоматов. Автоматика и вычислительная техника, 1974, № 1, с. 17-24.
3. Birkhoff G. Lattice theory. Amer. Math. Soc. Collog. Publ., 1948.
4. Л а у с м а а Т.М. Об алгебраических основах понятия энтропии. - Изв. АН ЭССР. Физ. матем., 1983, 32, № 2, с. 128-134.

Informational Criterion for the Decomposition  
of Finite Automata

Summary

A decomposition method of finite automata based on the classical algebraic structure theory of automata by J. Hartmanis and R. Stearns is presented. Assuming notion of partition on finite set as an algebraic equivalent of information, a quantitative measure of this information in the form of entropy for partitions and partition pairs is proposed. This enables to develop further the decomposition method of finite automata allowing more effectively to compare different alternative variants of synthesis and improve the final result.



### ДЕКОМПОЗИЦИЯ МИКРОПРОГРАММНЫХ АВТОМАТОВ С РАЗДЕЛЕНИЕМ ВЫХОДНЫХ ПЕРЕМЕННЫХ

В работах [1, 2] предложены методы декомпозиционного синтеза сложных микропрограммных автоматов (МПА) на базе программируемых БИС. Сущность подхода состоит в том, что синтезируемый автомат разлагается на абстрактном уровне в сеть автоматов с учетом ограниченных ресурсов базисных элементов (число внешних полюсов БИС, быстродействие программной реализации). Однако при декомпозиции реальных МПА часто возникают трудности с разложением выходной функции. В настоящей работе предлагается метод декомпозиции МПА с разделением выходных переменных. Вследствие этого выходная функция реализуется децентрализованно, что имеет существенное значение как при реализации на базе программируемых матриц, так и микропроцессоров. В последнем случае повышается быстродействие программной реализации.

Формально МПА (имеющий  $L$  входных и  $N$  выходных двоичных каналов) определяется как система [3]

$$A = (S, \{0, 1\}^L, \{0, 1\}^N, \delta, \lambda),$$

где  $S = \{s_i \mid i=1, \dots, M\}$  - множество внутренних состояний;

$\{0, 1\}^L$  - входной алфавит;

$\{0, 1\}^N$  - выходной алфавит;

$\delta: D(\delta) \rightarrow S$  - функция переходов;

$D(\delta) \subseteq S \times \{0, 1\}^L$  - область определения функции  $\delta$ ;

$\lambda: D(\lambda) \rightarrow \{0, 1\}^N$  - функция выходов;

$D(\lambda) \subseteq S$  - область определения функции  $\lambda$ .

Автомат  $M = (S, \{0, 1\}^L, \delta)$ , не имеющий явно выделенных выходов, будем называть полуавтоматом.

Сеть МПА с распределенной выходной функцией определяется как система

$$N_{\lambda} = (A, B, \{0,1\}^L, \Phi),$$

где  $I^0 A = \{A_i \mid i \in I\}$  - множество компонентных автоматов сети,  $|I| = n_1$ ,

$$A_i = (S_i \{0,1\}^{l_i+k_i}, \{0,1\}^{q_i}, \delta_i, \lambda_i).$$

Здесь  $\{0,1\}^{l_i}$  - внешний входной алфавит;  
 $\{0,1\}^{k_i}$  - внутренний входной алфавит;  
 $\{0,1\}^{q_i}$  - выходной алфавит;

$\delta_i: D(\delta_i) \rightarrow S_i$  - функция переходов,  
 где  $D(\delta_i) \subseteq S_i \times \{0,1\}^{l_i+k_i}$ ;

$\lambda_i: D(\lambda_i) \rightarrow \{0,1\}^{q_i}$  - функция выходов,  
 где  $D(\lambda_i) \subseteq S_i$ .

$2^0 B = \{M_j \mid j \in J\}$  - множество компонентных полуавтоматов сети,  $|J| = n_2$ ,

$$M_j = (S_j, \{0,1\}^{l_j+k_j}, \delta_j).$$

$3^0 \{0,1\}^L$  - входной алфавит сети.

$4^0 \Phi = \{\psi_v \mid v \in V = I \cup J\}$  - множество функций соединения,

$$|V| = n_1 + n_2 = n, \psi_v: \prod_{w \in V} S_w \rightarrow \{0,1\}^{k_v}, V_v \in V.$$

Определения понятия реализации и постановка задачи декомпозиции совпадают с приведенными в [3] с той разницей, что заданный МПА декомпозируется в сеть с распределенной выходной функцией.

Для решения поставленной задачи будем использовать понятие  $\mu(y_k)$  разбиения ( $y_k \in Y$ ) на множестве состояний  $S$  декомпозируемого автомата:

$$\mu(y_k)[s_i] = \mu(y_k)[s_j] \Leftrightarrow pr_k \lambda(s_i) = pr_k \lambda(s_j).$$

Здесь запись  $\mu(y_k)[s_i] = \mu(y_k)[s_j]$  означает, что состояния  $s_i$  и  $s_j$  содержатся в одном и том же блоке разбиения  $\mu(y_k)$ .

Конструктивно  $\mu(y_k)$  есть двублочное разбиение, в первый блок которого попадают состояния, в соответствующем выходном наборе, для которых  $k$ -я выходная переменная равна 1, а во второй блок - те состояния, для которых она равна 0.

Нетрудно показать, опираясь на результаты работ [1, 3], что значение  $k$ -й выходной переменной сети может быть вычислено  $i$ -м компонентным автоматом, если и только если соответствующее ему декомпозиционное разбиение  $\pi_i \leq \mu(y_k)$ .



Предлагаемый метод декомпозиции МПА исходит из ортогонального множества неполностью определенных разбиений  $\mathcal{P}$  на множестве состояний декомпозируемого автомата [1]. Уточним понятие неполностью определенного разбиения (НОР).

НОР на множестве  $S$  называется совокупность подмножеств множества  $S$   $\rho = (B_1, \dots, B_m, b_d)$  таких, что  $B_i \neq \emptyset$  ( $i=1, \dots, m$ ),  $B_i \cap B_j = \emptyset$  ( $i \neq j$ ),  $\bigcup_{i=1}^m B_i = S \setminus b_d$ ,  $b_d$  - специальный блок НОР.

НОР  $\rho \in \mathcal{P}$  можно интерпретировать как компактную запись целой совокупности  $G(\rho)$  разбиений на  $S$ . Причем,  $\pi \in G(\rho)$ , если и только если для каждого  $B' \in \pi$  либо найдется блок  $B \in \rho$  такой, что  $B \subseteq B' \subseteq B \cup b_d$ , либо  $B' \subseteq b_d$ . При преобразовании НОР в  $\pi \in G(\rho)$  элементами из  $b_d$  могут быть дополнены неспециальные блоки из  $\rho$  или образованы новые блоки разбиения  $\pi$ .

НОР  $\rho_1$  не меньше, чем НОР  $\rho_2$  ( $\rho_1 \geq \rho_2$ ), если и только если для каждого  $\pi_1 \in G(\rho_1)$  найдется  $\pi_2 \in G(\rho_2)$  такое, что  $\pi_1 \geq \pi_2$ .

Произведением НОР  $\rho_1, \rho_2, \dots, \rho_n$  называется НОР  $\prod_{i=1}^n \rho_i$ , специальный блок которого образуется объединением специальных блоков НОР  $\rho_1, \rho_2, \dots, \rho_n$ , а для его неспециальных блоков справедливо, что элементы  $z$  и  $z'$  принадлежат одному и тому же блоку, если и только если они принадлежат одному неспециальному блоку как в  $\rho_1$ , так и в  $\rho_2, \dots, \rho_n$ .

Множество НОР  $\mathcal{P}$  называется ортогональным, если произведение входящих в него НОР равно НОР  $\rho(0)$ , все блоки которого, кроме специального, состоят из одного элемента.

Теорема. Ортогональному множеству НОР  $\mathcal{P}$  может быть сопоставлена сеть с распределенной выходной функцией в том и только в том случае, если для любого разбиения  $\mu(y_k)$ ,  $y_k \in Y$  найдется  $\rho \in \mathcal{P}$  такое, что  $\rho \leq \mu(y_k)$ .

Доказательство. Для доказательства достаточности утверждения покажем, каким образом может быть задана функция выходов  $i$ -го компонентного автомата (соответствующего НОР  $\rho_i$ ). Пусть  $\mu(y) \geq \rho_i$ , следовательно, существует  $\pi \in G(\rho_i)$  такое, что  $\mu(y) \geq \pi$  ( $\pi$  может быть всегда получено, например, образованием одноэлементных блоков из состояний, принадлежащих специальному блоку НОР  $\rho_i$ ). Включим выходную переменную  $y$  в множество выходных переменных  $i$ -го компонентного

автомата  $A_i$  и будем считать, что если  $A_i$  находится в состоянии, соответствующем  $B \in \mathcal{L}$ , то  $y_i$  принимает значение, соответствующее блоку  $B'$  из  $\mu(t)$  такому, что  $B \subseteq B'$ .

Пусть  $\mu(y_{i_1}) \geq \rho_1, \dots, \mu(y_{i_m}) \geq \rho_m$ . Тогда по определению существует  $\pi \in G(\rho_i)$  такое, что  $\prod_{j=1}^m \mu(y_{i_j}) \geq \pi$ . Включим выход-

ные переменные  $y_{i_1}, \dots, y_{i_m}$  в множество выходных переменных  $i$ -го компонентного автомата и будем считать, что если  $A_i$  находится в состоянии  $B \in \mathcal{L}$ , то переменные  $y_{i_1}, \dots, y_{i_m}$  принимают значения, соответствующие блокам  $B'_{i_1}$  из  $\mu(y_{i_1}), \dots, B'_{i_m}$  из  $\mu(y_{i_m})$ , для которых  $\prod_{j=1}^m B'_{i_j} \supseteq B$ .

Так как по условию любой выходной переменной  $y_t$  декомпозируемого МПА найдется НОР  $\rho \in \mathcal{P}$  такое, что  $\mu(y_t) \geq \rho$ , то все выходные переменные декомпозируемого МПА могут быть распределены по компонентным МПА сети, получаемой в результате декомпозиции. Следовательно, на множестве выходных переменных  $Y$  может быть образовано разбиение  $\pi_\lambda$ , для каждого блока  $B$  которого существует НОР  $\rho \in \mathcal{P}$  такое, что переменная  $y$  принадлежит  $B$  в том и только в том случае, если  $\mu(y) \geq \rho$ .

Необходимость доказываемого утверждения следует непосредственно из результатов работы [3] и того факта, что множеству НОР  $\mathcal{P} = \{\rho_i | i=1, \dots, n\}$  можно поставить в соответствие ортогональное множество разбиений  $P = \{\pi_i | \pi_i \in G(\rho_i), i=1, \dots, n\}$ , если и только если множество НОР  $\mathcal{P}$  ортогонально.

В случае выполнения условия теоремы будем говорить, что множество НОР  $\mathcal{P}$  покрывает множество  $\{\mu(y_i) | y_i \in Y\}$ .

При доказательстве теоремы было введено понятие разбиения  $\pi_\lambda$  на множестве выходных двоичных переменных  $Y$  декомпозируемого МПА, которое обладает тем свойством, что

$$\forall B \in \pi_\lambda \exists \rho \in \mathcal{P} \left( \prod_{y \in B} \mu(y) \geq \rho \right).$$

Из доказательства следует, что значения всех выходных переменных, которые принадлежат блоку  $B$ , могут быть вычислены компонентным автоматом сети, соответствующим разбиению  $\pi \in G(\rho)$ .

При построении разбиения  $\pi_\lambda$  практически целесообразно проводить оптимизацию по числу блоков. Задача может быть сформулирована так: дано ортогональное множество НОР  $\mathcal{P} = \{\rho_i | i=1, \dots, n\}$ , требуется найти его подмножество  $\mathcal{P}'$  ми-

нимальной мощности такое, что для любой выходной переменной  $y \in Y$  найдется  $\rho \in \mathcal{P}'$  не большее, чем  $\mu$ -разбиение  $\mu(y)$ , т.е.

$$\forall y \in Y \exists \rho \in \mathcal{P}' (\mu(y) \geq \rho).$$

Остановимся на решении этой задачи.

Выходной структурной матрицей сети МПА назовем булеву матрицу  $T_\lambda$ , строки которой сопоставлены  $\mu(y_k)$ -разбиениям,  $y_k \in Y$ , а столбцы – НОР из множества  $\mathcal{P}$ . В  $k$ -ю строку матрицы записывается единица в  $i$ -й столбец, если и только если выполнено условие  $\mu(y_k) \geq \rho_i$ .

Задача нахождения  $\mathcal{P}' \subseteq \mathcal{P}$  минимальной мощности может быть интерпретирована как задача нахождения кратчайшего покрытия булевой матрицы  $T_\lambda$ . Действительно, покрытие множества  $\{\mu(y_i) \mid i=1, \dots, N\}$  будет в этом случае представляться подмножеством столбцов из  $T_\lambda$ , содержащих сообща не менее одной единицы в каждой из строк матрицы  $T_\lambda$ . Если это свойство подмножества утрачивается при удалении любого столбца, то покрытие называется простым. Простое покрытие с минимальным числом элементов называется кратчайшим. Задача нахождения кратчайшего покрытия может быть решена известными методами [4].

Заметим, что найденное подмножество  $\mathcal{P}'$  может и не обладать свойством ортогональности. Для того, чтобы получить множество разбиений на  $S$ , по которому можно декомпозировать заданный МПА, следует дополнить его некоторыми НОР из  $\mathcal{P} \setminus \mathcal{P}'$  до получения ортогонального множества. Очевидно, что последним НОР в сети будут соответствовать компонентные подавтоматы.

#### Л и т е р а т у р а

1. Лейс П.Л., Судницын А.В. Метод декомпозиции микропрограммных автоматов с учетом ограничений на число входных переменных. – Тр. Таллинск. политехн. ин-та, 1980, № 497, с. 41-47.

2. Кеэваллик А.Э., Судницын А.В. Использование методов декомпозиции конечных автоматов для построения сетей микропроцессоров. – В кн.: Прикладные аспекты теории автоматов: Тр. Международного семинара. Болгария, Варна, 1979, т. I, с. 54-65.

3. Jakobson G., Keevallik A., Leis P. Some aspects of the construction of microprogram automata networks. - In: Proceedings of the Second International Symposium on Discrete Systems. Dresden, 1977, vol. 1, p. 120-128.

4. Закревский А.Д. Логический синтез каскадных схем. М., Наука, 1981. 416 с.

A. Sudnitsyn

The Decomposition of Microprogram Automata  
with Distribution of Outputs

Summary

A method for the decomposition of the microprogram automata into the network of component automata with distribution of binary outputs is presented. The necessary and sufficient conditions for the automata decomposition existence are defined.

### ОПТИМИЗАЦИЯ БАЗИСА СЕТИ ПРИ ДЕКОМПОЗИЦИИ МИКРОПРОГРАММНЫХ АВТОМАТОВ

1. Введение. В связи с использованием больших и сверх-больших интегральных схем при синтезе дискретных устройств управления (ДУУ) наблюдается возрастание интереса к декомпозиционным методам синтеза ДУУ. Настоящая работа является продолжением работы [1] по декомпозиции микропрограммных автоматов.

Во многих случаях оптимизация полученной в результате декомпозиции сети автоматов позволяет получить более экономичные и надежные реализации. Данная работа посвящена минимизации компонентных автоматов сети благодаря целенаправленному кодированию входных алфавитов компонентных автоматов.

2. Основные понятия. Приведем определения основных понятий, используемых в данной работе.

Микропрограммным автоматом (без выхода) будем называть тройку  $M = (A, \{0,1\}^l, \delta)$ ,

где  $A$  - множество состояний,

$\{0,1\}^l$  - входной алфавит,

$\delta: D(\delta) \rightarrow A$  - функция переходов,  $D(\delta) \subseteq A \times \{0,1\}^l$ .

Модифицированным микропрограммным автоматом (ММПА) будем называть четверку  $M_m = (A, \{0,1\}^l, I, \delta)$ ,

где  $A$  - множество состояний,

$\{0,1\}^l$  - структурный входной алфавит,

$I$  - абстрактный входной алфавит,

$\delta: D(\delta) \rightarrow A$  - функция переходов,  $D(\delta) \subseteq A \times \{0,1\}^l \times I$ .

Каждому переходу МПА из состояния  $a_m$  в состояние  $a_b$  будем ставить в соответствие комплекс 0-кубов булева про-

странства  $\{0,1\}^{\ell} : K^{\circ}(a_m, a_s) = \{z \in \{0,1\}^{\ell} \mid \delta(a_m, z) = a_s\}$ .

Некоторое (обычно избыточное) покрытие комплекса  $K^{\circ}(a_m, a_s)$  обозначим через  $C(a_m, a_s)$ , а дизъюнктивную нормальную форму, соответствующую комплексу  $C(a_m, a_s)$ , через  $X(a_m, a_s)$ .

Сетью модифицированных микропрограммных автоматов будем называть систему

$$N_M = (B_M, \{0,1\}^{\ell}, \Phi_M),$$

где  $I^{\circ} B_M = \{M_{M_i} \mid i \in J = \{1, \dots, V\}\}$  - множество компонентных автоматов (базис сети),  $M_{M_i} = (A_i, \{0,1\}^{\ell_i}, I_i, \delta_i)$ .

Здесь  $\{0,1\}^{\ell_i}$  - внешний входной алфавит,  $\ell_i \leq \ell$ ;  
 $I_i$  - внутренний (абстрактный) входной алфавит.

$2^{\circ} \{0,1\}^{\ell}$  - входной алфавит сети.

$3^{\circ} \Phi_M = \{\varphi_{M_i} \mid i \in J\}$  - множество функций соединения (структура сети),

$$\varphi_{M_i} : X A_j \rightarrow I_i, J_j \in J.$$

Сетью микропрограммных автоматов будем называть систему

$$N_S = (B_S, \{0,1\}^{\ell}, \Phi_S),$$

где  $I^{\circ} B_S = \{M_{S_i} \mid i \in J = \{1, \dots, V\}\}$  - базис сети,

$$M_{S_i} = (A_i, \{0,1\}^{\ell_i + k_i}, \delta_i).$$

Здесь  $\{0,1\}^{\ell_i}$  - внешний входной алфавит,  
 $\{0,1\}^{k_i}$  - внутренний входной алфавит.

$2^{\circ} \{0,1\}^{\ell}$  - входной алфавит сети,

$3^{\circ} \Phi_S = \{\varphi_{S_i} \mid i \in J\}$  - структура сети,

$$\varphi_{S_i} : X A_j \rightarrow \{0,1\}^{k_i}, J_j \in J.$$

В [I] показано, что сетям  $N_M$  и  $N_S$  можно ставить в соответствие автоматы  $M_N^M$  и  $M_N^S$ , функционально эквивалентные  $N_S$  и  $N_M$  соответственно.

Под задачей декомпозиции микропрограммного автомата  $M$  понимается задача построения  $N_S$  (или сети  $N_M$ ) такой, что у автомата  $M_N^M$  (или автомата  $M_N^S$ ) существует изоморфный микропрограммному автомату  $M$  автомата [I].

3. Кодирование внутреннего входного алфавита. Изложенный в [1] метод декомпозиции заключается в следующем: сначала строится сеть  $N_m$ , а затем кодируются элементы внутренних алфавитов  $I_i, i \in J$ , для получения сети  $N_s$ . В данной работе мы предложим метод кодирования алфавитов  $I_i$ , позволяющий оптимизировать базис сети  $N_s$  (минимизировать компонентные автоматы).

Идея метода заключается в следующем. На фиг. 1 и 2 приведены таблицы переходов компонентного ММПА и МПА соответственно. Сравнение этих таблиц показывает, что строкам  $(a_2, a_1, 0_x, B_1)$ ,  $(a_2, a_1, 0_x, B_2)$  и  $(a_2, a_2, 1_x, B_1)$ ,  $(a_2, a_2, 1_x, B_2)$  таблицы переходов ММПА соответствуют лишь две строки таблицы переходов МПА:  $(a_2, a_1, 0_{xx})$  и  $(a_2, a_2, 1_{xx})$ , т.е. произошло склеивание строк. Очевидно, что для

$a_m$	$a_s$	$C(a_m, a_s)$		I
		$x_1$	$x_2$	
$a_1$	$a_1$	I	x	$B_1$
	$a_1$	x	0	$B_1$
	$a_1$	0	x	$B_2$
	$a_2$	0	I	$B_1$
	$a_2$	I	x	$B_2$
$a_2$	$a_1$	0	x	$B_1$
	$a_1$	0	x	$B_2$
	$a_2$	I	x	$B_1$
	$a_2$	I	x	$B_2$

Фиг. 1. Таблица переходов компонентного ММПА.

$a_m$	$a_s$	$C(a_m, a_s)$		
		$x_1$	$x_2$	$x_3$
$a_1$	$a_1$	I	x	0
	$a_1$	x	0	0
	$a_1$	0	x	I
	$a_2$	0	I	0
	$a_2$	I	x	I
$a_2$	$a_1$	0	x	x
	$a_2$	I	x	x

Фиг. 2. Таблица переходов компонентного МПА.

