С.И. Гуров

АЛГОРИТМ ПОЛУЧЕНИЯ РАЗЛОЖЕНИЯ Э.Н. ГИЛЬБЕРТА И ЕГО РЕАЛИЗАЦИЯ ДЛЯ ЗАДАЧИ СИНТЕЗА СХЕМ[1]

Введение

В [1] показано, что любая *полностью определенная булева функция* (ПБФ) f может быть представлена в виде

$$f = M_0 \& \overline{M_1 \& \overline{M_2 \& \ldots \& \overline{M_k}}}\,, \tag{1}$$

где $M_0, M_1, M_2, \ldots, M_k$ – некоторые монотонные булевы функции. Эти функции мы предлагаем называть *функциями Э. Гильберта*, а формулу (1) – *разложением Э. Гильберта* для БФ f. Хотя по тексту указанной статьи можно восстановить алгоритм нахождения функций $M_0, M_1, M_2, \ldots, M_k$, прямое использование его во многих практически важных случаях может вызвать серьезные затруднения. Поясним причину этого.

Указанный алгоритм использует операцию взятия отрицания от функции. На практике же *булевы функции* (БФ) и их системы часто задаются таблицами специального вида (см., например, [2] и ниже, п. 3.), строки которых описывают значения функций на гранях булева n-мерного единичного куба E^n. В этом случае получение таблицы для инверсной функции может оказаться весьма трудоемким. Это связано с тем, что указанные таблицы допускают наличие пересекающихся интервалов E^n, на которых функция может принимать различные значения. При этом для определения значения БФ в области пересечения интервалов необходимо учитывать приоритеты значений функции, с учетом которых построена таблица. При неравных приоритетах значений 0 и 1 инвертирование функции от n переменных и заданной на l интервалах требует, как будет видно, выполнения $O(n \cdot l)$ операций. Указанное обстоятельство, как и опыт работы автора, показывают, что от того, насколько удачно реализована операция инвертирования таблично заданных функций, в значительной степени зависит эффективность работы алгоритма в целом.

[1]Работа выполнена при финансовой поддержке Intel's Strategic CAD Labs (грант Конкурса исследовательских проектов в области автоматизации проектирования интегральных схем 2003 г.) и РФФИ (код проекта 04-01-00161).

Кроме того, на практике обычно приходится иметь дело *с частичными БФ* (ЧБФ). Ясно, что можно получить разложение вида (1) для некоторого доопределения ЧБФ, причем функции Э.Гильберта будут определяться, очевидно, неоднозначно. Поэтому желательно иметь алгоритм, генерирующий монотонные функции $M_0, M_1, M_2, \ldots, M_k$, такие, что

$$F = M_0 \& \overline{M_1 \& \overline{M_2 \& \ldots \& \overline{M_k}}} \tag{2}$$

для ПБФ F, являющейся одним из возможных и, желательно, в некотором смысле простейших, доопределений f. То, что ПБФ F есть (некотрое) доопределение f будем отмечать как $F = Ext(f)$.

Легко заметить, что на основе разложения Э.Гильберта можно предложить простой алгоритм синтеза ПБФ $F = Ext(f)$ в классе схем из функциональных элементов. Действительно, (2) эквивалентно

$$F = M_0 \& (M_1 | (M_2 | (\ldots (M_{k-1} | M_k) \ldots))), \tag{3}$$

где $|$ – символ функции Шеффера (отрицание конъюнкции). Ясно, что в указанном классе формула (3) может быть реализована регулярной каскадной схемой, которая будет выглядеть особенно простой при наличии в базисе элемента функции Шеффера.

Наиболее привлекательным представляется использование алгоритма получения представления (3) для автоматического синтеза цифровых комбинационно-логических схем. Действительно: (а) построение схемы из функциональных элементов является первым этапом синтеза в реальных проектных базисах (т.н. этап абстрактного синтеза); (б) условие наличия в базисе элемента функции Шеффера не является обременительным; (в) среди всех функциональных элементов (исключая инвертор) элемент функции Шеффера имеет простейшую реализацию в виде микроэлектронной электрической транзисторной схемы; и, наконец, (г) функции $M_0, M_1, M_2, \ldots, M_k$ могут быть заданы программируемой логической матрицей (ПЛМ), минимизированной с использованием известных эффективных методов и имеющей дополнительный выигрыш в площади за счет монотонности представляемых функций (т.к. отсутствует необходимость представления их в парафазном виде). Все это позволяет предположить, что использование разложения Э.Гильберта для автоматического синтеза комбинационных блоков цифровых больших интегральных схем (БИС) обещает привести к успеху.