склеивания строк таблицы переходов ММПА должно выполняться следующее условие: эти строки отличаются лишь элементами внутреннего абстрактного алфавита I, которые закодированы соседними кодами (два кода называются соседними, если они отличаются только значением в одной координате). В рассматриваемом примере соседние коды получаются автоматически, так как алфавит I состоит из двух элементов.

Будем называть строки

$$r_i = (a_m, a_s, C(a_m, a_s), \theta) \text{ и } r_j = (a_t, a_u, C(a_t, a_u), \rho)$$

таблицы переходов ММПА сходными,  $r_i \sim r_j$ , если и только если

$$a_m = a_t, a_s = a_u, \theta \neq \rho \text{ и } C(a_m, a_s) \approx C(a_t, a_u).$$

Здесь отношение ' $\approx$ ' является отношением слабой эквивалентности булевых функций. Неполностью определенные булевы функции  $f_1$  и  $f_2$  являются слабо эквивалентными, если и только если

$$f_1^{-1}(0) \cap f_2^{-1}(1) = f_1^{-1}(1) \cap f_2^{-1}(0) = \emptyset,$$

где  $f^{-1}(y) = \cup \{x | f(x) = y\}$ .

Легко заметить, что сходные строки  $r_i$  и  $r_j$  склеиваются, если и только если  $\theta$  и  $\rho$  закодированы соседними кодами.

Зададим на алфавите  $I$  отношение  $\Psi$  следующим образом:

$$\theta \Psi \rho \Leftrightarrow \exists r_i \exists r_j : r_i \sim r_j,$$

где  $r_i, r_j$  - строки таблицы переходов ММПА, содержащие  $\theta \in I$  и  $\rho \in I$  соответственно.

Отношение  $\Psi$  симметричное, но необязательно транзитное. Далее будет показано, что задачу кодирования алфавита  $I$  можно свести к задаче разбиения симметричного графа на минимальное число полных подграфов. Отметим, что эта задача равносильна хорошо известной задаче раскраски графа.

Пусть задан граф  $G = (P, E)$ ,

где  $P$  - конечное множество вершин,

$E \subseteq P \times P$  - некоторое отношение на  $P$  (множество ребер).

Граф  $G$  называется симметричным если, отношение  $E$  симметрично. Граф  $G$  называется полным, если  $E = P \times P$ . Граф  $G_1$  называется полным подграфом графа  $G$ , если и только если

1<sup>o</sup>  $G_1$  является подграфом  $G$ ;

2<sup>o</sup>  $G_1$  есть полный граф.

Множество  $\Pi(G) = \{G_1, \dots, G_n\}$  называется разбиением графа  $G$  на полные подграфы, если и только если

1<sup>o</sup>  $G_i = (P_i, E_i), 1 \leq i \leq n$  - полные подграфы  $G$ ;

2<sup>o</sup>  $\bigcup_{i=1}^n P_i = P$ ;

3<sup>o</sup>  $P_i \cap P_j = \emptyset, 1 \leq i, j \leq n, i \neq j$ .

Множество  $\pi_{\min}(G) = \{G_1, \dots, G_n\}$  называется разбиением  $G$  на минимальное число полных подграфов, если и только если

1<sup>o</sup>  $\pi_{\min}(G)$  является разбиением  $G$  на полные подграфы;



Параметры МПА		Кол. комп. ММПА		Компонентные ММПА					Направленная кодировка (параметры МПА)		Случайная кодировка (параметры МПА)				
Кол. входов	Кол. выходов	Кол-во рядов в табл. переходов	Кол. в сост.	Кол. комп. ММПА	пп №	Кол. входов	Кол. входов внутр. входов	Кол. рядов в табл. перех.	Кол. входов в табл. перех.	Кол. входов в табл. перех.	Кол. входов в табл. перех.	Относ. кол. рядов	Кол. входов в табл. перех.	Относ. кол. рядов	
															Кол. входов в табл. перех.
4	4	12	5	2	1	3	3	10	8	5	5	1,0	3	10	1,0
3	2	26	9	2	1	2	3	20	18	4	4	0,9	4	19	0,95
17	23	43	9	2	1	14	6	36	39	16	15	0,78	17	34	0,94
19	17	35	7	2	1	10	3	19	19	3	3	1,0	12	19	1,0
16	25	39	11	2	1	6	7	20	28	10	17	0,95	9	19	0,95
18	33	29	14	2	1	8	4	19	18	14	10	0,89	12	18	0,95
18	65	73	21	7	1	6	21	27	26	10	10	0,85	10	25	0,93
					2	5	19	30	30	12	21	0,81	10	26	1,0
					3	6	20	27	27	10	26	0,87	11	28	0,93
					4	6	20	33	33	12	23	0,85	11	25	0,93
					5	6	20	27	27	11	27	0,82	10	30	0,91
					6	6	20	33	33	10	23	0,85	10	23	0,85
					7	6	12	34	34	9	27	0,79	10	30	0,88

Фиг. 3. Результаты экспериментов.

$2^0$  не существует разбиения  $\pi'(G) = \{G'_1, \dots, G'_m\}$  такого, что  $m < n$ .

Некоторые возможные решения задачи разбиения симметричного графа на минимальное число полных подграфов обсуждались, например, в [2, 3, 4].

Пусть теперь  $G = (I, \Psi)$  — граф отношения  $\Psi, \pi(G) = \{G_1, \dots, G_n\}, G_t = (I_t, \Psi_t), 1 \leq t \leq n$ . Ясно, что элементы из  $I_t$  следует закодировать соседними кодами, причем для кодирования элементов необходимо  $2^{\lfloor \log_2 |I_t| \rfloor}$  соседних кодов. Последнее требование не всегда можно удовлетворить для всех  $1 \leq t \leq n$  при кодировании алфавита  $I$  кодами минимальной длины. Поэтому, в первую очередь, необходимо закодировать соседними кодами те множества  $I_t$ , которые приводят к максимальному сокращению строк таблицы переходов МПА. На основе вышеприведенных рассуждений предлагаем следующий алгоритм кодирования внутреннего входного алфавита ММПА.

Шаг 1. Определить отношение  $\Psi$ .

Шаг 2. Построить граф  $G$ .

Шаг 3. Найти  $\pi_{\min}(G) = \{G_1, \dots, G_n\}$ .

Шаг 4. Закодировать соседними кодами множество вершин  $I_t$  графа  $G_t, 1 \leq t \leq n$ .

Отметим, что хотя задача нахождения  $\pi_{\min}(G)$  является NP-полной, ее решение в данном случае не вызывает трудностей из-за небольшой мощности алфавитов  $I_t, 1 \leq t \leq n$  ( $|I_t|$  обычно не превышает 20).

Экспериментальное исследование было проведено на потоке реальных МПА и подтвердило эффективность разработанного метода кодирования внутренних алфавитов компонентных ММПА. На фигуре 3 приведена таблица с результатами экспериментов.

### Л и т е р а т у р а

1. Jakobson G., Keevallik A., Leis P. Some aspects of the construction of microprogram automata networks. - In: Proceedings of the Second International Symposium on Discrete Systems. Dresden, 1977, vol. 1, p. 120-128.

2. Вейнер Г.А. Об аппроксимации симметричного рефлексивного отношения отношением эквивалентности. - Тр. Таллинск. политехн. ин-та, 1971, № 313, с. 45-49.

3. Визинг В.Г., Плесневич Г.С. К проблеме минимальной раскраски вершин графа. - Сибирский математический журнал, 1965, № I, с. 42-47.

4. Рыжков А.П. Разбиение графа на минимальное число полных подграфов. - Кибернетика, 1975, № 6, с. 90-95.

P. Leis, P. Ellervee

The Network Basis Optimization on the  
Decomposition of Microprogram Automata

Summary

In this paper the method of minimization of component automata in the microprogram automata network is developed.



А.Э. Кеэваллик, А.В. Судницын,  
Б.Е. Беркман

### СИСТЕМА АВТОМАТИЗАЦИИ ДЕКОМПОЗИЦИОННОГО СИНТЕЗА ДИСКРЕТНЫХ УПРАВЛЯЮЩИХ УСТРОЙСТВ (DILOS)

Введение. В современных условиях повышение качества и снижение затрат на проектирование дискретных устройств управления уже не мыслимо без широкого внедрения методов и средств машинного проектирования. Эффективность машинного проектирования зависит существенным образом от пригодности методов и алгоритмов для решения задач синтеза устройств реальной сложности на логическом уровне проектирования [1]. Это особенно важно в тех случаях, когда из-за сложности заданного алгоритма управления и ограниченных ресурсов базисных элементов (число внешних полюсов БИС, быстродействие программной реализации и т.д.) мы вынуждены строить реализации в виде сетей из БИС. В работах [2, 3] показано, что многие проблемы синтеза в современном элементном базисе могут быть сведены к решению соответствующих задач декомпозиции автоматов. В результате декомпозиции автомат разлагается в сеть автоматов таким образом, что ее компонентные автоматы и вся сеть в целом удовлетворяли бы заданным требованиям. Такой подход позволяет формализовать и автоматизировать процесс синтеза и применим как для получения реализации на базе "жесткой логики" и программируемых матриц, так и для получения программных реализаций на базе микропроцессорных комплектов.

Характеристика задач декомпозиционного синтеза. Проведенный в работах [2, 3, 4, 5, 6] анализ показал, что при решении практических задач декомпозиции целесообразно опираться на аппарат алгебры пар разбиений [7], обладающий необходимыми средствами для решения задач декомпозиционного синтеза, ориентированных на получение реализаций в базисе

электронных компонентов различной степени интеграции. Решение задач декомпозиции сводится при этом к нахождению соответствующих систем разбиений на множествах состояний, входов и выходов микропрограммного автомата (МПА) [8], являющегося автоматной моделью проектируемого дискретного управляющего устройства. Все рассматриваемые задачи являются чрезвычайно трудоемкими (многие из них относятся к классу NP-полных), что повлекло необходимость разработки эффективных приближенных алгоритмов декомпозиционного синтеза, приемлемых в смысле точности решения и затрат на проектирование.

Описание системы DILOS. Разработанные в [2, 3, 4, 5, 6] формализованные методы и алгоритмы декомпозиционного синтеза явились методическим обеспечением автоматизированной системы декомпозиционного синтеза дискретных управляющих устройств (DILOS). В качестве входного языка пользователя для описания алгоритма функционирования проектируемого устройства используется язык граф-схем алгоритмов (ГСА) [8]. ГСА является эффективным средством описания дискретных управляющих устройств реальной сложности, широко используемым на практике.

Основные задачи, решаемые системой DILOS, следующие:

1. Редактирование исходной ГСА. Осуществляется трансляция ГСА, выявляется наличие синтаксических и семантических ошибок, производится их исправление.

2. Синтез МПА по заданной ГСА [8]. МПА представляется в форме прямой таблицы переходов. Причем может быть синтезирована как модель Мили, так и модель Мура.

3. Построение системы разбиений для декомпозиции МПА с учетом ограничений на число внешних выводов БИС.

4. Построение системы разбиений для декомпозиции МПА с целью повышения быстродействия программной реализации.

5. Построение сети МПА по системе разбиений на множестве состояний декомпозируемого автомата.

6. Реализация МПА на стандартных ПЛИС.

7. Программная реализация МПА. Автомат представляется в виде совокупности бинарных графов.

8. Реализация МПА в базе малых и средних интегральных схем.

В основу построения программного обеспечения системы (в соответствии с ГОСТ 23501.0-79) положены такие качества как адаптируемость к различным конфигурациям ЭВМ и их операционным системам, модульность построения, обеспечение мультипрограммной работы, режим диалога, режим разделения времени. Программное обеспечение составляет система программ, написанных на языке ЛЯПАС-М [9], интегрированная с модифицированной системой программирования ТОМЛЕС [10].

ЛЯПАС-М (Логический Язык для Представления Алгоритмов Синтеза) ориентирован на программирование задач комбинаторно-логического характера. К задачам этого класса и относятся перечисленные задачи синтеза. Последнее обстоятельство как и богатые возможности языка, его модульность, открытость (способность к расширению) явились причиной его использования для разработки представляемой системы логического проектирования.

К базовым объектам [1], из которых образуются объекты, фигурирующие в задачах синтеза, решаемых системой, отнесены булева матрица, троичная матрица, последовательность натуральных чисел, секционированная последовательность натуральных чисел, секционированная последовательность символов, секционированная булева матрица, секционированная троичная матрица, таблица натуральных чисел.

Тексты всех программ системы DILOS хранятся в архивах, которые для операционных систем ЕС ЭВМ представляют собой единый файл прямого доступа. Причем, работа модулей архивной службы не связана с действием блоков операционной системы, за исключением выполнения основных макрокоманд ввода-вывода.

Система ТОМЛЕС модифицирована в расчете на одновременную работу нескольких пользователей в режиме разделения времени. Ее блоки хранятся в виде загрузочных модулей в библиотеке ОС ЕС или в виде М-сегментов (сегментов на машинном языке) в файле системных архивов.

По желанию пользователя допускается как пакетный, так и диалоговый режим работы. В первом случае все сегменты, ре-

шающие отдельные задачи синтеза, объединяются в единую иерархическую систему, имеющую входной язык описания задания и осуществляющую сквозной синтез в соответствии с конкретным заданием. Общая схема работы сегментов системы показана на фиг. 1. Стрелками показаны направления передачи информации между внешними носителями (перфокарты, диски) и программами. Ромбики означают выбор между альтернативными способами получения информации программой. Каждый модуль, реализующий отдельный этап декомпозиционного синтеза, является самостоятельной подсистемой и характерен тем, что:

1<sup>0</sup> Входные объекты для него представляют из себя тексты в формате п/карт или структуры в архивах.

2<sup>0</sup> Выходные объекты для него представляют из себя структуры в архивах.

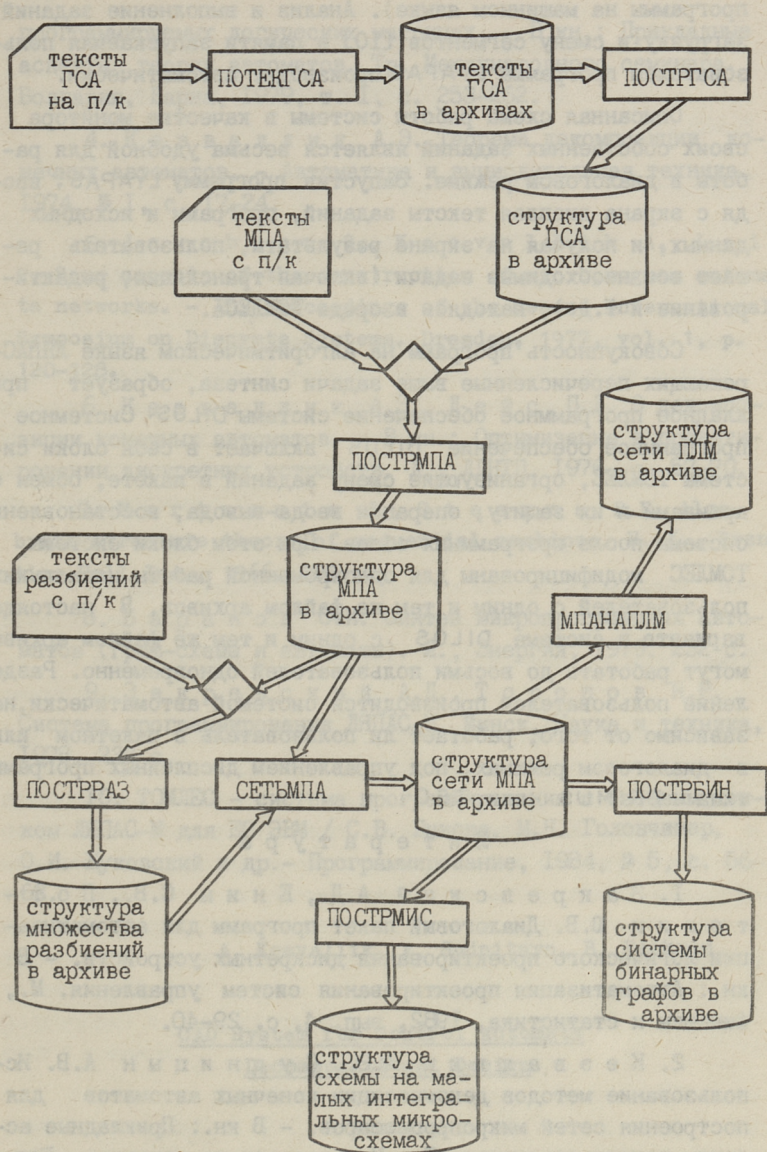
Подобная организация системы, соединенная с аппаратом кодов завершения, позволяет, во-первых, работать с объектами больших размерностей без угрозы нехватки объема оперативной памяти, а во-вторых, при аварийных завершениях работы системы не повторять уже выполненные этапы синтеза (так как полученная информация сохраняется в архивах на дисках).

Использование диалогового режима работы проектировщика с системой DILOS дает ему возможность решать каждую задачу синтеза, представляющую самостоятельный интерес, в ходе отдельного сеанса диалога. Тем самым проектировщик имеет возможность активно вмешиваться в процесс синтеза с целью более глубокой оптимизации результата.

Отметим особенности системы ТОМЛЕС, которые позволили эффективно организовать диалоговый режим работы как при отладке, так и при реализации системы DILOS. Задания, состоящие из набора директив, заказа (из модулей на ЛЯПАС-М) и исходных данных, располагаются в файле CARDIN. При этом задания могут иметь сложную сегментированную структуру, что позволяет сократить требуемый ресурс памяти. Л-модули, необходимые для реализации алгоритма конкретной задачи и образующие заказ, могут находиться как в файле

CARDIN, так и в архивах системы. В архивах системы хранятся также М-модули (переместимые реентерабельные





Фиг. 1. Общая схема работы сегментов системы DILOS.

программы на машинном языке). Анализ и выполнение заданий, загрузку и смену сегментов [10] в памяти запускаемая пользователем программа LYAPAS производит автоматически.

Описанная схема работы системы в качестве монитора своих собственных заданий является весьма удобной для работы в диалоговом режиме. Запустив программу LYAPAS, вводя с экрана дисплея тексты заданий, программ и исходных данных, и получая на экране результаты, пользователь решает все необходимые задачи (включая трансляцию, редактирование и т.д.), находясь в среде ТОМЛЕСа.

Совокупность программ на алгоритмическом языке ЛЯПАС-М, решающих перечисленные выше задачи синтеза, образует прикладное программное обеспечение системы DILOS. Системное программное обеспечение DILOS включает в себя блоки системы ТОМЛЕС, организующие смену заданий в пакете, обмен с архивами и их защиту, операции ввода-вывода, восстановление системы после программных сбоев. При этом блоки системы ТОМЛЕС модифицированы для одновременной работы нескольких пользователей с одним и тем же файлом архивов. В настоящем варианте в системе DILOS с одним и тем же файлом архивов могут работать до восьми пользователей одновременно. Разделение пользователей производится системой автоматически, независимо от того, работает ли пользователь в пакетном или в диалоговом режимах, под управлением дисплейных программ типа PRIMUS или TSO.

#### Л и т е р а т у р а

1. З а к р е в с к и й А.Д., Е н и н С.В., П о т т о с и н Ю.В. Диалоговый пакет программ для автоматизации логического проектирования дискретных устройств. - В кн.: Автоматизация проектирования систем управления. М., Финансы и статистика, 1982, вып. 4, с. 29-40.

2. К е э в а л л и к А.Э., С у д н и ц ы н А.В. Использование методов декомпозиции конечных автоматов для построения сетей микропроцессоров. - В кн.: Прикладные аспекты теории автоматов. Тр. Международного семинара. Болгария, Варна, 1979, т. I, с. 54-65.

3. Л е й с П.Л., С у д н и ц ы н А.В. Один метод декомпозиционного синтеза микропрограммных автоматов на

программируемых логических матрицах. - В кн.: Прикладные аспекты теории автоматов. Тр. Международного семинара. Болгария, Варна, 1979, т. I, с. 253-262.

4. К е э в а л л и к А.Э. Теорема декомпозиции конечных автоматов. - Автоматика и вычислительная техника, 1974, № I, с. I7-24.

5. 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 microprogram automata networks. - In: Proceedings of the Second International Symposium on Discrete Systems. Dresden, 1977, vol. 1, p. 120-128.

6. К е э в а л л и к А.Э., Л е й с П.Л. О декомпозиции конечных автоматов. - В кн.: Оптимизация в проектировании дискретных устройств. Л., ЛДНТП, 1976, с. 52-70.

7. 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.

8. Б а р а н о в С.И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л., Энергия, 1979. 232 с.

9. З а к р е в с к и й А.Д., Т о р о п о в Н.Р. Система программирования ЛЯПАС-М. Минск, Наука и техника, 1978. 238 с.