В данной статье приведен и обоснован алгоритм определения функций Э.Гильберта для произвольной ЧБФ f. Для часто используемой на практике системы приоритетов значений таблично заданных функций предложен эффективный способ нахождения их отрицаний. Приведен пример работы алгоритма в составе системы автоматического синтеза комбинационно-логических блоков БИС.

1 Основные определения и обозначения

В статье используются стандартные обозначения величин и основные понятия, ставшие уже общепринятыми в кругу специалистов в области дискретной математики, и поэтому автор позволил себе не оговаривать их специально; все они имеются, например, в [3].

Множество единиц и нулей БФ f обозначается, как обычно, N_f^1 и N_f^0; если $N_f^- = E^n \setminus \{N_f^1 \cup N_f^0\} \neq \varnothing$, то БФ f является частичной. Под взятием отрицания (инвертированием) ЧБФ f будем понимать переход к функции $\varphi = \overline{f}$, у которой $N_\varphi^1 = N_f^1 \cup N_\varphi^0 = N_f^0$. Если $N_\vartheta^1 \subseteq N_\varphi^1$, то функция φ называется *доопределением* ϑ, что обозначается $\varphi = ext(\vartheta)$. Если при этом φ – ПБФ (ЧБФ), то φ – *полное* (*частичное*) *доопределение* ϑ; обозначение для полных доопределений: $\varphi = Ext(\vartheta)$. Множество всевозможных полных доопределений ϑ обозначим $EXT(\vartheta)$. Функции констант (произвольного множества фиктивных переменных) обозначаются $\mathbf{1}$ и $\mathbf{0}$. Обозначения двуместных функций традиционны.

Грань E^n наименьшей размерности, содержащую вершины $\tilde{\omega}$ и $(1, \ldots, 1)$ ($(0, \ldots, 0)$) будем называть *верхним* (*нижним*) *конусом* $\tilde{\omega}$ и обозначать $\tilde{\omega}^\triangle$ ($\tilde{\omega}^\triangledown$). *Верхним (нижним) конусом совокупности вершин* Ω из E^n будем называть объединение верхних (нижних) конусов всех вершин Ω и обозначать его $\Omega^\triangle$ ($\Omega^\triangledown$).

Единичный набор БФ f есть *нижняя единица* f, если его нижний конус не содержит других единиц f. Множество нижних единиц f обозначается $LU(f)$. Нулевой набор БФ f есть *верхний нуль* f, если его верхний конус не содержит других нулей f. Множество верхних нулей f обозначается $UZ(f)$.

Множество монотонных функций известного числа аргументов обозначим M. Ясно, что любое из множеств $LU(f)$ и $UZ(f)$ полностью определяет f, если $f \in M$ (при этом $N_f^1 = LU(f)^{\triangle}$ и $N_f^0 = UZ(f)^{\nabla}$). Заметим, что, например, функция $\mathbf{1}$ может быть задана либо пустым множеством верхних нулей, либо множеством $\{\tilde{0}\}$ нижних единиц, а функция $\mathbf{0}$ – либо пустым множеством нижних единиц, либо множеством $\{\tilde{1}\}$ верхних нулей. Для (возможно частичной) БФ φ ее *мажоранта* f_φ^1 есть монотонная функция, определяемая соотношением $LU(f_\varphi^1) = LU(\varphi)$.

2 Нахождение функций Э. Гильберта

Пусть дана ЧБФ f. Будем искать разложение Э. Гильберта для некоторой функции $F = Ext(f) \in EXT(f)$. Функция F будет у нас построена в процессе нахождения указанного разложения, поэтому мы сразу можем пользоваться формулой (2).