10. ТОМЛЕС - система программирования с базовым языком ЛЯПАС-М для ЕС ЭВМ / С.В. Быкова, М.Н. Головчинер, О.И. Жуковский и др.- Программирование, 1984, № 5, с. 56-63.

A. Keevallik, A. Sudnitsyn, B. Berkman

CAD System for Control Automata  
Decompositional Design

Summary

A description of the CAD system for control automata logical design is given. The methods for decompositional logical design resting on the bases of the system are presented.



В.Р. Аланго, П.А. Китсник, Т.К. Коньт

ПОДХОД К ОРГАНИЗАЦИИ ДАННЫХ ДЛЯ ЛОГИЧЕСКОГО  
ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ

1. Введение. Повышение функциональной сложности элементной базы цифровой техники делает традиционные методы проектирования цифровых систем (ЦС) неприменимыми. Решение проблемы заключается в создании систем автоматизированного проектирования (САПР), в том числе для логического проектирования ЦС.

Логическое проектирование ЦС включает две задачи [1]:

- проектирование структурной реализации цифровых автоматов на заданной элементной базе;
- проектирование тестового обеспечения для полученной реализации.

При машинном решении этих задач существенное значение имеет как внутренняя организация данных проектирования, так и организация их ввода и отладки. Оценка эффективности существующих систем логического проектирования ЦС получена в основном по алгоритмам синтеза и анализа структуры или тестов. При этом мало учтен осуществляемый пользователем трудоемкий процесс подготовки, ввода и отладки данных проектирования. Однако характерно, что для синтеза структуры и тестов требуются минуты или часы машинного времени, а для подготовки и отладки данных - дни и даже недели человеческого времени.

В условиях повышения сложности проектируемых ЦС и их компонентов (БИС и СБИС) пользователю САПР требуется не только входной язык, но и среда проектирования [2]. Последняя включает, в частности, средства интерактивной обработки, поддерживаемые используемой ЭВМ, а также систему управления базой данных. Учитывая это, наиболее перспективной следует считать разработку такой САПР на базе мини-ЭВМ.

Ниже представлены основные положения по организации данных для логического проектирования ЦС.

2. Модель объекта проектирования. При логическом проектировании объектом является дискретная связанная структура (граф алгоритма, логическая схема, соединения структурных компонентов и др.). Рассматриваются воздействия, активизирующие компоненты объекта. Под моделью объекта проектирования подразумевается его структурное и функциональное представление в памяти ЭВМ. В то же время не существенно, описана ли аппаратура или алгоритмы. Этим достигается универсальность модели относительно конкретных объектов и задач проектирования.

Модель объекта проектирования предлагается основывать на его модульном представлении. Согласно положениям работы ГЗ1 ЦС рассматривается состоящая из модулей трех типов:

- структурный модуль;
- функциональный модуль;
- управляемый модуль.

Структурный модуль является составным и представляет собой множество связанных подмодулей. Функциональный модуль является простым (атомарным) и задается описанием его функции. Управляемый модуль представляется также функциональным описанием, но оно задается при помощи специальной структуры данных.

Каждый структурный модуль содержит подмодули, находящиеся на одном иерархическом уровне. Передача воздействий между ними осуществляется в модели через порты.

Для реализации модели объекта проектирования предлагается следующая структура данных. Каждый модуль представляется записью, имеющей вид (фиг. 1). Таблицы в записях различных модулей могут иметь различную длину. Таблицы переменных состояния и подмодулей необязательны. На фиг. 1 использована следующая терминология:

Абсолютное имя - уникальное обозначение модуля, определяющее его однозначно;

Локальное имя - обозначение компонента структурного модуля, имеющее значение только на заданном иерархическом уровне;

Признаки
Абсолютное имя
Таблица портов
Таблица переменных состояния
Таблица подмодулей

Флажки	Сигнал	Имя	Ист.	Пр.1	Пр.2
--------	--------	-----	------	------	------

б)

Локальное имя	П.М.
---------------	------

в)

а)

Фиг. 1. Описание модуля.

а) Запись модуля.

б) Элемент таблицы портов (запись порта).

в) Элемент таблицы подмодулей.

Признаки - данные о типе модуля, характере обработки и текущем состоянии;

Порт - обозначение внешнего соединения модуля;

Флажки - управляющие данные для сканирования модели;

Сигнал - код передаваемого воздействия (интерпретация зависит от типа объекта и задачи);

Имя - локальное обозначение порта;

Ист. - ссылка на порт-источник;

Пр.1 - ссылка на порт-приемник данного иерархического уровня;

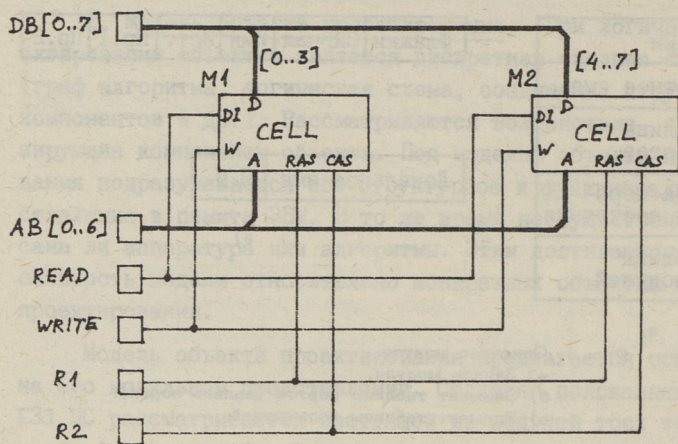
Пр.2 - ссылка на порт нижнего иерархического уровня (в случае неструктурного модуля указывает на запись данного модуля);

П.М. - ссылка на запись подмодуля.

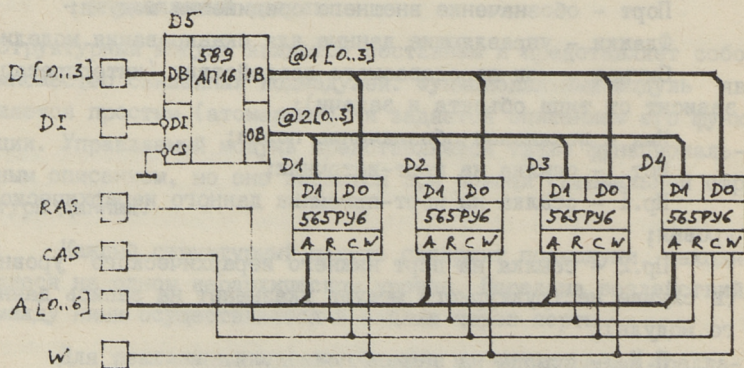
Приведенная структура данных является основной для структурных модулей. При неструктурных модулях вместо таблицы подмодулей задается описание функции модуля или ссылка на такое описание. В зависимости от задачи могут быть введены также дополнительные структуры данных.

Приводим пример структуры предложенной выше модели. Объект в виде принципиальной схемы задан на фиг. 2, а фрагмент соответствующей модели - на фиг. 3. Структурные и функциональные модули обозначены на фиг. 2 соответственно через SM и FM.

## SM: MEMORY



## SM: CELL



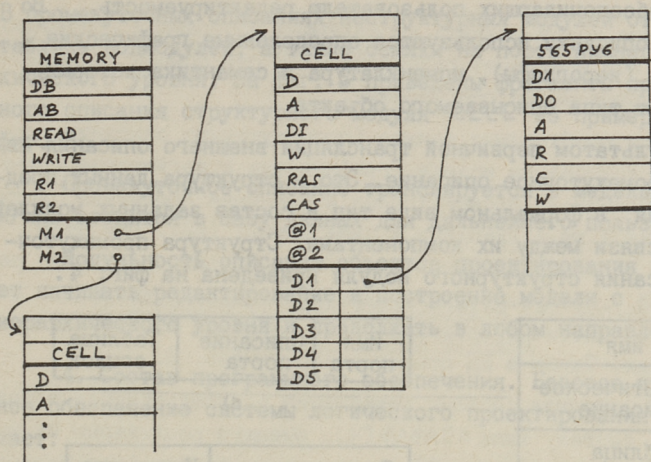
FM: 589 AP16

FM: 565PY6

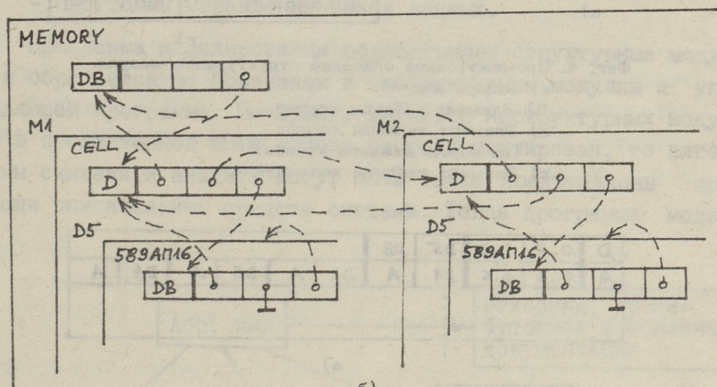
Фиг. 2. Пример.

Приведенная организация ссылок допускает описание объектов, где источником воздействия может служить порт любого модуля и где воздействия могут быть многонаправленными. В случае однонаправленных воздействий (задаются признаком в записи порта) количество ссылок может быть уменьшено.





а)



б)

Фиг. 3. Фрагмент модели к примеру.

а) Записи модулей.

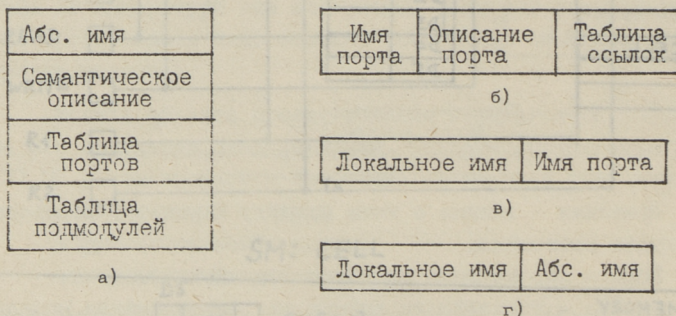
б) Ссылки в записях портов шины DB.

Предложенная модель обеспечивает реализацию алгоритмов синтеза и анализа, базирующихся на активизации и моделировании воздействий по структуре объекта.

3. Внешнее и промежуточное описания. Внешнее описание объекта проектирования реализуется при помощи вербально-графических средств, отражающих специфику предметной об-

ласти и обеспечивающих пользователю редактируемость. Во внешнем описании используются определенные графические примитивы (иероглифы), номенклатура и семантика которых зависит от типа описываемого объекта.

Результатом первичной трансляции внешнего описания является промежуточное описание. Это – структура данных, представляющая в формальном виде тип и состав заданных модулей, а также связи между их компонентами. Структура промежуточного описания структурного модуля приведена на фиг. 4.



Фиг. 4. Промежуточное описание структурного модуля.

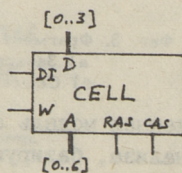
- а) Запись модуля.
- б) Элемент таблицы портов.
- в) Элемент таблицы ссылок (на порты нижнего уровня).
- г) Элемент таблицы подмодулей.

D	0..7	0..3	D5	DB						
A	0..6	0..6	D1	A	D2	A	D3	A	D4	A
		⋮								

а)

D1	K565PУ6
D2	K565PУ6
⋮	
D5	583АП16

б)



в)

Фиг. 5. Пример промежуточного описания.

- а) Таблица портов.
- б) Таблица подмодулей.
- в) Иероглиф.

В промежуточных описаниях неструктурных модулей отсутствуют таблицы подмодулей, а также ссылки на порты нижнего иерархического уровня. На фиг. 5 приведены фрагменты промежуточного описания структурного модуля CELL из примера на фиг. 2.

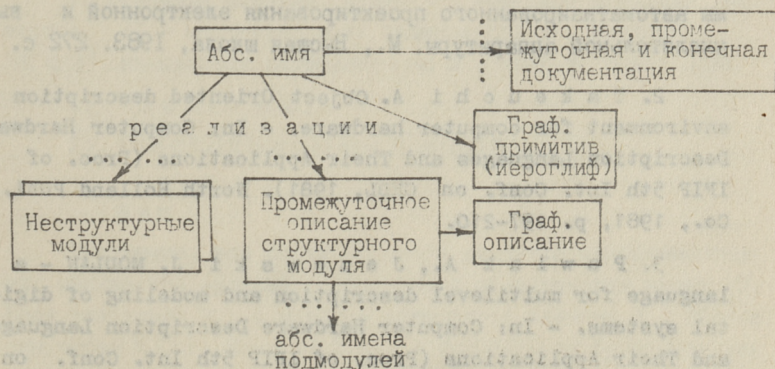
Промежуточное описание транслируется в модель объекта и/или вводится в базу данных для дальнейшего пользования.

Модульность описания объектов проектирования позволяет начинать редактирование и построение модели с любого иерархического уровня и продолжать в любом направлении.

4. Состав программного обеспечения. Базовое программное обеспечение системы логического проектирования ЦС включает:

- управляющую программу;
- программу моделирования;
- программы интерактивной обработки;
- программы управления базой данных.

Программа моделирования обрабатывает структурные модули и обращается по признакам к неструктурным модулям и управляющей программе. Поскольку характер неструктурных модулей в предложенной выше модели не регламентирован, то алгоритмы синтеза и анализа могут быть также представлены при помощи описательных средств системы. Тогда программа моде-



Фиг. 6. Связи в базе данных.

лирования может служить управляющей программой проектирования. Это позволяет расширить САПР при помощи ее базового программного обеспечения.

Все необходимые данные (промежуточные описания модулей, компоненты модели, графические и логические примитивы, исходная, промежуточная и конечная проектная документация и пр.) хранятся в базе данных (БД). При этом с каждым абсолютным именем может быть связано несколько реализаций. Требуемые для конкретной задачи реализации выбираются частично автоматически, частично по запросу пользователя. Связи в БД представлены на фиг. 6.

В качестве системы управления БД предлагается использовать систему типа ADABAS [4], имеющую реализацию на мини-ЭВМ. Такая система БД позволяет отображать как иерархические, так и сетевые структуры данных. Предложенные выше структура модели объекта и его промежуточного описания отображаются в ней естественным образом.

5. Заключение. Комплексное описание современных ЦС затруднено. Это относится также к задачам их логического проектирования на базе БИС и СБИС. В частности, этим и определяется целесообразность предложенного подхода к организации данных.

#### Л и т е р а т у р а

1. Н о р е н к о в И.П., М а н и ч е в В.Б. Системы автоматизированного проектирования электронной и вычислительной аппаратуры. М., Высшая школа, 1983. 272 с.

2. T a k e u c h i A. Object Oriented description environment for computer hardware. - In: Computer Hardware Description Languages and Their Applications (Proc. of IFIP 5th Int. Conf. on CHDL, 1981). North Holland Publ. Co., 1981, p. 197-210.

3. P a w l a k A., J e z e w s k i J. MODLAN - a language for multilevel description and modeling of digital systems. - In: Computer Hardware Description Languages and Their Applications (Proc. of IFIP 5th Int. Conf. on CHDL, 1981). North Holland Publ. Co., 1981, p. 79-93.

4. А т р е Ш. Структурный подход к организации баз данных. М., Финансы и статистика, 1983. 317 с.

V. Alango, P. Kitsnik, T. Kont

An Approach to Data Organization for Logical  
Design of Digital Systems

Summary

In this paper some aspects of computer-aided logical design of digital systems are discussed. Both internal data structures and external description of the design object are based on its modular representation. Data processing will be supported by interactive means of the host computer and an instrumental data base management system. This approach is oriented to the realization in a minicomputer environment.



### КОНТРОЛЬНЫЕ ЭКСПЕРИМЕНТЫ С МИКРОПРОГРАММНЫМИ АВТОМАТАМИ

1. Введение. Проблема проведения контрольных экспериментов с конечными автоматами, заданными либо в табличной форме, либо в форме графа, подробно исследована в [1-3]. Показано, что длина контрольной последовательности (КП), построенной по методу Хенни [1], существенно зависит от длин диагностической и переходных последовательностей (ДП и ПП) [2]. Однако дискретные управляющие устройства реальной сложности обычно не удается описать с помощью классической модели конечного автомата. Поэтому при практическом синтезе целесообразнее использовать модель микропрограммного автомата (МПА) [4], имеющего двоичные входы и выходы и описывающего переходы состояний обобщенными входными сигналами. Задача построения КП для МПА до сих пор не изучена. В данной работе предлагается метод построения КП для МПА, учитывающий его характерные свойства.

2. Основные понятия. Пусть  $S$  - конечное непустое множество,  $2^S$  - множество всех подмножеств  $S$ . Множество  $\pi \subset 2^S$  называется разбиением на  $S$ , если и только если

$$1) \forall B \in \pi [B \neq \emptyset];$$

$$2) (\forall B \in \pi)(\forall B' \in \pi) [B \neq B' \Rightarrow B \cap B' = \emptyset];$$

$$3) \cup B = S.$$

$$B \in \pi$$

Разбиение, все блоки которого являются одноэлементными, называется нулевым разбиением и обозначается через  $0$ . Блок разбиения  $\pi$ , в котором содержится элемент  $x$ , обозначим через  $\pi[x]$ . Если  $x \in B$  и  $y \in B, B \in \pi$ , то будем писать  $x \equiv_{\pi} y$ .

Множество  $\alpha \subset 2^S$  называем  $\alpha$ -группой на множестве  $S$ , если и только если

$$1) \forall B \in \alpha [B \neq \emptyset];$$

$$2) \sum_i |B_i| = |S|.$$

Произведение  $\alpha$ -групп определяем следующим образом:

$$\alpha_1 \cdot \alpha_2 = \{B \cap B' \mid B \in \alpha_1 \& B' \in \alpha_2 \& B \cap B' = \emptyset\}$$

$\alpha$ -группу, все блоки которого являются одноэлементными, называем нулевой и обозначаем через  $O_\alpha$ . Заметим, что любое разбиение также является  $\alpha$ -группой.

МПА (имеющий  $L$  двоичных входов и  $N$  двоичных выходов) определяется как система:

$$A = (S, \{0,1\}^L, \{0,1\}^N, \delta, \lambda),$$

где  $S = \{s_i \mid i = 1, \dots, M\}$  - множество внутренних состояний;

$\{0,1\}^L$  - входной алфавит;

$\{0,1\}^N$  - выходной алфавит;

$\delta: S \times \{0,1\}^L \rightarrow S$  - функция переходов;

$\lambda: S \rightarrow \{0,1\}^N$  - функция выходов.

При построении КП для МПА целесообразно учитывать следующие ее свойства.

Во-первых, каждый переход МПА, имеющий, как правило, большое количество двоичных входов, определен только несколькими из них. Это дает возможность описать компактно целую группу переходов [4]. Во-вторых, по сравнению с соответствующей классической моделью, МПА имеет избыточное число двоичных выходов, что способствует проведению контрольных экспериментов (информация о состояниях автомата легче доступна).

3. Нахождение диагностической последовательности. Алгоритмы нахождения ДП для классической модели конечного автомата опираются, как правило, на использование структурного дерева [2,3]. Этот подход практически не приемлем для модели МПА из-за большого количества входных каналов последнего.

В данной работе предлагается алгоритм нахождения ДП, опирающийся на избыточность выходов МПА.

Определяем разбиение  $\pi_\mu$ :

$$S \equiv_{\pi_\mu} t \Leftrightarrow \lambda(S) = \lambda(t).$$

Обозначим через  $\alpha(\pi_\mu/x)$   $\alpha$ -группу, в которую трансформируется  $\pi_\mu$  при подаче на входы МПА последовательности



входных наборов  $X$ , где

$$X = \{x_1, x_2, \dots, x_s\};$$

$$x_i = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m\};$$

$$\tilde{x}_j \in \{x_j, \bar{x}_j, 1\}.$$

Обозначим  $\alpha^i = \alpha(\pi_\mu/x_1, x_2, \dots, x_i)$ . В частности,  $\alpha^0 = \pi_\mu$ .

Если существует ДП для МПА, состоящий из входных наборов  $\{x_1, \dots, x_p\}$ , то должно выполняться условие

$$\prod_{i=0}^p \alpha^i = 0. \quad (I)$$

Условие (I) выполняется, если входной набор  $x_j$  ( $j=1, \dots, p$ ) не вызывает совмещения состояний [2], которые находятся в одном блоке  $\alpha$ -группы  $\alpha^{j-1}$ . Алгоритм нахождения ДП заключается в следующем:

1. Найдем разбиение  $\pi_\mu$ .

2. Принимаем  $\tau = \pi_\mu$ ;  $i=1$ .

3. Найдем входной набор  $x_i$ , не вызывающий совмещения состояний, которые находятся в одном блоке  $\alpha$ -группы  $\tau$ . Если такого не существует, то перейти к 7.

4. Вычисляем произведение  $\beta = \tau \cdot \alpha(\tau/x_i)$ .

5. Если  $\beta = 0_\alpha$ , то перейти к 8.

6.  $\tau = \beta$ . Перейти к 2.

7. Аварийное завершение.

8. Успешный конец.

Так как реальные МПА имеют обычно значительную избыточность по выходам, то подавляющее большинство из них имеет ДП длиной лишь несколько входных наборов. Также можно предполагать, что для многих МПА ДП состоит из одинаковых входных наборов, что дает возможность значительно укоротить КП, как это показано в следующем разделе.

4. Нахождение КП для МПА. При построении КП для МПА будем в общих чертах следовать методу Хенни [1], используя при этом ранее отмеченные характерные свойства МПА.

Во-первых, рассмотрим случай, если ДП для МПА состоит из  $P$  одинаковых входных наборов  $X$ .

Введем следующие обозначения:

$P(S_i) = \{S_{j1}, S_{j2}, \dots, S_{js}\}$  - множество конечных состояний, каждое из которых соответствует одному переходу из состояния  $S_i$ ;

$W \subseteq S$ , где  $(\forall S_i \in W) [P(S_i) \neq \emptyset]$ ;

ПП(S,t) – переходная последовательность, переводящая автомат из состояния S в состояние t (если ПП(S,t) состоит из одного входного набора, то обозначим его  $\chi(S,t)$ );

$\delta(t/\chi)$  – состояние, в которое перейдет МПА из состояния t под воздействием входного набора  $\chi$ ;

L(S,t) – длина минимальной переходной последовательности из состояния S в состояние t.

Алгоритм построения КП для МПА заключается в следующем.

I. Используя ДП и ПП ( $S_i, S_1$ ), приведем МПА в начальное состояние  $S_1$ ;  $t = S_1$ .

2. Если  $P(t) = \emptyset$ , то перейти к 9.

3. Выбираем переход в состояние  $q_0 \in P(t)$ . Соответствующий входной сигнал  $\chi(t, q_0)$  включаем в КП;  $P(t) = P(t) \setminus q_0$ .

4. Для проверки состояния  $q_0$  включаем ДП в КП. МПА перейдет через состояния  $q_1, q_2, \dots, q_{p-1}$  в состояние  $q_p$ .

5.  $j = 1$ .

6. Если  $q_j \neq q_i, i=0, 1, \dots, j-1$ , то добавим в КП один набор  $\chi$ , проверяя этим переход  $\delta(q_{j-1}/\chi) = q_j$ ;

$P(q_{j-1}) = P(q_{j-1}) \setminus q_j$ . Иначе, перейти к 8.

7.  $j = j + 1$ . Перейти к 6.

8. Обозначим текущее значение МПА через t. Перейти к 2.

9. Если  $P(S_i) = \emptyset$  для  $i=1, \dots, n$ , то перейти к I3.

10. Найдем состояние  $S_k \in W$ , чтобы  $L(t, S_k) = \min_{S_i \in W} [L(t, S_i)]$ .

11. Найдем ПП (t,  $S_k$ ). Включаем ее в КП.

12.  $t = S_k$ , перейти к 3.

13. Конец.

Во-вторых, рассмотрим случаи, если ДП имеет длину 0, т.е. разбиение  $\pi_\mu = 0$ . В таком случае можно использовать упрощенный вариант приведенного алгоритма, в котором отсутствуют шаги 4-7.

В-третьих, если МПА имеет ДП, состоящий из разных входных наборов или не имеет ее вовсе (аварийное завершение алгоритма нахождения ДП), целесообразно ввести в МПА некоторую избыточность для повышения его контролепригодности.

5. Заключение. По предложенному методу и алгоритмам нахождения КП для МПА были составлены программы на языке ЛЯПАС-М для проведения машинного эксперимента. Эксперимент проводился на потоке спроектированных на практике МПА. Усредненные результаты эксперимента приведены в таблице I, где

- М - число состояний МПА;
- Н - длина таблицы переходов МПА;
- Л - число двоичных входов;
- Н - число двоичных выходов;
- Д - средняя длина ДП;
- С - средняя длина КП.

Т а б л и ц а I

М	Н	Л	Н	Д	С
10	30	8	7	1,1	50
20	55	15	15	1,3	110
30	70	34	23	1,7	160
40	100	50	32	2,0	280
50	140	64	37	2,6	390

Эксперимент подтвердил, что использование модели МПА дает возможность построить для реальных ДУУ относительно короткие КП, причем, для большинства МПА можно построить ДП, состоящий из одинаковых входных наборов. Проведение контрольных экспериментов для модели МПА облегчает диагностику ДУУ на схемном уровне.

#### 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. - In: Proc. Fifth Ann. Symp. on Switching Circuit Theory and Logical Design, 1964, p. 95-110.
2. Г и л л А. Введение в теорию конечных автоматов. М., Наука, 1966.
3. F r i e d m a n A.D., M e n o n P.R. Fault detection in digital circuits. Englewood Cliffs, N.-Y. Prentice-Hall, 1971.
4. Б а р а н о в С.И. Синтез микропрограммных автоматов. Л., Энергия, 1979.

Checking Experiments for Microprogram Automata

Summary

The checking experiments design method for microprogram automata is discussed. The algorithms for distinguishing and checking sequences generation are described. It is shown that usually these sequences are very short for microprogram automata. The results of computer experiments are presented.

ГЕНЕРИРОВАНИЕ УНИВЕРСАЛЬНЫХ ТЕСТОВ ДЛЯ  
ДИСКРЕТНЫХ УСТРОЙСТВ НА АЛЬТЕРНАТИВНЫХ ГРАФАХ

I. Введение. Традиционные постановки задач синтеза тестов для дискретных устройств основываются на задании описания конкретной структурной схемы устройства. Синтезируемые тесты должны гарантировать обнаружение всех неисправностей, возможных в заданной структурной схеме.

Имеются и попытки синтеза т.н. универсальных тестов, синтезируемых на основе заданной функции и независимых от конкретной структурной реализации устройства [1, 2]. Ценность таких тестов, в первую очередь, заключается в том, что они применимы для широкого диапазона конкретных реализаций одной и той же функции. Кроме того, в таких случаях, когда внутренняя структура дискретного объекта (например, интегральной схемы) неизвестна, функциональный подход к синтезу тестов является единственным.

Существует тривиальное решение задачи получения универсальных тестов – полный перебор всех возможных входных воздействий. Но для инженерной практики такое решение применимо всего лишь в редких случаях для несложных объектов. В общем случае требуется найти некоторое компромиссное решение между двумя крайностями – полным перебором тестов и универсальностью в полном смысле. Это приводит к поиску приемлемой длины теста, являющегося универсальным при определенных ограничениях на классы рассматриваемых устройств и неисправностей.

В работе [1] показаны возможности синтеза универсальных тестов на основе заданной булевой функции, обнаруживающих все одиночные и кратные константные неисправности в любой И/ИЛИ-реализации комбинационной схемы, т.е. в схеме, в которой используются только элементы И, ИЛИ и НЕ, а ин-

верторы могут находиться только на входе. Предложенный в работе [1] алгоритм трудоемок, так как базируется на попарном сравнении строк таблицы истинности заданной функции.

В работе [2] предложены тесты для обнаружения константных неисправностей и коротких замыканий на входах и выходах произвольных дискретных устройств, приведены оценки длины этих тестов, а также их полноты. Однако то, что предлагаемые в [2] тесты не предназначены для обнаружения внутренних дефектов дискретных устройств, существенно снижает их практическое значение.

В данной статье предлагается метод синтеза универсальных тестов, предназначенных для обнаружения произвольных кратных константных неисправностей в комбинационных схемах, построенных на элементах И и ИЛИ, с возможными инверторами на входах. Предлагаемые алгоритмы основываются на представлении булевой функции альтернативным графом [3] и обладают меньшей трудоёмкостью, чем алгоритмы, предложенные в [1].

2. Основные понятия. Рассмотрим некоторое комбинационное устройство, заданное булевой функцией  $z = F(z_1, \dots, z_n)$ . Обозначим через  $\gamma$  - входной набор устройства,  $r(\gamma)$  - значение разряда  $r$  в наборе  $\gamma$  и  $F(\gamma)$  - выходную реакцию устройства на этот набор.

Определение 1. Набор  $\gamma_1$  покрывает набор  $\gamma_2$  (обозначим это через  $\gamma_1 \geq \gamma_2$ ) тогда, когда для всех разрядов имеет место

$$r_i(\gamma_1) \geq r_i(\gamma_2), \quad i = \overline{1, n}.$$

Определение 2. Наборы  $\gamma_1$  и  $\gamma_2$  несравнимы, если существует два разряда  $r_i$  и  $r_j$ , для которых

$$r_i(\gamma_1) > r_i(\gamma_2),$$

$$r_j(\gamma_1) < r_j(\gamma_2).$$

Предлагаемый метод синтеза универсальных тестов для И/ИЛИ-схем основывается на следующих двух свойствах этих схем [1].

Свойство монотонности. Для любых частично упорядоченных входных наборов  $\gamma_1 \geq \gamma_2$  соответствующие выходные реакции схемы удовлетворяют неравенству  $F(\gamma_1) \geq F(\gamma_2)$ .

Свойство сохранения монотонности. Если схема  $F$  обладает свойством монотонности, то она сохраняет это свойство при любом сочетании константных неисправностей в схеме типа  $\equiv 0$  и  $\equiv 1$ .

Пусть  $\hat{Z}$  - упорядоченное множество существенных литералов монотонной функции  $z = F(Z)$ , где  $Z = \{z_1, z_2, \dots, z_n\}$ . Формальным правилом для включения литерала  $\hat{z}_i \in \{\bar{z}_i, z_i\}$  в множество  $\hat{Z}$  служит условие

$$\partial F / \partial \hat{z}_i = 1. \quad (1)$$

Так, например, для функции

$$z = (z_1 \vee \bar{z}_2 \vee \bar{z}_4)(z_3 \vee z_2 \bar{z}_4) \quad (2)$$

имеем  $\hat{Z} = \{z_1, z_2, \bar{z}_2, z_3, \bar{z}_4\}$ .

Пусть  $Z^* \subseteq Z$  - подмножество переменных, относительно которых функция  $z = F(Z)$  юнатна, а  $\hat{Z}^* = \{\hat{z}_i | i: z_i \in Z^*\} \subseteq \hat{Z}$  - соответственно, множество литералов, представляющих эти переменные в выражении функции. Так, например, для функции (2) имеем  $Z^* = \{z_1, z_3, z_4\}$ ,  $\hat{Z}^* = \{z_1, z_3, \bar{z}_4\}$ . Для юнатной функции имеет место  $Z^* = Z$ , для бинатной функции  $Z^* = \emptyset$ .

Отметим, что все реализации комбинационных схем, образующие класс, для которого синтезируемые тесты универсальны, должны использовать одинаковое множество входных литералов  $\hat{Z}$ .

Обозначим через  $\alpha = \{\alpha_i\}$  множество наборов непротиворечивых значений литералов множества  $\hat{Z}$ , для которых  $F(\alpha_i) = 1$ , и через  $\beta = \{\beta_j\}$  - множество наборов, для которых  $F(\beta_j) = 0$ .

Из свойства сохранения монотонности И/ИЛИ-схем в случае константных неисправностей вытекают следующие правила, которые можно использовать для отсеивания избыточных тестов.

Правило 1. Если  $\alpha_i > \alpha_j$ , то набор  $\alpha_j$  можно исключить из множества тестовых наборов.

Правило 2. Если  $\beta_i > \beta_j$ , то набор  $\beta_j$  можно исключить из множества тестовых наборов.

Метод синтеза тестов, предложенный в [1], основывается на попарном переборе всех сравнимых входных наборов и

отсеивании путем применения правил 1 и 2 всех покрывающих наборов  $\alpha_i$  и всех покрываемых наборов  $\beta_j$ .

Лемма I. Для выделения покрывающих наборов  $\alpha_i$  (или покрываемых наборов  $\beta_j$ ) значений литералов множества  $\hat{Z}$  достаточно провести попарное сравнение и применение правила 1 (или 2) для соседних наборов  $\alpha_i \in \alpha$  (или  $\beta_j \in \beta$ ) относительно разрядов  $Z_s \in \hat{Z}^*$ .

Доказательство. Так как для любого разряда  $r_s \in \hat{Z} \setminus \hat{Z}^*$  существует всегда другой разряд  $r_t \in \hat{Z} \setminus \hat{Z}^*$ , имеющий противоположное значение  $r_t = \bar{r}_s$ , то, согласно определению 2, два набора  $\alpha_i$  и  $\alpha_j$  (или  $\beta_i$  и  $\beta_j$ ) сравнимы только тогда, когда

$$r_s(\alpha_i) = r_s(\alpha_j) \quad (\text{или } r_s(\beta_i) = r_s(\beta_j)).$$

а следовательно, для соседних наборов  $\alpha_i(\beta_j)$  относительно разрядов  $r_s \in \hat{Z} \setminus \hat{Z}^*$  никогда не могут быть выполнены условия правила 1 (или 2).

Легко заметить, что утверждение леммы I позволяет сократить перебор необходимых попарных сравнений наборов во множествах  $\alpha$  и  $\beta$ .

3. Синтез тестов на модели альтернативных графов. Рассмотрим альтернативный граф (АГ), представляющий булеву функцию  $\bar{z} = F(Z)$ , где  $Z = \{z_1, \dots, z_n\}$ , как некоторый ориентированный граф  $G = (M, \Gamma)$ , вершины  $m \in M$  которого взвешены переменными  $z(m) \in Z$ , а  $\Gamma$  - отображение множества  $M$  во множество  $M$  [3].

Каждому пути  $\ell$  в АГ из начальной вершины в некоторую терминальную вершину  $m_\kappa \in M$  соответствует множество  $M(\ell) = M$  пройденных вершин, множество переменных

$$Z(\ell) = \{z(m) \mid m \in M(\ell)\} \subseteq Z,$$

множество литералов

$$\hat{Z}(\ell) = \{\hat{z}(m) \mid m \in M(\ell)\} \subseteq \hat{Z},$$

входной набор  $\gamma(\ell)$  (если  $Z(\ell) = Z$ ) или подмножество входных наборов  $\{\gamma\}_\ell$  (если  $Z \setminus Z(\ell) \neq \emptyset$ ), а также соответствующее этому пути значение функции  $F(\ell)$ , равное весу терминальной вершины  $F(\ell) = \hat{z}(m_\kappa)$ .

Задачей синтеза теста на модели АГ является отбор некоторого множества тестовых путей и соответствующих им



входных наборов. Для этого необходимо пройти все пути на графе и при каждом из них решить вопрос включения или исключения его из множества тестовых путей.

Определение 3. Назовем альтернативный граф  $G=(M, \Gamma)$  универсальным (УАГ) относительно представляемой им функции  $z = F(Z)$  со множеством существенных литералов  $\hat{Z}$ , если для всех вершин  $m \in M$ , где  $z(m) \in Z^* \subseteq Z$ , имеет место  $\hat{z}(m) \in \hat{Z}^* \subseteq \hat{Z}$ .

При построении АГ для некоторой функции  $z = F(Z)$  со множеством существенных литералов  $\hat{Z}$  в общем случае могут быть применены структурный подход (метод суперпозиции графов) и функциональный подход (метод разложения функции по переменным) [3]. При структурном подходе все литералы  $\hat{z} \in \hat{Z}$ , используемые в исходном выражении функции  $z = F(Z)$ , становятся непосредственно весами  $\hat{z}(m)$  вершин в синтезированном графе. При функциональном подходе к синтезу АГ в качестве весов можно зафиксировать вершинам  $m$  произвольные литералы  $\hat{z}(m)$ . При синтезе универсальных графов необходимо выполнять требования, сформулированные в определении 3.

Лемма 2. Набор  $\gamma_i \in \{\gamma\}_\ell$ , порождаемый путем  $\ell$  на УАГ  $G = (M, \Gamma)$  для функции  $z = F(Z)$ , можно исключить из множества наборов универсального теста функции  $z = F(Z)$ , если хотя бы для одного литерала  $\hat{z} \in \hat{Z}^* \setminus \hat{Z}(\ell)^*$  имеет место  $\hat{z}(\gamma_i) = F(\ell)$ , где  $\hat{z}(\gamma_i)$  — значение литерала  $\hat{z}$  на наборе  $\gamma_i$ .

Доказательство. Действительно для такого набора, из-за условия  $\hat{z} \in \hat{Z}(\ell)^*$  всегда существует другой набор  $\gamma_j \in \{\gamma\}_\ell$ , для которого

$$\forall z_\kappa \in Z \setminus z : z_\kappa(\gamma_i) = z_\kappa(\gamma_j)$$

и

$$\hat{z}(\gamma_i) > \hat{z}(\gamma_j)$$

для случая, где  $F(\ell) = 1$

и

$$\hat{z}(\gamma_i) < \hat{z}(\gamma_j)$$

для случая, где  $F(\ell) = 0$ . Следовательно, согласно правилам 1 и 2, набор  $\gamma_i$  можно исключить из множества тестовых наборов.

Определение 4. Назовем существенными наборами пути  $\ell$  наборы  $\gamma_i \in \{\gamma\}_\ell$ , в которых для всех литералов  $\hat{z} \in \hat{Z}^* \setminus \hat{Z}(\ell)^*$  имеет место  $\hat{z}(\gamma_i) \neq F(\ell)$ .

Легко заметить, что путь  $\ell$  порождает единственный существенный набор тогда, когда этот набор полностью определен, т.е. при  $Z(\ell) = Z$ , а также тогда, когда функция юната относительно неопределенных переменных, т.е. при  $Z \setminus Z(\ell) \subset Z^*$ .

Рассмотрим некоторый путь  $\ell$  в АГ, а также существенный набор пути  $\gamma$ . Берем произвольную вершину  $m \in M(\ell)$  на этом пути. Изменим значение веса  $\hat{z}(m)$  на противоположное и обозначим получаемый из  $\ell$  новый путь через  $\ell(m)$ .

Лемма 3. Существенный набор  $\gamma \in \{\gamma\}_\ell$  пути  $\ell$  на модели УАГ  $G = (M, \Gamma)$  для функции  $z = F(Z)$  можно исключить из множества наборов универсального теста функции  $z = F(Z)$  тогда, когда имеет место

$$\exists m \in M(\ell)^* : F(\ell) = F(\ell(m)), \quad (3)$$

где

$$M(\ell)^* = \{ m | m \in M(\ell), \hat{z}(m) \in \hat{Z}^*, \hat{z}(m) = F(\ell) \}.$$

Доказательство. Согласно правилам 1 и 2, критерием отсеивания набора  $\gamma \in \{\gamma\}_\ell$  служит существование покрывающего  $\gamma$  набора  $\alpha_i \in \alpha$ , если  $F(\ell) = 1$ , или покрываемого набором  $\gamma$  набора  $\beta_j \in \beta$ , если  $F(\ell) = 0$ . Рассмотрим набор  $\gamma$  в трех частях. В разрядах, соответствующих литералам  $\hat{z} \in \hat{Z} \setminus \hat{Z}^*$ , согласно лемме 1, набор  $\gamma$  с другими наборами несравним. В разрядах, соответствующих литералам  $\hat{z} \in \hat{Z} \setminus \hat{Z}(\ell)$ , сравнение не нужно, согласно лемме 2 и определению 4. Для оставшихся разрядов предпосылкой для существования другого покрывающего набора (при  $F(\ell) = 1$ ) или покрываемого набора (при  $F(\ell) = 0$ ) служит условие  $\hat{z}(m) = F(\ell)$ . Выполнение условия (3) является признаком того, что сравнимые наборы относятся к одному и тому же множеству, ко множеству  $\alpha$ , если  $F(\ell) = 1$ , или ко множеству  $\beta$ , если  $F(\ell) = 0$ . Лемма доказана.

Обозначим через  $L$  множество всех возможных путей на заданном графе, а через  $\gamma(L)$  - множество существенных наборов, порождаемых путями  $\ell \in L$ . Тогда можно сформулировать следующую теорему.

Теорема. Множество  $\gamma(L)$  является универсальным тестом функции  $z = F(z)$ , представленной универсальным АГ  $G = (M, \Gamma)$ , тогда, когда для всех путей  $\ell \in L \setminus L'$  удовлетворено условие (3) леммы 3.

Из теоремы вытекает следующий алгоритм синтеза универсальных тестов комбинационных схем на модели АГ.

### Алгоритм

1<sup>0</sup>. Построить очередной путь  $\ell$  на АГ.

2<sup>0</sup>. Определить существенный набор  $\gamma$  для построенного пути.

3<sup>0</sup>. Включить набор  $\gamma$  в универсальный тест, если условие (3) леммы 3 не удовлетворено.

4<sup>0</sup>. Если не все пути  $\ell \in L$  построены, перейти к п. 1<sup>0</sup>, иначе конец.