Преобразуем (2):

$$F = M_0\&\overline{M}_1 \vee M_0\&M_2\&\overline{M}_3 \vee M_0\&M_2\&M_4\&\overline{M}_5 \vee \dots$$
$$\dots \vee \begin{cases} M_0\&M_2\&\dots\&\overline{M}_{k-1}\&M_k\,, k \text{ – нечетное,} \\ M_0\&M_2\&\dots\&M_k\,, k \text{ – четное или } 0. \end{cases}$$

Последнее соотношение эквивалентно совокупности равенств

$$\begin{aligned} F = f_0 &= M_0\&(\overline{M}_1 \vee \overline{f_2})\,; \\ f_2 &= M_2\&(\overline{M}_3 \vee \overline{f_4})\,; \\ &\dots\dots\dots\dots\dots\dots \\ f_{k-1} &= M_{k-1}\&\overline{M}_k\,, \quad k \text{ – нечетное,} \\ f_k &= M_k\,, \quad k \text{ – четное или } 0. \end{aligned}$$

(иными словами, $f_{k+1} = 0$ или $f_{k+2} = 1$ при нечетном или, соответственно, четном k).

Отсюда ясно, что функции Э. Гильберта могут быть получены в ходе итерационного процесса, на каждом шаге которого для некоторой ЧБФ $\Phi = ext(\varphi)$, находится разложение

$$\Phi = Me\&(\overline{Mo} \vee \vartheta), \tag{4}$$

где Me и Mo – некоторые подходящие монотонные БФ.

Пусть Me и Mo, удовлетворяющие (4) найдены. Определим функцию $S = Me\&\overline{Mo}$. Тогда в качестве ϑ можно взять функцию, задаваемую множествами своих единиц и нулей следующим образом:

$$N_{\vartheta}^{1} = N_{\varphi}^{1} \setminus N_{S}^{1}, \quad N_{\vartheta}^{0} = N_{\varphi}^{0} \cap N_{Me}^{1}. \tag{5}$$

В данном процессе на первом шаге $\varphi = f$ и далее Me и Mo суть функции Э.Гильберта соответствующего четного и нечетного индексов. Если окажется, что $\vartheta \equiv \mathbf{1}$ или $\vartheta \equiv \mathbf{0}$, то итерации следует прервать, в противном же случае ϑ есть φ для следующего шага. Ясно, что выполнение

$$N_{\vartheta}^{1} \subset N_{\varphi}^{1} \tag{6}$$

обеспечит последовательность строгих вложений множеств единичных наборов разлагаемых функций на каждом шаге и, следовательно, конечность указанного процесса. Ясно также, что в результате описанного процесса будет найдено разложение Э.Гильберта для некоторой функции $F = Ext(f) \in EXT(f)$.

Утверждение. Для ЧБФ φ и $\Phi = ext(\varphi)$ в разложении (4) с учетом выполнения (6) в качестве Me и Mo могут быть взяты мажоранты функций φ и $Me\&\overline{\varphi}$ соответственно.

Доказательство. Нам нужно показать для (4) при $\Phi = ext(\varphi)$ возможность

$$LU(Me) = LU(\varphi)\,; \quad LU(Mo) = LU(Me\&\overline{\varphi})\,, \tag{7}$$

$Me, Mo \in M$ с одновременным выполнением (6).

В тривиальном случае $\varphi = \mathbf{0}$ процесс обрывается не начавшись и в (2) полагаем $k = 0$, $Me = \vartheta = \mathbf{0}$, $Mo = \mathbf{1}$ с обеспечением справедливости (4), (6) и (7). Если φ – монотонная не равная тождественно 0 функция, то, приняв (7) имеем $Me = \varphi$, откуда $Mo = \mathbf{0}$ и, положив $\vartheta = \mathbf{0}$, обеспечиваем выполнение (4) с учетом (6); в этом случае также $k = 0$ в (2).

Далее в рамках доказательства считаем, что φ немонотонна.

Заметим, что в данном случае критерием выполнения (4) является справедливость соотношений $N_{\varphi}^{1} \subseteq N_{Me}^{1}$ и $N_{S}^{1} \cap N_{\varphi}^{0} = \varnothing$. Тогда согласно определению ϑ, (6) будет выполнятся, если N_{S}^{1} содержит хотя бы одну единицу φ, т.е. конечность процесса будет обеспечена при $N_{S}^{1} \cap N_{\varphi}^{1} \neq \varnothing$.

1. Покажем, что $LU(Me) = LU(\varphi)$. Поскольку для справедливости (4) любая единица φ должна быть единицей Me, необходимо выполнение $Me \in EXT(\varphi) \cap M$, откуда вытекает требование $LU(\varphi)^{\triangle} \subseteq LU(Me)^{\triangle}$. Выбор Me из условия $LU(Me) = LU(\varphi)$ обеспечивает выполнения полученного включения. Легко показывается и достаточность данного условия: определив функцию Mo условием $UZ(Mo) = LU(\varphi)$, убеждаемся, что (4) выполняется и $N_S^1 \cap N_\varphi^1 \neq \varnothing$.