4. Оценка эффективности предлагаемого метода. Алгоритм синтеза универсальных тестов, предложенный в [1], базируется на попарном сравнении всех возможных наборов в таблице истинности заданной функции. Из этого вытекает следующая верхняя оценка числа операций попарных сравнений

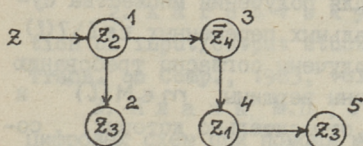
$$N_1 \leq \frac{n(n-1)}{2} = O(n^2), \quad (4)$$

где  $n$  – число строк в таблице истинности заданной функции.

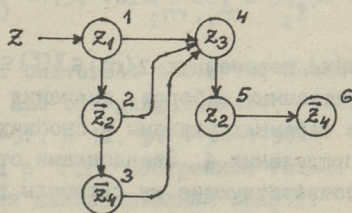
Для предлагаемого подхода трудоемкость работы определяется двумя показателями: максимальным числом  $L \leq n$  путей в графе и максимальным числом вершин  $p \leq |Z^*|$  на одном пути. Так как каждая такая вершина на каждом пути в общем случае может порождать операцию сравнения двух наборов, то получим следующую верхнюю оценку числа операций

$$N_2 \leq n|Z^*| \leq n \log_2 n = O(n \log_2 n), \quad (5)$$

которая заметно ниже, чем оценка  $N_1$  для известного метода [1].



Фиг. 1.



Фиг. 2.

Так как соответствие между заданной функцией  $z = F(Z)$  и ее моделью УАГ неоднозначно, то с целью повышения эффективности предложенного метода представляет интерес задача синтеза альтернативных графов с минимальным числом путей  $L$ . Так, например, на фиг. 1 и 2 приведены два УАГ, представляющих функцию (2). Граф, представленный на фиг. 1, получен по функциональному подходу и имеет 6 путей, а граф на фиг. 2 составлен по структурному подходу и включает 9 различных путей. Верхней оценкой путей для данной функции является  $n = 16$ .

5. Пример. Рассмотрим применение метода синтеза универсальных тестов для функции (2) на примере двух УАГ, представленных на фиг. 1 и 2. Процессы синтеза тестов на этих графах иллюстрированы, соответственно, в табл. 1 и 2, где перечислены для каждого пути на графе пройденные вершины, существенные наборы и принятые в универсальный тест тестовые наборы. Каждый путь  $\ell$  порождает определенные зна-

Т а б л и ц а 1

№	Путь $\ell_i$	Пройденные вершины $M(\ell_i)$	Наборы $\gamma$				Тестовые наборы				
			$z_1$	$z_2$	$z_3$	$\bar{z}_4$	$z_1$	$z_2$	$z_3$	$z_4$	$z$
1	$\ell_1$	$m_1, m_3^*$	(0)	1	(0)	$1^X$	0	1	0	0	1
2	$\ell_2$	$m_1, m_3, m_4^*, m_5^*$	$1^X$	1	$1^X$	0	1	1	1	1	1
3	$\ell_3$	$m_1, m_3^*, m_4, m_5^*$	1	1	$0^X$	$0^X$	1	1	0	1	0
4	$\ell_4$	$m_1, m_3^*, m_4^*$	$0^X$	1	(1)	$0^X$	0	1	1	1	0
5	$\ell_5$	$m_1, m_2^*$	(0)	0	$1^X$	(0)	0	0	1	1	1
6	$\ell_6$	$m_1, m_2^*$	(1)	0	$0^X$	(1)	1	0	0	0	0

чения переменных  $z(m) \in Z(\ell) \subseteq Z$ . Для получения множества существенных наборов, значения остальных переменных  $z \in Z \setminus Z(\ell)$  (в таблицах указаны в скобках) получены согласно требованию определения 4. Звездочками отмечены вершины  $m \in M(\ell)^*$  и соответствующие им литералы  $\hat{z}(m)$ , относительно которых, согласно лемме 3, проверяется необходимость включения набора в универсальный тест. Так, например, для графа фиг. 1 все существенные наборы, порождаемые путями графа, являются и тестовыми наборами. Число проведенных операций (число звез-

дочек в табл. наборов  $\gamma$ ) для первого графа  $N' = 9$ , для второго графа  $N'' = 17$ . Верхней оценкой, согласно (5), является  $N_2 = 48$ , так как  $Z^* = \{z_1, z_3, z_4\}$  и  $|Z^*| = 3$ . Заметим, что тестовые наборы, полученные на этих примерах, одинаковы и соответствуют универсальному тесту, полученному для этой функции в работе [1]. Число попарных сравнений, требуемое для этой задачи при методе [1], согласно (4), равно  $N_1 = 120$ , так как  $n = 16$ .

Т а б л и ц а 2

№	Путь $l_i$	Пройденные вершины $M(l_i)$	Наборы $\gamma$				Тестовые наборы				
			$z_1$	$z_2$	$z_3$	$\bar{z}_4$	$z_1$	$z_2$	$z_3$	$z_4$	$z$
1	$l_1$	$m_1^*, m_4^*$	$I^X$	(0)	$I^X$	(0)					I
2			$I^X$	(1)	$I^X$	(0)	I	I	I	I	I
3	$l_2$	$m_1^*, m_4, m_5, m_6^*$	$I^X$	I	0	$I^X$					I
4	$l_3$	$m_1, m_4^*, m_5, m_6^*$	I	I	$0^X$	$0^X$	I	I	0	I	0
5	$l_4$	$m_1, m_4^*, m_5$	I	0	$0^X$	(I)	I	0	0	0	0
6	$l_5$	$m_1, m_2, m_4^*$	0	0	$I^X$	(0)	0	0	I	I	I
7	$l_6$	$m_1^*, m_2, m_4^*, m_5$	$0^X$	0	$0^X$	(I)					0
8	$l_7$	$m_1, m_2, m_3^*, m_4^*$	0	I	$I^X$	$I^X$					I
9	$l_8$	$m_1, m_2, m_3^*, m_4, m_5, m_6^*$	0	I	0	$I^X$	0	I	0	0	I
10	$l_9$	$m_1^*, m_2, m_3^*$	$0^X$	I	(I)	$0^X$	0	I	I	I	0

#### Л и т е р а т у р а

1. A k e r s S.B. Universal test sets for logic networks. - IEEE Trans. on Comp., 1973, vol. C-22, N 9, p. 835-839.

2. К а р п о в с к у М. Universal tests for detection of input/output stuck-at and bridging faults. - IEEE Trans. on Comp., 1983, vol. C-32, N 12, p. 1194-1197.

3. П л а к к М.П., У б а р Р.Р. Построение тестов цифровых схем при помощи модели альтернативных графов. - Автоматика и телемеханика, 1980, № 5, с. 152-163.

Herstellung von universalen Tests für digitale  
Einrichtungen auf dem Modell von alternativen Graphen

Zusammenfassung

Es wird ein Verfahren zur Synthese von universalen Tests für die Entdeckung von beliebigen mehrfachen konstanten Fehlern in kombinatorischen AND/OR-Schaltungen mit möglichen Invertoren in Eingängen vorgeschlagen. Die Tests sind unabhängig von der Realisation der zu prüfenden Schaltung. Die vorgeschlagenen Algorithmen beruhen auf der Darstellung der Boole'schen Funktionen mit alternativen Graphen und sind leistungsfähiger als bekannte Algorithmen.

### О МОДЕЛИРОВАНИИ ДЛИННЫХ ВХОДНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ В ДИСКРЕТНЫХ УСТРОЙСТВАХ, СОДЕРЖАЩИХ СЧЕТНЫЕ СТРУКТУРЫ

Известные подходы к моделированию дискретных устройств, базирующиеся на алгебре логики, не применимы в случае длинных входных последовательностей, состоящих из сотен тысяч и миллионов тактов. Такие последовательности типичны для дискретных устройств, содержащих счетные структуры.

Рассматривается новый подход к моделированию "длинных" последовательностей, где логические методы анализа потенциальных сигналов сочетаются с арифметической обработкой числа фронтов в импульсных пачках. Предлагаемый метод является дальнейшим развитием идеи, изложенной в работах [1, 2].

1. Сигналы и переменные. Рассмотрим дискретное устройство (ДУ) в виде некоторого множества функций непосредственных связей на множестве переменных

$$Z = Z_{вх} \cup Z_{вых} \cup Z_{вн},$$

где  $Z_{вх}, Z_{вых}$  и  $Z_{вн}$  - соответственно, множества входных, выходных и внутренних переменных устройства.

Каждой функции соответствует некоторая одновходная под-схема или элемент памяти. Выделим в ДУ два класса одновходных подсхем - логические и счетные подсхемы. При декомпозиции исходного ДУ на счетные и логические подсхемы образуются следующие непересекающиеся подмножества переменных:

$Z_l \subset Z_{вн} \cup Z_{вых}$  - подмножество логических переменных, описывающих поведение логических подсхем, и  $Z_{сч} \subset (Z_{вн} \cup Z_{вых}) \setminus Z_l$  - подмножество переменных, описывающих поведение счетных подсхем.

Целесообразно определить еще подмножество

$$Z^n = Z^n_l \cup Z^n_{сч} \subset Z_{bn},$$

где  $Z^n_l \subset Z_l$  - подмножество логических переменных памяти в логических подсхемах,

$Z^n_{сч} \subset Z_{сч}$  - подмножество переменных памяти в счетных подсхемах.

Далее исходим из положения, что сигналы в любой точке схемы, описываемой некоторой переменной  $z_k \in Z \setminus Z^n$ , могут быть как потенциальными, так и произвольными импульсными пачками. Для задания функций логических подсхем в условиях, когда описываемые сигналы могут быть как потенциальными, так и импульсными пачками, используем трехзначный алфавит  $A^3 = \{0, 1, X\}$ , в котором 1 и 0 представляют потенциалы, а X - соответствует случаю, когда через рассматриваемую точку в течение рассматриваемого такта  $t$  передается некоторая последовательность фронтов (изменений потенциала). Переменные  $z_k \in Z \setminus Z^n$  будут принимать значения из алфавита  $A^3$ .

Для уточненного описания поведения ДУ разделим трехзначные сигналы  $z_k \in Z \setminus Z^n$  на составляющие

$$z_k = \{z_k^c, z_k^d\},$$

где  $z_k^c$  - статический компонент сигнала  $z_k$ , представляющий установившееся значение потенциала в конце рассматриваемого такта и принимающий значения из двоичного алфавита  $A^2 = \{0, 1\}$ ,

$z_k^d$  - динамический компонент сигнала  $z_k$ , принимающий значения из области натуральных чисел и определяющий число фронтов (изменений потенциала) в течение рассматриваемого такта.

Под тактом понимаем дискретное время тестера, в течение которого на входы контролируемого объекта подается очередной набор, причем, согласно определению сигналов, наборы могут состоять из потенциалов и импульсных пачек.

Согласно определению переменных  $z_k \in Z \setminus Z^n$  можем писать

$$z_k = \begin{cases} z_k^c, & \text{если } z_k^d = 0, \\ X, & \text{если } z_k^d \neq 0. \end{cases} \quad (I)$$

Обозначим через  $Z^c = \{z_k^c\}$  и  $Z^d = \{z_k^d\}$ , соответственно, множества переменных, описывающих, соответственно, статические и динамические компоненты сигналов  $z_k \in Z \setminus Z^n$ .



Переменные  $z_k^n \in Z_l^n$  имеют двоичные значения, а переменные  $z_k^n \in Z_{сч}^n$  принимают значения из областей натуральных чисел  $A^{n_k} = \{0, 1, \dots, n_k - 1\}$ , где  $n_k$  - период счетчика (число фронтов, через которое его внутреннее состояние повторяется).

2. Функции ДУ со счетными подсхемами. Описание логических подсхем задается множеством логических функций

$$z_k = \lambda_k(Z_k), \quad z_k \in Z_l \setminus Z_l^n \quad (2)$$

и множеством функций переходов для элементов памяти

$$z_k = \delta_k(Z_k'), \quad z_k \in Z_l^n, \quad (3)$$

где  $Z_k \subset Z \setminus Z_{сч}^n$  - подмножество входных переменных и переменных элементов памяти рассматриваемой подсхемы, от которых зависит переменная функции  $z_k$ .

Здесь штрих означает, что значение переменной берется из конца предыдущего такта.

Каждая счетная подсхема с выходной переменной  $z_k \in Z_{сч} \setminus Z_{сч}^n$  и с переменной памяти  $z_k^n \in Z_{сч}^n$  задается системой из двух логико-арифметических функций

$$z_k^n = f_k^n(Z_k^A, z_k^{n'}; Z_k), \quad (4)$$

$$z_k^A = f_k^A(Z_k^A, z_k^{n'}; Z_k), \quad (5)$$

для описания, соответственно, функций переходов и динамического выходного поведения счетчика, и из предикатной функции

$$z_k^c = f_k^c(z_k^n), \quad (6)$$

для описания статического выходного поведения счетчика в конце рассматриваемого такта. Здесь  $Z_k \subset Z \setminus Z_{сч}^n$  - множество логических переменных, определяющих режим работы счетчика или служащих для поразрядной установки счетчика,

$Z_k^A$  - множество переменных, соответствующих счетным входам счетчика, а  $z_k^{n'}$  - переменные состояния счетчика.

Функции (4) и (5) в общем случае принимают вид следующих условных выражений, соответственно,

$$z_k^A = f_{k,i}^A, \text{ если } \varphi_{k,i}(Z_k) = 1, \quad k: z_k \in Z_{сч} \setminus Z_{сч}^n, \quad (7)$$

$$\text{и } z_k^n = f_{k,i}^n, \text{ если } \varphi_{k,i}(Z_k) = 1, z_k^n \in Z_{сч}^n, \quad (8)$$

для  $i = 1, 2, \dots, m_k$ , где  $m_k$  - число различных режимов, в которых может работать соответствующая счетная схема.

Каждому логическому условию  $\varphi_{k,i}(Z_k)$  соответствует определенный режим работы, описываемый функциями  $f_{k,i}^A$  и  $f_{k,i}^n$ , которые могут быть

- а) константами (в режимах установки);
- б) арифметическими зависимостями вида  $f_{k,i}(z_{k,i}^A, z_k^n)$ ,

описывающими режимы счета фронтов на некотором входе  $z_{k,i}^A \in Z_k^A$  при исходном состоянии  $z_k^n$  памяти схемы;

в) функциями специального вида  $f_{k,i}(Z_k)$  для перевода значений двоичных переменных  $z \in Z_k$ , соответствующих поразрядным установочным входам схемы, в соответствующее значение переменной памяти  $z_k^n$ , взятое из области  $A^{n_k}$ .

Для того, чтобы гарантировать выполнение не более одного режима, условия  $\varphi_{k,i}$  должны обладать свойством ортогональности

$$\forall i, j = \overline{1, m_k}: \varphi_{k,i}(Z_k) \wedge \varphi_{k,j}(Z_k) = 0. \quad (9)$$

В случае, когда значение  $\varphi_{k,i}(Z_k)$  не определено (т.е.  $\varphi_{k,i}(Z_k) = X$ ), не могут быть определены из выражений (6) и (7) также значения переменных  $z_k^A$  и  $z_k^n$ . Этот случай рассматривается ниже.

Функции (2) и (3) по определению выражают только булевы зависимости между переменной  $z_k$  и переменными  $z \in Z_k$ . Следовательно, при помощи функций (2) могут быть определены в принципе только статические компоненты выходного сигнала  $z_k^c$ . В случае, когда на входы подсхемы, описываемой функцией (2), поступают импульсные последовательности, поведение схемы не может быть определено традиционными методами булева исчисления. Для вычисления динамического компонента выходного сигнала  $z_k^A$  разработаны приведенные ниже соответствующие алгоритмы, базирующиеся на представлении функций (2)-(6) моделью альтернативных графов (АГ).

3. Модель АГ. Рассмотрим альтернативный граф, представляющий некоторую функцию  $z_k = f_k(Z_k)$  в виде  $G_k = \{M_k, \Gamma_k\}$ , где  $M_k$  - множество вершин графа,  $\Gamma_k$  - отображение множества

$M_k$  во множество  $M_k$  [3]. Каждой нетерминальной вершине  $m \in M_k$ , где  $\Gamma_m \neq \emptyset$ , сопоставлен вес  $e(m)$ , которым является некоторая булева переменная  $z(m) \in Z_k$  или некоторый предикат  $P(Z_k)$ . В терминальных вершинах  $m \in M_k$ , где  $\Gamma_m = \emptyset$ , весами  $e(m)$  служат переменные  $z(m) \in Z_k$  или некоторые функции  $\varphi_m(Z_k)$ . Для булевых переменных  $z_k$  веса терминальных вершин графа  $G_k$  также булевы, для небулевых переменных  $z_k$  веса терминальных вершин небулевы.

Значение переменной  $z_k$  может быть определено путем логического моделирования в виде процесса движения по вершинам графа  $G_k$  так, чтобы направление выхода из каждой вершины  $m$  определялось значением  $e(m)$  в одно из двух направлений. Искомое значение функции находится в терминальной вершине  $m_k$  пройденного пути и равно значению ее веса  $e(m_k)$ .

В АГ, представляющих булевы функции вида (2) и (3), все вершины взвешены булевыми переменными  $z \in Z_k$ . В АГ, представляющих небулевы функции вида (4) и (5), нетерминальные вершины взвешены булевыми переменными  $z \in Z_k$ , а терминальные — небулевыми функциями вида  $f_{k,i}^A$  или  $f_{k,i}^N$ . В АГ, представляющих булевы функции вида (6), все вершины взвешены предикатами.

Легко заметить, что описанный метод вычисления значений переменных  $z_k$ ,  $z_k^n$  и  $z_k^A$  на альтернативных графах приводит к однозначному результату только в том случае, когда все значения весов  $e(m)$  на пройденном пути принимают двоичные значения  $e(m) \in \{0, 1\}$ . В случае, когда для хотя бы одной вершины  $m$  имеет место  $e(m) = X$ , значение вычисляемой переменной будет неопределенным.

В разделе 5 будет предложен метод моделирования цифровых схем на модели АГ в общем случае, когда веса  $e(m)$  могут иметь неопределенные значения  $X$ .

4. Выражения для вычисления поведения счетных подсхем.  
Рассмотрим теперь более подробно некоторые соотношения, позволяющие упростить процедуры вычисления поведения счетных подсхем. Эти соотношения могут служить весовыми функциями терминальных вершин соответствующих АГ.

Рассмотрим работу счетной подсхемы с выходом  $z_k$  и памятью  $z_k^n$  в режиме, где в такте  $t$  происходит счет импульсов по определенному входу  $z_i^A \in Z_k^A$ .

Легко заметить, что состояние счетчика  $z_k^n$  в конце рассматриваемого такта  $t$ , согласно определению в разделе I и функции (4) может быть определено в следующем виде

$$z_k^n = f_k^n(z_i^A, z_k^{n'}) = (z_i^A + z_k^{n'}) \bmod n_k, \quad (10)$$

где  $z_i^A$  - число поданных в течение такта  $t$  фронтов на входе счетчика,  
 $z_k^{n'}$  - исходное состояние счетчика до начала такта  $t$ ,  
 $n_k$  - период счетчика.

Определим далее функцию  $\gamma(z_k^n)$  - число выходных фронтов, выданных счетчиком в рассматриваемом периоде при условии, что он находится в состоянии  $z_k^n$ . Пусть  $\alpha_i$  - состояние счетчика, при котором выдается  $i$ -ый выходной фронт. Формально принимаем  $\alpha_0 = 0$  и  $\alpha_{m_k+1} = n_k$ , где  $m_k$  - число выходных фронтов, выдаваемых счетчиком за один период. Тогда функция  $\gamma(z_k^n)$  может быть определена следующим образом

$$\gamma(z_k^n) = j \mid \alpha_j \leq z_k^n < \alpha_{j+1}, \quad 0 \leq j \leq m_k. \quad (11)$$

Статический компонент выходного сигнала может быть определен теперь, согласно (6), непосредственно из нового состояния счетчика, достигнутого в конце такта  $t$

$$z_k^c = f_k^c(z_k^n) = z_{k,0}^c \oplus \gamma(z_k^n) \bmod 2, \quad (12)$$

где  $z_{k,0}^c$  - начальное значение статического компонента в начале периода, где  $z_k^n = 0$ .

Динамический компонент выходного сигнала счетчика может быть определен из следующего конструктивного выражения

$$z_k^A = f_k^A(z_i^A, z_k^{n'}) = \left[ \frac{z_i^A + z_k^{n'}}{n_k} \right] m_k + \gamma(z_k^n) - \gamma(z_k^{n'}), \quad (13)$$

где  $[a]$  - целая часть выражения  $a$ .

Здесь первый член определяет число выходных фронтов, выдаваемых счетчиком в течение целого числа периодов, причем для рассматриваемого такта как начальный, так и конечный моменты формально переведены к началам соответствующих

периодов счетчика. Остальные члены суммы (I3) вводят соответствующие поправки для снятия неточностей из-за округления в первом члене.

В ряде случаев при анализе поведения счетных схем требуется решить обратную задачу - определить необходимый динамический компонент входного сигнала для получения заданного динамического компонента выходного сигнала. Соответствующую обратную функцию можно выразить в следующем конструктивном виде

$$\bar{z}_k^A = f_k^{A-1}(\bar{z}_k^A, \bar{z}_k^{n'}) = \left[ \frac{\gamma(\bar{z}_k^{n'}) + \bar{z}_k^A}{m_k} \right] n_k - \bar{z}_k^{n'} + \alpha(\bar{z}_k^{n'}, \bar{z}_k^A). \quad (I4)$$

Здесь

$$\alpha(\bar{z}_k^{n'}, \bar{z}_k^A) = \begin{cases} \alpha_j, & \text{если } 1 \leq j \leq m_k, \\ -(n_k - \alpha_m), & \text{если } j = 0, \end{cases} \quad (I5)$$

где

$$j = [\gamma(\bar{z}_k^{n'}) + \bar{z}_k^A] \bmod m_k. \quad (I6)$$

Первый член суммы (I4) определяет число входных фронтов, необходимых для целого числа периодов рассматриваемого такта, причем как начальные, так и конечные моменты такта формально переведены к началам соответствующих периодов счетчика. Остальные члены суммы (I4) вводят соответствующие поправки для снятия неточностей из-за округления, содержащихся в первом члене. В случае, когда конечный момент такта совпадает с концом периода (случай  $j=0$ ) необходимо исключить входные фронты, поступающие в конце периода счетчика после выдачи последнего выходного фронта.

#### 5. Условия активизации импульсных последовательностей.

Рассмотрим теперь особенности процесса моделирования входного набора на АГ в случае, когда значениями весовых переменных  $z(m)$  при вершинах  $m \in M_k$  на пройденном пути могут быть и недвоичные значения,  $z(m) = X$ . Для исследования данной ситуации может быть применен аппарат булевого дифференцирования.

Рассмотрим функцию  $\bar{z}_k = f_k(Z_k)$  и представляющий ее АГ  $G_k = (M_k, \Gamma_k)$ , где вершины  $m \in M_k$  взвешены переменными  $z(m) \in Z_k$ .

Известно, что значение функции  $f_k$  зависит от переменной  $z(m_i)$ , если выполнено равенство

$$\partial f_k / \partial z(m_i) = 1. \quad (I7)$$

Применительно к модели АГ выполнению условия (I7) соответствует существование на графе при заданном входном наборе следующих непротиворечивых путей:

- путь  $l_D$  из начальной вершины графа через вершину  $m_i$ , где  $z(m_i) = D, D \in \{0, 1\}$ , в некоторую терминальную вершину графа  $m_D$ , где  $z(m_D) = e$ ;

- путь  $l_{\bar{D}}$  из начальной вершины графа через вершину  $m_i$ , при значении  $z(m_i) = \bar{D}$ , в некоторую другую терминальную вершину  $m_{\bar{D}}$ , где  $z(m_{\bar{D}}) = \bar{e}$ .

Обозначим множество пройденных в графе  $G_k$  на путях  $l_D$  и  $l_{\bar{D}}$  вершин через

$$M_k(l) = M_k(l_D) \cup M_k(l_{\bar{D}}) \subseteq M_k.$$

Пусть имеется некоторое подмножество переменных

$$Z_{k,d} = \{z \mid z \in Z_k, z = X\},$$

представляющих сигналы с динамическими компонентами.

Рассмотрим элементы множества  $A^3$  в следующем отношении

$$0 < X < 1.$$

Тогда может быть введена функция [4]

$$\max_z \{ \partial f_k / \partial z(m_i) \}, \quad (I8)$$

где  $z(m_i), z \in Z_{k,d}$ .

Функция (I8) представляет собой обобщение булевой производной для случая многозначных сигналов. Наиболее важным свойством функции (I8) является ее равенство 1, если динамический сигнал  $z(m_i)$  на входе схемы влияет на выходной сигнал  $Z_k$ , превращая его также в динамический, независимо от динамических сигналов на остальных входах  $z \in Z_{k,d} \setminus z(m_i)$ .

Используя функцию максимума булевой производной можно доказать следующее утверждение.

Утверждение I. Для того, чтобы выходной сигнал схемы с функцией  $z_k = f_k(Z_k)$  при заданном входном наборе с динамическими сигналами на входах  $Z_{k,d} \subset Z_k$  не имел динамического компонента, необходимо и достаточно, чтобы выполнялось условие

$$\bigvee_{z_i \in Z_{\kappa, \Delta}} (\max_{z \in Z_{\kappa, \Delta}} \partial f_{\kappa} / \partial z_i) = 0. \quad (19)$$

Использование этого результата позволяет упростить процесс моделирования, так как предварительная проверка выполнения условия (19) позволяет отказаться от вычисления истинных значений динамических компонентов сигналов на входах  $Z_{\kappa, \Delta} \subset Z_{\kappa}$ . Проверка условия (19) может быть легко проведена на модели АГ путем многозначного моделирования [4].

Рассмотрим теперь случай, когда условие (19) не выполняется. Тогда может быть доказано следующее утверждение.

Утверждение 2. Значение динамического компонента выходного сигнала схемы с функцией  $z_{\kappa} = f_{\kappa}(Z_{\kappa})$  при заданном входном наборе с динамическими сигналами на входах  $Z_{\kappa, \Delta} \subset Z_{\kappa}$  равно динамическому компоненту входного сигнала  $z_i \in z_{\kappa, \Delta}$  тогда и только тогда, когда

$$\min_{z \in Z_{\kappa, \Delta}} \partial f_{\kappa} / \partial z_i = 1. \quad (20)$$

Доказательство. Действительно, так как при любой комбинации значений переменных  $z \in Z_{\kappa, \Delta} \setminus z_i$ , будет выполняться зависимость  $z_{\kappa}$  от  $z_i$ , то динамический компонент на входе  $z_i$  полностью передается на выход  $z_{\kappa}$ .

Использование утверждения 2 позволяет определить значение динамического компонента сигнала  $z_{\kappa}$  без вычисления динамических сигналов на входах  $z \in z_{\kappa, \Delta} \setminus z_i$ . В результате сокращается общая трудоемкость при моделировании длинных последовательностей.

В случае, когда условие (20) не выполняется, выходной сигнал  $z_{\kappa}$  зависит в течение рассматриваемого такта  $t$  от нескольких динамических сигналов. Для определения выходного поведения схемы необходимо такт  $t$  разделить на подтакты так, чтобы для каждого подтакта выполнялось либо условие (19), либо (20).

6. Процесс моделирования длинных последовательностей. Процесс моделирования входного набора сводится в общем случае к выделению в текущем такте  $t$  микротактов  $t_p$ ,  $p = 1, 2, \dots$  и вычислению в каждом из них искомого выходного значения  $z_{\kappa}$ .

Обозначим через  $T_0$  момент начала рассматриваемого такта  $t$ ,  $T_j$  - момент конца микротакта  $t_j$ , а  $T(z)$  - момент времени внутри рассматриваемого микротакта  $t_j$ , где сигнал  $z$  изменяет свое значение относительно значения в момент  $T_{j-1}$ . Заметим, что значением момента  $T_j$  внутри рассматриваемого такта будет число входных фронтов в промежутке времени между моментами  $T_0$  и  $T_j$ . Значения  $T(z)$  находятся при помощи вычисления обратных функций вида (14).

Рассмотрим процедуру выделения очередного микротакта  $t_j$ . Из утверждения 2 вытекает следующее следствие.

Следствие I. Значение динамического компонента выходного сигнала схемы с функцией  $z_k = f_k(Z_k)$  при заданном входном наборе равно динамическому компоненту входного сигнала  $z(m_i) \in Z_{k,d}$  в течение некоторого промежутка времени от  $T_{j-1}$  до  $T_j$  тогда и только тогда, когда в этом промежутке

$$\min_{z(m) \in Z_{k,d} \setminus z(m_i)} \partial f_k / \partial z(m_i) = 1. \quad (21)$$

Для использования следствия необходимо найти сначала переменную  $z(m_i)$ , для которой в начале рассматриваемого участка времени условие (21) удовлетворено, а затем определить момент времени  $T_j$ , в котором условие (21) перестает выполняться.

Рассмотрим модель АГ, представляющую данную схему с функцией  $z_k = f_k(Z_k)$ . Пусть  $M_{k,d} = \{m | z(m) = X\} \subset M_k$ . Определим путь  $l_D$  на модели, образуемый статическими сигналами  $z \in Z_k$  в момент  $T_{j-1}$ . Найдём подмножество вершин  $M_{k,d}(l_D) \subset M_k(l_D)$ , такое, что для любой  $m_i \in M_{k,d}(l_D)$  в момент времени  $T_{j-1}$  выполнялось бы условие (17). Найдём вершину  $m_i$ , для которой

$$T(z(m_i)) = \min_{m \in M_{k,d}(l_D)} \{T(z(m))\}. \quad (22)$$

В присутствии нескольких таких вершин берем ближнюю к началу пути  $l_D$  вершину. Предположим, что  $z(m_i) = e$ , где  $e \in \{0, 1\}$ . Тогда, согласно определению булевой производной на модели АГ, должен существовать путь  $l_{\bar{D}}$  на АГ, образуемый статическими сигналами  $z \in Z_k$  в момент  $T_{j-1}$  при условии  $z(m_i) = \bar{e}$ . Найдём подмножество вершин  $M_{k,d}(l_{\bar{D}}) \subset M_k(l_D)$ , такое, что для любой  $m_i \in M_{k,d}(l_{\bar{D}})$  в момент времени  $T_{j-1}$  при



$z(m_i) = \bar{e}$  выполнялось бы условие (17). Тогда может быть доказано следующее утверждение.

Утверждение 3. Для того, чтобы выходной сигнал схемы с функцией  $z_k = f_k(Z_k)$  при заданном входном наборе равнялся динамическому компоненту входного сигнала  $z(m_i) \in Z_{k,d}$  в течение некоторого промежутка времени от  $T_{j-1}^1$  до  $T_j$  достаточно, чтобы выполнялось условие

$$\min_{m \in M_{k,d}(\ell_D) \setminus m_i} \{T(z(m))\} > T(z(m_i)). \quad (23)$$

Доказательство. Действительно, при выполнении условия (23) в момент времени  $T(z(m_i))$  продолжается существование путей  $\ell_D$  и  $\ell_{\bar{D}}$ , следовательно, выполняется и условие (17) зависимости выходного сигнала  $z_k$  от входного сигнала  $z(m_i)$ .

Конечный момент  $T_j$  очередного такта  $t_j$  определяется из выражения

$$T_j = \max \{T(z(m_i)), \min_{m \in M_{k,d}(\ell) \setminus m_i} \{T(z(m))\}\}, \quad (24)$$

где  $M_{k,d}(\ell) = M_{k,d}(\ell_D) \cup M_{k,d}(\ell_{\bar{D}})$ .

В случае, когда условие (23) не выполняется,  $T_j = T(z(m_i))$  и сигнал на выходе будет статическим, равным  $z(m_D)$ , где  $m_D$  — терминальная вершина графа на пути  $\ell_D$ .

Процесс моделирования может быть упрощен за счет отказа от определения подмножеств  $M_{k,d}(\ell_D)$  и  $M_{k,d}(\ell_{\bar{D}})$ . Вместо них в выражениях (22)–(24) могут быть использованы подмножества  $M_k(\ell_D)$  и  $M_k(\ell_{\bar{D}})$ . В результате такого упрощения может увеличиться число микротактов, необходимых при моделировании текущего набора.

На применении утверждения 3 основывается следующий итеративный алгоритм моделирования схемы в очередном такте  $t_j$ .

#### Алгоритм

1°. Определяется путь  $\ell_D$  на АГ с конечной вершиной  $m_D$ , образуемый статическими компонентами входных сигналов в момент  $T_{j-1}$ .

2°. Находится вершина  $m_i$ , для которой выполняется условие (21).

3°. Проверяется выполнение условия (I7) путем построения пути  $\mathcal{L}_D$ . Если условие выполняется, то перейти к п. 5°.

4°. Определяются:  $z_k^c := z(m_D), T_j := T(z(m_j))$ . Конец.

5°. Проверяется выполнение условия (22). Если условие не выполняется, то перейти к п. 4°.

6°. Определяются  $z_k^A := z(m_j)^A, T_j$  из выражения (23). Конец.

Процесс моделирования схемы на заданном входном наборе (в такте  $t$ ) начинается с некоторого выхода  $z_{\text{вых}}$ . Для соответствующего выходного графа выполняется троичное моделирование [4]. Если при моделировании для текущей весовой переменной  $z(m)$  имеет место  $z(m) \in Z_{\text{сч}}$ , принимается  $z(m) = X$ . Для остальных  $z(m) \in Z_{\text{вн}}$  происходит вычисление значения  $z(m)$  на соответствующем графе по аналогичной рекурсивной процедуре.

Моделирование текущего такта  $t$  закончено, если установлено, что  $z_{\text{вых}} \neq X$ . В противном случае выполняется уточненное моделирование по микротактам для вычисления динамического компонента на выходе, согласно приведенному выше алгоритму. В частном случае моделирование такта  $t$  может быть проведено и за один микротакт.

7. Оценки эффективности предложенного метода моделирования. Приведем теперь оценки увеличения эффективности моделирования для случая двоичных счетчиков, сравнивая предложенный подход с логическим моделированием потенциальных сигналов. Определим коэффициент увеличения эффективности в виде

$$K_3 = N_l / N,$$

где  $N_l$  — число моделируемых тактов при логическом моделировании,

$N$  — число моделируемых тактов при предложенном методе моделирования.

В случае последовательных каскадов счетчиков имеем

$$K_3 = T,$$

где  $T$  — длина моделируемой в рассматриваемом такте импульсной пачки.

В случае параллельных каскадов счетчиков, образующих сходящееся разветвление, где

$$M_{1,\min} \leq M_{2,\min} \leq \dots \leq M_{k,\min},$$

$M_{i,\min} (M_{i,\max})$  – представляет собой минимальный (максимальный) промежуток между соседними выходными фронтами в  $i$ -ом параллельном каскаде, измеряемый числом входных фронтов, имеем

$$M_{2,\min} \leq K_3(t) \leq \max\{T, M_{k,\max}\}.$$

Для двоичного счетчика под  $M_i = M_{i,\min} = M_{i,\max}$  понимаем коэффициент пересчета.

В случае параллельных каскадов счетчиков, образующих сходящееся разветвление, которое охвачено глобальной обратной связью, имеем

$$M_{1,\min} \leq K_3(t) \leq \max\{T, M_{k,\max}\}.$$

В двух последних случаях для коэффициента эффективности приведены оценки верхней и нижней границ. Реальное значение  $K_3$  существенно зависит как от самой конфигурации схемы, так и от входного набора на рассматриваемом такте  $t$ , способного активизировать в общем случае различные комбинации параллельных каскадов. Следовательно, значение  $K_3(t)$  может здесь резко изменяться при переходе от набора к набору.

### Л и т е р а т у р а

1. В и й л у п А.А., У б а р Р.Р., Э в а р т с о н Т.А. Управление процессом поиска дефектов в цифровых схемах, содержащих счетные структуры. – В кн.: Тезисы докладов X Всесоюзной школы-семинара по технической диагностике. Таллин, 1984, с. 28–32.
2. Э в а р т с о н Т.А. Моделирование цифровых схем при длинных импульсных последовательностях. – Тр. Таллинск. политехн. ин-та, 1984, № 577, с. 15–26.
3. У б а р Р. Тестовая диагностика цифровых устройств. Таллинский политехнический институт, 1980. II 4 с.
4. В о о л а й н е А.А., П а л л ь М.А., У б а р Р.Р. Обобщенный подход к многозначному моделированию цифровых схем на модели альтернативных графов. – Тр. Таллинск. политехн. ин-та, 1982, № 530, с. 23–37.

Zur Modellierung von langen Testfolgen in  
digitalen Einrichtungen mit Zählern

Zusammenfassung

Es wird ein neuer Zugang zur Modellierung von langen Testfolgen in digitalen Einrichtungen mit Zählern vorgeschlagen, wo logische Methoden zur Analyse von Potenzialen mit der arithmetischen Bearbeitung von Impulsfolgen gemischt wird. Die Formeln zur Berechnung des dynamischen Verhaltens von Zählern und die Bedingungen der Aktivierung von Impulsfolgen in irregulären Teilen einer Schaltung sind gegeben. Ein neuer Algorithmus zur Modellierung und die Bewertungen zur Effektivität des Zuganges sind vorhanden.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ЗАДАНИЯ СТРУКТУРЫ И  
ФУНКЦИЙ МИКРОПРОЦЕССОРНЫХ БИС

Характерной чертой современной вычислительной техники является интенсивное внедрение микропроцессорных БИС. Трудности использования известных структурных методов синтеза тестов для микропроцессорных БИС, обусловленные большой размерностью математических моделей, основанных на вентильном представлении контролируемых схем, приводят к необходимости пересмотра традиционных и разработки новых математических моделей и методов описания дискретных объектов типа микропроцессорных БИС.

В данной статье рассматривается обобщенная модель альтернативных графов (АГ) с целью формализации процесса синтеза тестов для широкого класса дискретных объектов, включая микропроцессорные БИС. В работах [1-4] рассмотрен аппарат альтернативных графов, обеспечивающий более сжатое описание цифровых схем, чем традиционный язык булевых формул, а также предложены методы синтеза тестов, базирующиеся на аппарате АГ. Ниже рассматриваются пути дальнейшего развития модели АГ.

В основе обобщенной модели АГ лежит возможность работы не только на множестве двоичных сигналов, но и на множестве групп сигналов, рассматриваемых как единые целые и разнесенных как "в пространстве", так и "во времени". Такими группами сигналов могут служить, например, параллельные и последовательные коды (данные, адреса, команды, состояния микропрограммных автоматов), импульсные пачки и т.д.

Модель АГ введена для описания функций компонентов рассматриваемого объекта диагностирования (ОД), получаемых в результате структурной или функциональной декомпозиции

объекта. Такими компонентами могут быть, например, производные функционально законченные части объекта (если объект рассматривается на схемном уровне), программно- или микропрограммно-доступные элементы (если объект рассматривается, соответственно, на программном или микропрограммном уровнях) и пр. Вершины АГ взвешены переменными ОД, заданными своими областями определения. Число выходов из каждой вершины равно числу различных значений соответствующей весовой переменной из ее области определения. Пути АГ задают отношения порядка различных сигналов объекта, позволяющие организовать целенаправленный перебор различных функциональных режимов. Каждому выходу графа соответствует определенное значение функции, описываемой графом. Все задачи диагностирования, связанные с синтезом и анализом тестов, приводимы к задачам выбора и прохождения путей на альтернативных графах.

Весовые переменные вершин АГ классифицируются по структуре на простые переменные (потенциальные и импульсные сигналы), слова (параллельные и последовательные коды, последовательности импульсов) и массивы. По типу переменные модели можно классифицировать на функциональные переменные, позволяющие описывать функциональные зависимости рассматриваемого объекта, и специальные переменные (временные и пространственные переменные, переменные "рукопожатия" и вспомогательные переменные).

Временные переменные введены для деления рассматриваемого промежутка времени (команды, машинного цикла, такта) на более детальные части. Временных переменных может быть более одной. Если некоторый путь проходит на графе через несколько вершин с разными временными переменными, то первая пройденная вершина определяет время высшего уровня (такт), следующая - время низшего уровня (микротакт). Вершина, взвешенная временной переменной и имеющая выходов более одного, порождает некоторый процесс, при котором необходимо циклически продолжать движение по разным направлениям из рассматриваемой вершины. Временная переменная может и сама являться функцией от других переменных, описывая микропрограмму с ветвлениями. Независимые временные переменные описывают стандартные циклы работы объекта.

Временные переменные требуется ввести в модель объекта в следующих случаях:

- если значение некоторой входной переменной вызывает временную последовательность значений некоторой выходной переменной, причем в каждый момент реализуется некоторая различная функция;
- если ввод значения некоторой входной переменной требует в некоторые определенные последующие моменты времени ввода значений других входных переменных;
- если некоторая функция объекта не реализуется в виде простого алгебраического выражения и требует представления в виде алгоритма, где протекание процесса зависит от промежуточных результатов (если интерес представляет только функциональное описание объекта, то вводимая шкала времени не должна отражать реальное время, а порядок следования определенных существенных моментов).

Понятие иерархического времени удобно вводить тогда, когда в некоторый момент времени (когда временная шкала уже стандартизирована относительно основных прослеживаемых или вводимых сигналов) в некотором частном случае возникает необходимость еще детальнее следить за выходным поведением объекта или процессом ввода информации.

Введение вершин АГ с временными переменными позволяет:

- часть процессов (непрослеживаемые внутренние процессы объекта) описывать грубее, часть же процессов (сеансы с тестером) с требуемой степенью детальности;
- предъявлять произвольные микропрограммные модели в виде модели альтернативных графов.

Пространственные переменные введены для избежания дублирования повторяющихся подфункций разных разрядов или полей формата вычисляемой на графе переменной. Вершины АГ, взвешенные пространственными переменными, определяют входы в разные подграфы, предназначенные для вычисления разных частей формата рассматриваемого слова. При стремлении к компактному представлению объекта, каждая вершина, взвешенная пространственной переменной, дает эффект

$$E = \frac{KC + C_0}{C + C_0}, \quad (I)$$

где  $C$  - цена подфункции, выведенной из-под области влияния пространственной переменной;

$C_0$  - общая цена остальных подфункций.

В частных случаях имеем

$$\text{и} \quad \begin{aligned} E &= K && \text{при } C \gg C_0 \\ E &= 1 && \text{при } C_0 \gg C. \end{aligned} \quad (2)$$

Критерий выбора вершин АГ с пространственными переменными служит очевидно  $E = \max$ . В общем случае имеем дело со сложной комбинаторной задачей декомпозиции функций объекта.

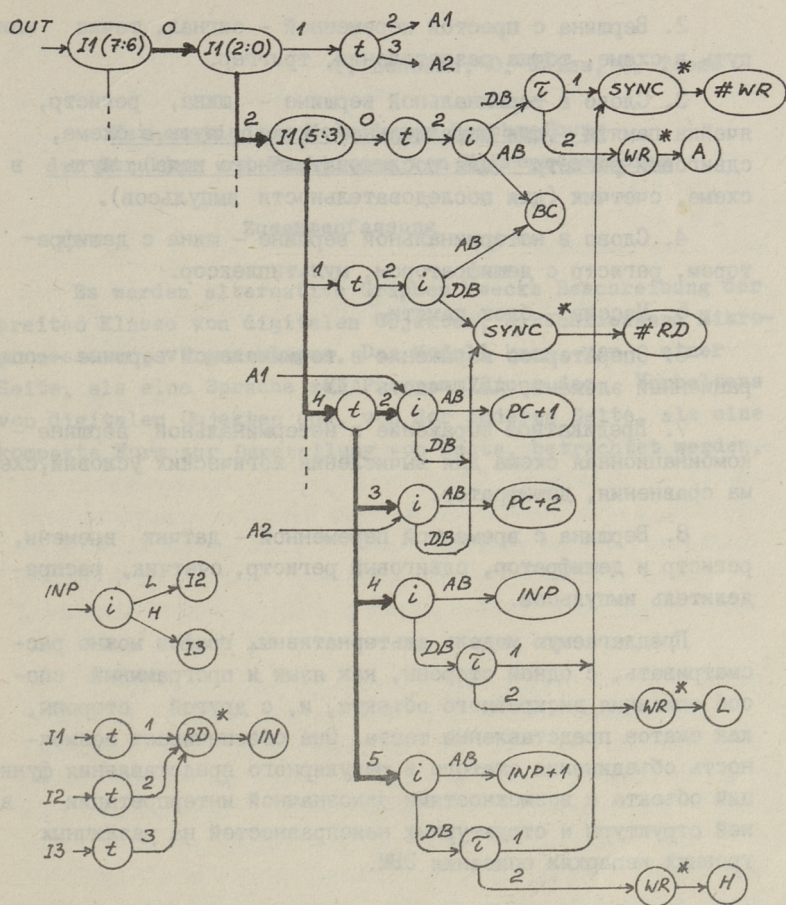
Переменные "рукопожатия" введены с целью синхронизации процессов, происходящих в объекте, с тестовыми воздействиями, генерируемыми в тестере. Они позволяют задавать формальные условия для определения моментов, когда необходимо ввести информацию в объект, или когда необходимо зафиксировать выходное поведение объекта. Если переменные "рукопожатия" представляют собой многоазрядные слова (параллельные или последовательные коды, последовательности импульсов), то из них составляют предикатные выражения.

В качестве примера рассмотрим на фиг. 1 фрагмент модели АГ, описывающей выходное поведение микропроцессора INTEL 8080. Здесь переменные  $I1$ ,  $I2$  и  $I3$  представляют, соответственно, байт кода операции и второй и третий байты команды,  $t$  и  $\tau$  представляют временные переменные, а  $i$  пространственную переменную. Через  $\#WR$  и  $\#RD$  обозначены константы, соответствующие выдаваемым микропроцессором словам состояния. Остальные веса в вершинах представляют какой-нибудь регистр, входное слово или функцию на регистрах. При этом  $I1$ ,  $I2$ ,  $I3$  и  $INP$  можно рассматривать как внутренние переменные, значения которых определяются другими графами. Переменные "рукопожатия" обозначены звездочкой. Жирной линией определено выходное поведение микропроцессора во время выполнения команды SHLD:

$$\begin{aligned} I1 &= 00 \cdot 100 \cdot 010 \\ (I3 \cdot I2) &\leftarrow L \\ ((I3 \cdot I2) + 1) &\leftarrow N. \end{aligned} \quad (3)$$

Заметим, что через вершины  $I1$  проходит один единственный путь, определяемый значением  $I1$ , в то время как временная





Фиг. 1.

переменная  $t$  порождает цикл для всех значений  $t = 2, 3, 4, 5$ , а пространственная переменная  $i$  - цикл для магистрали данных ( $i = DB$ ) и для магистрали адресов ( $i = AB$ ).

Модель АГ может использоваться как для представления функций микропроцессорных ВИС, так и для представления их структуры. Рассмотрим некоторые возможности структурной интерпретации вершин АГ, взвешенных теми или другими типами переменных или выражениями на основе их.

I. Константа в терминальной вершине - ячейка ПЗУ.

2. Вершина с простой переменной - сигнал, точка или путь в схеме, точка разветвления, триггер.
3. Слово в терминальной вершине - шина, регистр, ячейка памяти (для параллельного кода), путь в схеме, сдвиговый регистр (для последовательного кода), путь в схеме, счетчик (для последовательности импульсов).
4. Слово в нетерминальной вершине - шина с дешифратором, регистр с дешифратором, мультиплексор.
5. Массив - блок памяти.
6. Операторное выражение в терминальной вершине - операционный элемент, АЛУ, часть АЛУ.
7. Предикатное выражение в нетерминальной вершине - комбинационная схема для вычисления логических условий, схема сравнения, дешифратор.
8. Вершина с временной переменной - датчик времени, регистр и дешифратор, сдвиговый регистр, счетчик, распределитель импульсов.

Предлагаемую модель альтернативных графов можно рассматривать, с одной стороны, как язык и программный способ описания дискретного объекта, и, с другой стороны, как сжатое представление теста. Она обеспечивает возможность объединения сжатого и регулярного представления функций объекта с возможностями однозначной интерпретации в ней структуры и структурных неисправностей на различных уровнях иерархии описания ЭВМ.

#### Л и т е р а т у р а

1. У б а р Р.Р. Генерирование тестов на модели альтернативных графов. - Тр. Таллинск. политехн. ин-та, 1976, № 409, с. 75-81.

2. А к е г с S.B. Binary decision diagrams. - IEEE Trans. on Comp., June 1978, vol. C-27, p. 509-516.

3. П л а к к М.П., У б а р Р.Р. Построение тестов цифровых схем на модели альтернативных графов. - Автоматика и телемеханика, 1980, № 5, с. 152-163.

4. У б а р Р.Р. Тестовая диагностика цифровых устройств. I и П ч. Таллин, ТИИ, 1981. 226 с.

Mathematisches Modell zur Beschreibung  
der Struktur und Funktionen von Mikroprozessoren

Zusammenfassung

Es werden alternative Graphen zwecks Beschreibung der breiten Klasse von digitalen Objekten, einschliessend Mikroprozessoren, vorgeschlagen. Das Modell kann, von einer Seite, als eine Sprache zur Programmierung des Verhaltens von digitalen Objekten und, von der anderen Seite, als eine kompakte Form zur Darstellung von Tests, betrachtet werden.



М.Г. Шендрик, Б.Г. Тамм

ОБ ОДНОМ ПОДХОДЕ К ФОРМУЛИРОВАНИЮ ЦЕЛЕЙ  
ПРОЕКТИРОВАНИЯ

## Введение

Существующие в настоящее время подходы к формулированию целей проектирования новых машин базируются на использовании гуманистической системы проектировщика и обладают множеством недостатков. К ним относятся высокие квалифицированные требования, значительная трудоемкость, потеря и искажение знаний и опыта формулирования целей проектирования [1].

Устранение этих недостатков для общего случая формулирования целей проектирования и различных приложений в настоящее время затруднительно, но для узко специализированных проблемных областей проектирования удается их избежать посредством использования для формулирования целей проектирования экспертных систем с базами целей [2]. При создании таких систем решаются вопросы общения проектировщика с экспертной системой, представления знаний о целях проектирования и выводы решения о целевом описании проблемы проектирования.

В предлагаемой работе рассматриваются вопросы лингвистического моделирования формулировок целей проектирования, создания моделей целей проектирования, предлагается структура экспертной системы формулирования целей проектирования и рассматриваются режимы ее функционирования.

## I. Формулировка цели проектирования

Цель проектирования имеет лингвистическую форму представления и описывается побудительным предложением естественного языка. Анализ этого предложения позволяет выделить лингвистические выражения, которые разделяются на семан-

тические классы: проектное действие, объект проектирования, предметная область и граница.

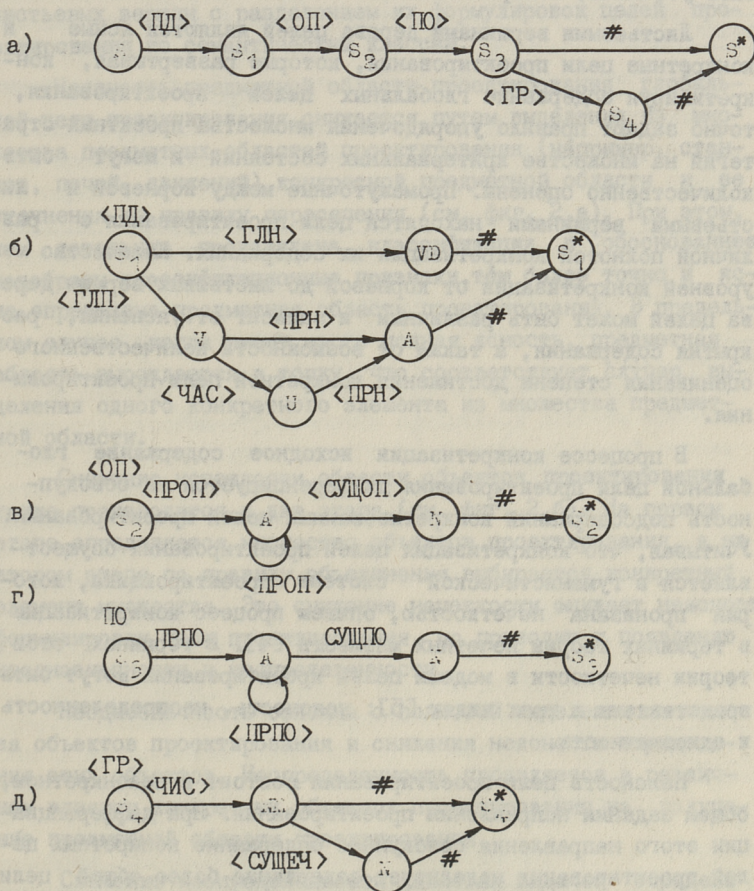
Лингвистическое выражение в каждом из этих классов состоит из набора слов, характеризуемого семантикой названия класса, а совокупная семантика классов образует семантику формулировки цели проектирования в целом. Совокупность семантических классов, составляющих формулировку цели проектирования, представим в виде сети с конечным числом состояний (см. фиг. 1,а). В ней семантический класс представляет переход из одного состояния анализа формулировки цели проектирования в другое. В свою очередь, каждый семантический класс представим в виде сети грамматики лингвистического выражения каждого семантического класса (см. фиг. 1,б-д). В этой сети заложен синтаксис лингвистического выражения семантического класса. Например, в формулировке цели проектирования "Обеспечить наружную температуру электрической методической печи не более пятидесяти градусов" лингвистическое выражение "Обеспечить... не более..." входит в класс проектного действия, "наружную температуру" - в класс объекта проектирования, "электрической методической печи" - в класс предметной области, "пятьдесят градусов" - в класс границы.

Рассмотренное сетевое представление формулировки цели проектирования создает базис для определения языка формулирования целей проектирования. Граматику этого языка опишем расширенной сетью переходов В.А. Вудса в виде тройки [3]  $G = (V, L, N)$ , где  $V$  - описание словарей формулировки цели проектирования,  $L$  - функция выделения семантических классов,  $N$  - описание вершин с выходящими из них дугами - кустов.

Использование этой грамматики для определения языка формулирования целей проектирования обусловлено компактностью представления, а также эффективностью и мощностью грамматической модели языка.

## 2. Модель целей проектирования

Модель целей проектирования состоит из упорядоченного набора целей проектирования и задает совокупность направленных проектирования. Цели проектирования в модели целей иерархически упорядочены, взаимосвязаны и образуют древовидную структуру. Корневой вершиной дерева целей является глобаль-



Фиг. 1. Сети переходов формулировки цели проектирования.

ная цель проектирования, в самом общем виде предъявляющая требования к направлению проектирования. Эта цель сильно обобщена и не поддается непосредственной оценке. Для непосредственного использования в проектировании она должна быть уточнена, конкретизована.

Листьевыми вершинами дерева целей являются ясные и конкретные цели проектирования, которые развертывая, конкретизируя содержание глобальных целей проектирования, точно задают правило упорядочения множества проектных стратегий на множестве критериальных состояний и могут быть количественно оценены. Промежуточные между корневой и листьевыми вершинами находятся цели проектирования с различной полнотой конкретизации их содержания. Количество уровней конкретизации от корневой до листьевых вершин дерева целей может быть различным и зависит от уяснения, раскрытия содержания, а также от возможности количественного оценивания степени достижения конкретной цели проектирования.

В процессе конкретизации исходное содержание глобальной цели проектирования дифференцируется на совокупность подсодержаний конкретизованных целей проектирования. Учитывая, что конкретизация целей проектирования осуществляется в гуманистической системе проектировщика, которая "пронизана" нечеткостью, опишем процесс конкретизации в терминах теории нечетких множеств [4]. В терминах этой теории нечеткости в модели целей проектирования могут быть представлены в трех видах [5]: неясность, неопределенность и однозначность.

Неясность цели проектирования состоит в неконкретном, общем задании направления проектирования. При дифференциации этого направления совокупное содержание конкретных целей проектирования моделирует содержание более общей цели проектирования, и как следствие имеет место ошибка моделирования. Наличие этой ошибки приводит к неоднозначности, то есть к неполному, суженному воспроизведению содержания более общей цели проектирования.

Вклад содержаний более конкретных целей проектирования в содержание более общей цели проектирования существенно различен и априори неизвестен.



Неизвестность этого вклада при конкретизации целей проектирования обусловлена появлением нечеткости, называемой неопределенностью.

Анализ путей снижения нечеткости в модели целей проектирования проведем только для наиболее важных корневой и листовых вершин с разделением их формулировок целей проектирования по семантическим классам.

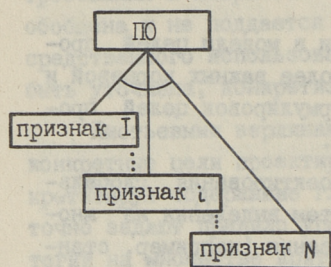
Неясность предметной области проектирования глобальной цели проектирования снижается путем выделения из множества предметных областей проектирования (например, станков, печей, движений) конкретной предметной области и ее уточнения по правилу пересечения (см. фиг. 2,а). При этом, чем детальнее произведена классификация и обоснованнее подобраны классификационные признаки, тем более точно и ясно определена предметная область проектирования. В предельном случае, когда имеет место полная ясность, предметная область вырождается в точку, что соответствует случаю выделения одного конкретного элемента из множества предметной области.

Снижение нечеткости области объектов проектирования также производится в два этапа (см. фиг. 2,б). На первом этапе определяется множество объектов проектирования, а на втором этапе по правилу объединения выбирается конкретный элемент множества. Это снижение нечеткости снижает неясность формулировки цели проектирования, но приводит к появлению неоднозначности и неопределенности.

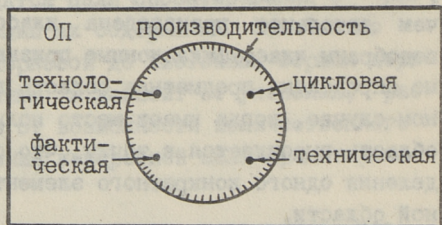
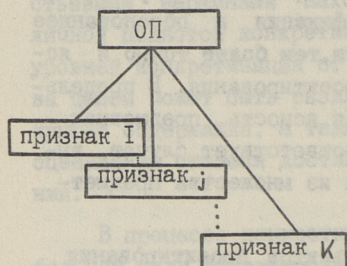
Неоднозначность связана с полнотой выделения множества объектов проектирования и снижения неясности определения этих объектов. Неопределенность проявляется в различном влиянии конкретных объектов проектирования на получение предметной области проектирования.

Снижение неопределенности является задачей определения относительной значимости конкретных целей проектирования. Решение этой задачи в большинстве практических случаев целесообразно производить не на этапе построения модели конкретизированных целей проектирования, а при оценивании степени достижения целей с помощью критериев. По этой причине рассмотрение путей снижения неопределенности в дальнейшем рассмотрении опускается.

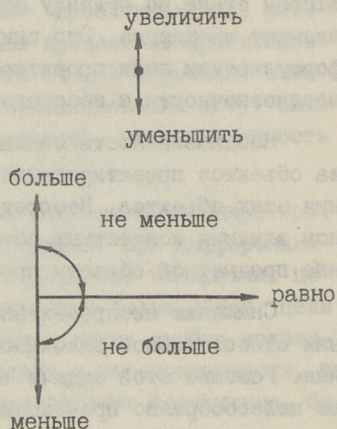
а)



б)



в)



Фиг. 2. И-ИЛИ графы семантических классов и их поясняющие примеры.

Неясность проектного действия снижается в два этапа. На первом этапе определяется ограничено или нет направление проектирования, а на втором этапе – собственно направление проектирования (см. фиг. 2, в). Снижение неясности проектного действия производится путем перехода от множественного задания направления к единственному. Например, переходом от лингвистического выражения "Обеспечить... около...", к лингвистическому выражению "Обеспечить... равно...".

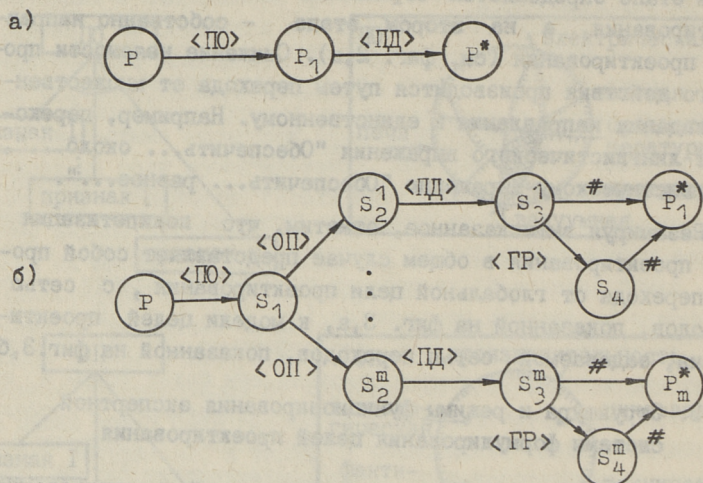
Резюмируя вышесказанное, отметим, что конкретизация целей проектирования в общем случае представляет собой процесс перехода от глобальной цели проектирования, с сетью переходов, показанной на фиг. 3, а, к модели целей проектирования, задаваемой сетью переходов, показанной на фиг. 3, б.

### 3. Структура и режимы функционирования экспертной системы формулирования целей проектирования

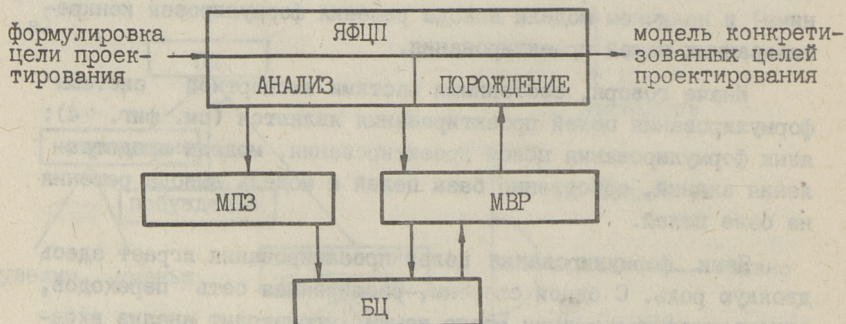
Экспертные системы формулирования целей проектирования отличаются строгим ограничением предметной области проектирования (например, из множества предметных областей используется только предметная область электрических печей сопротивления), использованием предыдущего и текущего опыта проектировщика, наличием базы целей (задание словарей, функций выделения семантических классов и описание отношений между ними) и наличием модели вывода решения формулировок конкретизированных целей проектирования.

Иначе говоря, составными частями экспертной системы формулирования целей проектирования являются (см. фиг. 4): язык формулирования целей проектирования, модель представления знаний, собственно база целей и модель вывода решения на базе целей.

Язык формулирования целей проектирования играет здесь двоякую роль. С одной стороны, расширенная сеть переходов, описывающая грамматику этого языка, производит анализ входных цепочек формулировок целей, производя их синтаксическую и семантическую интерпретацию. С другой стороны, порождает, продуцирует формулировки целей проектирования после вывода решения о модели формулировок целей проектирования на базе целей.



Фиг. 3. Сети переходов глобальной цели проектирования и модели целей проектирования.



Фиг. 4. Общая структура экспертной системы формулирования целей проектирования.

При этом возможны следующие режимы функционирования экспертной системы.

Проектировщик подает на вход системы запрос в виде глобальной цели проектирования (см. фиг. 3,а), оформленный по правилам языка формулирования целей проектирования и на выходе без его участия получает модель конкретизованных целей проектирования. В этом режиме текущие знания проектировщика к работе не привлекаются, а для снижения неясности используется только прошлый опыт, описанный в базе целей.

Проектировщик подает на вход системы глобальную цель проектирования и, привлекая как прошлый опыт из базы целей по объектам проектирования, проектным действиям и границам, так и дополняя эти семантические классы, используя текущий опыт, создает новые структуры в базе целей.

Проектировщик вводит новую предметную область проектирования и, используя прошлый и текущий опыт, вводит новые объекты проектирования, устанавливает связи между существующими и вновь введенными предметными областями и объектами проектирования. Кроме того, проектировщик устанавливает связи объектов проектирования с семантическими классами проектных действий и границ.

В первом режиме функционирования модель представления знаний не задействована, и модель целей проектирования получается посредством поиска в сети переходов, отвечающей предметной обработке глобальной цели проектирования. Этот режим функционирования соответствует традиционному пути использования базы целей [1]. Здесь модель вывода решения представляет собой поисковый алгоритм в сети, описанной И-ИЛИ графами.

Продуцирование формулировок целей проектирования осуществляется посредством использования порождающих возможностей расширенной сети переходов.

Во втором режиме функционирования, который полностью включает в себя первый режим, проектировщик дополняет посредством модели представления знаний необходимые с его точки зрения объекты проектирования в базу целей. При этом он снижает неоднозначность, а также связывает вновь введенные объекты проектирования с семантическими классами про-

ектов действия и границы. Связь устанавливается не только между семантическими классами, но и с элементами словарей. В качестве модели представления знаний здесь используется простой и эффективный аппарат построения сети И-ИЛИ графов.

Третий режим функционирования представляет собой обобщение текущего опыта проектировщика и занесение знаний в базу целей. Но в отличие от второго режима функционирования модель представления знаний используется не только внутри вновь вводимой предметной области, но и для определения связей объектов проектирования вводимой предметной области со смежными предметными областями.

В последнем случае, связи между семантическими классами проектного действия и границы, имеют вид фантомных, то есть уточняемых проектировщиком связей. Наличие фантомных связей приводит к невозможности реализации первого режима функционирования до их уточнения проектировщиком, но существенно упрощает работу проектировщика и повышает гибкость системы формулирования целей проектирования.

Обобщая вышесказанное, отметим, что использование экспертных систем формулирования целей проектирования является эффективным и мощным средством повышения качества и снижения трудоемкости формулирования целей проектирования.

### Заключение

Рассмотренный подход к формулированию целей проектирования был опробован для создания системы формулирования целей проектирования электрических карусельных печей. Эксплуатация этой системы подтвердила ее эффективность и удобство использования.

Таким образом, вышеописанный подход к формулированию целей проектирования может быть использован для различных предметных областей проектирования модели формулировки и модели целей, которые могут быть описаны в соответствии с рассмотренной методологией.

### Л и т е р а т у р а

И. Кузнецов Л.Т. Основы кибернетики. М., Энергия, т. 2, 1979. 584 с.

2. Т а м м Б.Г. Банки данных в САПР. - В сб.: Прикладная информатика. Вып. 2(7). М., Финансы и статистика, 1984, с. 6-16.

3. В у д с В.А. Сетевые грамматики для анализа естественных языков. - В кн.: Кибернетический сборник. Новая серия. Вып. 13. М., Мир, с. 120-158.

4. G a i n e s B.R. Foundations of fuzzy reasoning. - International Journal of Man-Machine Studies, 1976, vol. 8, N 6, p. 623-668.

5. В л а с к М. Vagueness: an exercise in logical analysis. - Phil. Science (USA), 1965, vol. 4, p. 427-455.

M. Shendrik, B. Tamm

On an Approach for Specifying  
Design Goals

Summary

Modeling of syntax and semantics of mechanical design goals is discussed. A specification language for formulating mechanical design goals based on extended grammar of transition network is suggested. The structure of a corresponding expert system and some conditions of its functioning are considered.

## С о д е р ж а н и е

1.	Кеэваллик А., Лаусмаа Т. Декомпозиция конечных автоматов на основе информационного критерия...	3
2.	Судницын А. Декомпозиция микропрограммных автоматов с разделением выходных переменных.....	13
3.	Лейс П., Эллервез П. Оптимизация базиса сети при декомпозиции микропрограммных автоматов....	19
4.	Кеэваллик А., Судницын А., Беркман Б. Система автоматизации декомпозиционного синтеза дискретных управляющих устройств DILOS. ....	27
5.	Аланго В., Китсник П., Коньт Т. Подход к организации данных для логического проектирования цифровых систем.....	35
6.	Круус М. Контрольные эксперименты с микропрограммными автоматами.....	45
7.	Убар Р. Генерирование универсальных тестов для дискретных устройств на альтернативных графах.	51
8.	Убар Р., Эвартсон Т. О моделировании длинных входных последовательностей в дискретных устройствах, содержащих счетные структуры.....	61
9.	Лохуару Т., Ряйза О., Тооме Т. Математическая модель для задания структуры и функций микропроцессорных БИС .....	75
10.	Шендрик М., Тамм Б. Об одном подходе к формулированию целей проектирования.....	83







№ 601

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED

ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

МЕТОДЫ СИНТЕЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ СХЕМ

Электротехника и автоматика XXVI

УДК 62-507

Декомпозиция конечных автоматов на основе информационного критерия. Кеэваллик А., Лаусмаа Т. - Труды Таллинского политехнического института, 1985, № 601, с. 3-11.

Предложен способ усовершенствования парного метода декомпозиции путем введения количественных оценок в виде энтропии для разбиений и пар разбиений. Показано, что на основе энтропийных оценок можно количественно моделировать процесс обработки информации в сетях автоматов и направлять процесс декомпозиционного синтеза.

Библ. наименований - 4.

УДК 62-507

Декомпозиция микропрограммных автоматов с разделением выходных переменных. Судницын А.В. - Труды Таллинского политехнического института, 1985, № 601, с. 13-18.

Предложен метод декомпозиции микропрограммных автоматов с разделением выходных двоичных переменных. Дается необходимое и достаточное условие существования требуемой декомпозиции.

Библ. наименований - 4.

УДК 62-507

Оптимизация базиса сети при декомпозиции микро-  
программных автоматов. Лейс П., Эллервез П. -  
Труды Таллинского политехнического института,  
1985, № 601, с. 19-25.

Разработан метод минимизации компонентных автоматов,  
полученных в результате декомпозиции микропрограммного  
автомата.

Фигур - 3, библиографических наименований - 4.

УДК 681.142.3

Система автоматизации декомпозиционного синтеза  
дискретных управляющих устройств. Кеэваллик А.Э.,  
Судницын А.В., Беркман Б.Е. - Труды Таллинского  
политехнического института, 1985, № 601, с. 27-33.

Приводится общее описание структуры и функционирова-  
ния автоматизированной системы логического синтеза дискрет-  
ных управляющих устройств. Дается характеристика задач де-  
композиционного синтеза, решаемых системой.

Фигур - 1, библиографических наименований - 10.

УДК 681.3

Подход к организации данных для логического  
проектирования цифровых систем. Аланго В.Р.,  
Китсник П.А., Коньт Т.К. - Труды Таллинского  
политехнического института, 1985, № 601, с. 35-43.

В статье рассматриваются вопросы построения автомати-  
зированной системы для логического проектирования цифровых  
систем. Как внутренние структуры данных, так и внешнее опи-  
сание объекта проектирования базируются на его модульном  
представлении. Поддержка обработки данных осуществляется  
интерактивными средствами используемой ЭВМ и инструменталь-  
ной системой управления базой данных. Предложенный подход  
ориентирован на реализацию в среде мини-ЭВМ.

Фигур - 6, библиографических наименований - 4.

УДК 62-507

Контрольные эксперименты с микропрограммными автоматами. Круус М.Э. - Труды Таллинского политехнического института, 1985, № 601, с. 45-50.

В статье рассмотрены проблемы проведения контрольных экспериментов с микропрограммными автоматами. Предложены алгоритмы для построения диагностической и контрольной последовательностей для микропрограммных автоматов, учитывающие их характерные свойства. Показано, что эти последовательности относительно короткие. Приведены результаты экспериментального исследования алгоритмов.

Таблиц - 1, библиографических наименований - 4.

УДК 681.32

Генерирование универсальных тестов для дискретных устройств на альтернативных графах. Убар Р.Р. - Труды Таллинского политехнического института, 1985, № 601, с. 51-60.

Предлагается метод синтеза универсальных тестов, предназначенных для обнаружения произвольных кратных константных неисправностей в комбинационных схемах, построенных на элементах И/ИЛИ с возможными инверторами на входах. Получаемые тесты универсальны в том смысле, что не зависят от реализации схемы. Предлагаемые алгоритмы основываются на представлении булевой функции альтернативным графом и обладают меньшей трудоемкостью, чем известные алгоритмы.

Таблиц - 2, фигур - 2, библиографических наименований - 3.

УДК 681.32

О моделировании длинных входных последовательностей в дискретных устройствах, содержащих счетные структуры. Убар Р.Р., Эвартсон Т.А. - Труды Таллинского политехнического института, 1985, № 601, с. 61-74.

Рассматривается новый подход к моделированию длинных последовательностей в дискретных устройствах, содержащих счетные подсистемы, где логические методы анализа потенциальных сигналов сочетаются с арифметической обработкой числа

фронтов в импульсных пачках. Приводятся выражения для вычисления динамического поведения счетных подсхем и условия активизации импульсных последовательностей в нерегулярных частях схемы. Приводятся алгоритм моделирования и оценки эффективности предлагаемого подхода.

Библ. наименований - 4.

УДК 681.32

Математическая модель для задания структуры и функций микропроцессорных БИС. Лохуару Т., Ряйза О., Тооме Т. - Труды Таллинского политехнического института, 1985, № 601, с. 75-81.

Рассматривается модель альтернативных графов для описания широкого класса дискретных объектов, включая микропроцессорные БИС. Модель можно рассматривать, с одной стороны, как язык и программный способ задания дискретного объекта и, с другой стороны, как сжатое представление теста. Она объединяет возможность сжатого и регулярного представления функций объекта с возможностью однозначной интерпретации в ней структуры и структурных неисправностей на различных уровнях иерархии описания ЭВМ.

Фигур - 1, библ. наименований - 4.

УДК 6.001.63

Об одном подходе к формулированию целей проектирования. Шендрик М.Г., Тамм В.Г. - Труды Таллинского политехнического института, 1985, № 601, с. 83-93.

Рассматриваются задачи моделирования синтаксиса и семантики целей проектирования. Предлагается язык формулирования целей проектирования на основе грамматики расширенной сети переходов.

Проводится анализ модели целей проектирования и предлагается ее описание в виде расширенной сети переходов, где каждый переход представляет семантический класс, представленный в форме И-ИЛИ графа. Предлагается структура эксперт-

ной системы формулирования целей проектирования и рассматриваются различные режимы её функционирования.

Дается оценка реализации рассмотренного подхода на примере экспертной системы формулирования целей проектирования электрических карусельных печей.

Фигур - 4, библиографических наименований - 5.









TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED  
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

СИНТЕЗ И ДИАГНОСТИКА ЦИФРОВЫХ УСТРОЙСТВ И СИСТЕМ  
Электроника и автоматика XIX

УДК 681.32

Проектирование контрольных экспериментов в автоматизированной системе контроля цифровых автоматов "НАКС". Воолайне А.А., Йыги А.Р., Палль М.А. - Труды Таллинского политехнического института, № 530, 1982, с. 3-21.

Статья посвящена вопросам использования системы проектирования контрольного эксперимента для цифровых устройств. Система предназначена для проверки описания, составления модели, составления тест-процедуры и учета полноты тест-процедуры для цифрового автомата. Модель и тест-процедура используются для управления тестером при проведении контрольного эксперимента с объектом контроля. Система работает на ЕС ЭВМ с ОС.

Фигур - 4, библиографических наименований - 6.

Обобщенный подход к многозначному моделированию цифровых схем на модели альтернативных графов. Воолайне А.А., Палль М.А., Убар Р.Р. - Труды Таллинского политехнического института, № 530, 1982, с. 23-38.

Предлагается метод многозначного моделирования цифровых схем на модели альтернативных графов. Метод является универсальным как относительно элементной базы моделируемых объектов, так и относительно алфавита сигналов, выбираемых для моделирования. Для обоснования метода использован аппарат булевого дифференциального исчисления.

Таблиц - 4, фигур - 4, библиографических наименований - 4.

Анализ полноты тестов для цифровых схем на модели альтернативных графов. Китсник П.А. - Труды Таллинского политехнического института, № 530, 1982, с. 39-51.

Рассматривается задача определения множества неисправностей цифровой схемы, обнаруживаемых заданными тестами. Для решения этой задачи применяется модель цифровой схемы в виде системы альтернативных графов. Предлагается аналитический метод, базирующийся на реверсивной обработке тестов и позволяющий расширить класс рассматриваемых логических неисправностей.

Фигур - 4, библиографических наименований - 7.

Анализ методов тестирования микропроцессоров.

Тоомсалу А.К.-Ф. - Труды Таллинского политехнического института, № 530, 1982, с. 53-62.

В статье рассматриваются существующие методы для контроля однокристалльных микропроцессорных схем. Производится анализ методов, указываются положительные и отрицательные стороны и сферы применения каждого метода.

Фигур - 2, библиографических наименований - 13.

Генерирование операндов при синтезе тестов для микропроцессоров. Тоомсалу А.К.-Ф., Убар Р.Р. - Труды Таллинского политехнического института, № 530, 1982, с. 63-73.

Рассмотрен вопрос формализации процесса синтеза тестов для проверки работоспособности микропроцессоров. В качестве математической модели объекта используются векторные альтернативные графы. Выводятся обобщенные функции, описывающие поведение объекта в условиях неисправностей. Путем дифференцирования этих функций находятся условия, которые должны быть выполнены при синтезе тестов.

Таблиц - 3, фигур - 2, библиографических наименований - 2.

Метод алгоритмического генерирования тестов при контроле цифровых схем. Григорьева К.В., Лохуару Т.В., Эвартсон Т.А. - Труды Таллинского политехнического института, № 530, 1982, с. 75-83.

Предлагается метод генерирования тестов в реальном времени при контроле цифровых схем автоматическим тестером. Проверяемый объект задается в виде модели альтернативных графов. Алгоритм генерирования тестов основан на сканировании всех возможных путей на модели альтернативных графов. Рассматриваются два возможных способа доопределения пути на графе до конкретного теста: случайное и детерминированное.

Таблиц - 1, фигур - 4, библиографических наименований - 3.

Синтез коммутаторов на основе ЦИМ для микропроцессоров, реализующих наборы крупных операций.

Божич В.И., Галуев Г.А., Судницын А.В.

Предложен декомпозиционный метод синтеза коммутаторов для микропроцессоров, реализующих наборы крупных операций. Показано, что представление коммутатора в виде логического преобразователя позволяет реализовать коммутатор с высоким быстродействием на программируемых логических матрицах.

Фигур - 4, библиографических наименований - 7.

Метод выбора разбиений для декомпозиции МПА с разделением входных переменных. Беркман Б.Е. -

Труды Таллинского политехнического института, № 530, 1982, с. 93-105.

Развит метод декомпозиции микропрограммных автоматов МПА с ограничением на число входных переменных компонентных автоматов. Предложен алгоритм нахождения исходного множества разбиений на множестве состояний декомпозируемого МПА. Алгоритм реализован в системе программирования ЛЯПАС-М.

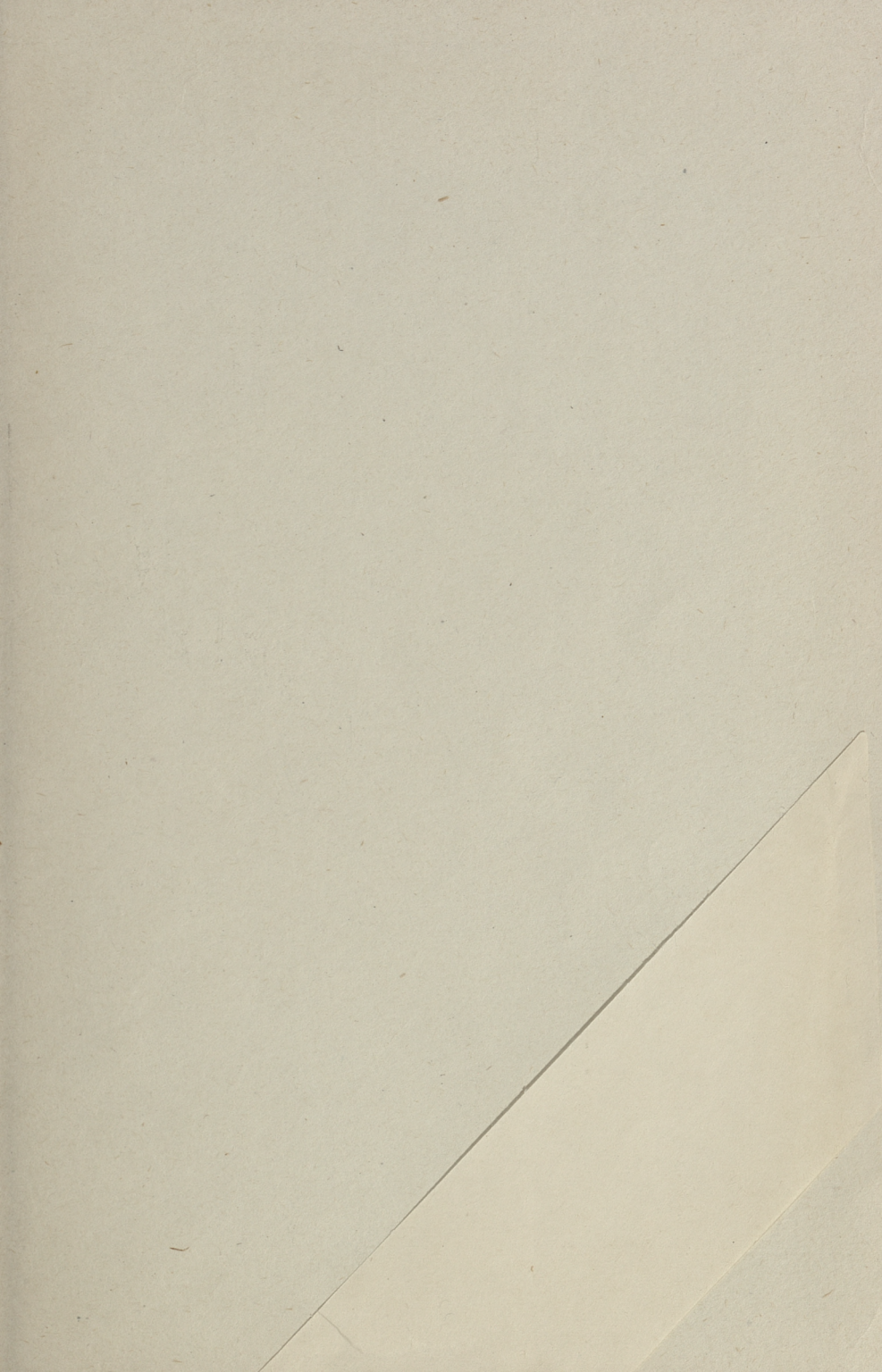
Таблиц - 1, библиографических наименований - 6.

Моделирование технологического процесса кремниевых силовых полупроводниковых приборов на ЭВМ.

Ранг Т.Х., Велмре Э.Э. - Труды Таллинского политехнического института, № 530, 1982, с. 107-112.

В статье рассматривается разработка модели технологического процесса кремниевых силовых полупроводниковых приборов. Приведенная модель с большой точностью описывает технологический процесс, хотя имеет ряд допущений, которые введены с целью ускорения работы программы.

Таблиц - 1, фигур - 1, библиографических наименований - 8.



EESTI AKADEEMILINE RAAMATUKOGU



1 0200 00089867 0

Цена 80 коп.