2. Покажем возможность $LU(Mo) = LU(Me\&\overline{\varphi})$. Функцию Mo можно определить и более ``выгодным" способом, чем тот, что указан выше. Пусть функция Me определена условием $LU(Me) = LU(\varphi)$. Тогда справедливо $N_\varphi^1 \subseteq N_{Me}^1$. Положим $LU(Mo) = LU(Me\&\overline{\varphi})$. Тогда все нули $Me\&\varphi$ содержатся во множестве единиц Mo, и, следовательно, ни один нуль $Me\&\varphi$ не находится в единицах $\overline{Mo}$. Отсюда множество единиц функции $S = Me\&\overline{Mo}$ не содержит нулей φ и критерий справедливости (4) соблюдается. С другой стороны, $N_S^1 \cap N_\varphi^1 \neq \varnothing$, поскольку $LU(\varphi) = N_S^1$ и описываемый процесс конечен. □

Замечание. Следствием экстремальности (7) в условиях (4) является справедливость соотношений $|N_\varphi^- \cap N_S^1| = \min$ и $|N_\varphi^1 \cap N_S^1| = \max$ при определении Me и Mo соответственно, и отсюда – $|N_\Phi^1 \setminus N_\varphi^1| = \min$.

3 Табличное задание систем частичных булевых функций и особенности выполнения операций над ними

Значения БФ будем задавать на интервалах (гранях) E^n. Интервалы E^n задаются наборами $\overline{\sigma} = \sigma_1\sigma_2\ldots\sigma_n$, где $\sigma_j \in \{1, 0, -\}$, $j = \overline{1,n}$. Как известно, каждая грань $\overline{\sigma}$ булева куба соответствуют некоторой элементарной конъюнкции $K_{\overline{\sigma}}(\tilde{x}) = \bigwedge\limits_{j:j\in J} x_j^{\sigma_j}$ так, что вершина $\tilde{\alpha}$ из E^n принадлежит $\overline{\sigma}$, если и только если $K_{\overline{\sigma}}(\tilde{\alpha}) = 1$. Здесь J – подмножество таких индексов из $\{1, 2, \ldots, n\}$, что $\sigma_j \in \{1, 0\}$. Таким образом переменная x_j присутствует в конъюнкции $K_{\overline{\sigma}}(\tilde{x})$ в прямой (инверсной) форме, если $\sigma_j = 1$, и отсутствует в последней при $\sigma_j = -$. Это в свою очередь означает следующую трактовку символа $-$ в записи интервала $\overline{\sigma}$: интервал $\overline{\sigma}$ может быть заменен двумя интервалами, отличающимися от $\overline{\sigma}$ наличием 1 и 0 в соответствующей позиции.

Для того, чтобы покоодинатная конъюнкция интервалов соответствовала их теоретико-множественному произведению, определим функции $\bigwedge$ и $\&$ над интервалами. Введем на множестве $\{1, 0, -\}$ частичный порядок с помощью рефлексивного несимметричного отношения <<$\geq$>>: $- \geq 1$, $- \geq 0$, 1 и 0 несравнимы. Будем трактовать функции $\bigwedge$ и $\&$ как $\max$ и $\min$ соответственно на множестве $\{1, 0, -\}$ с указанным частичным порядком.

Частичная БФ принимает значения из множества $\{1, 0, -\}$, при этом значение $-$ трактуется как неопределенное. Будем определять ее через множество значения конъюнкций $\{K_{\overline{\sigma}}(\tilde{x})\}$ на некоторой совокупности интервалов $\{\overline{\sigma}\}$. Интервалы $\{\overline{\sigma}\}$ будем называть задающими. Если на всех наборах интервала $\overline{\sigma}$ конъюнкция $K_{\overline{\sigma}}(\tilde{x})$ принимает значение $\omega \in \{1, 0, -\}$, то будем писать $K_{\overline{\sigma}}(\tilde{x}) = \omega$. Значения упорядоченного набора конъюнкций $K^1_{\overline{\sigma}}(\tilde{x})$, $K^2_{\overline{\sigma}}(\tilde{x}), \ldots K^m_{\overline{\sigma}}(\tilde{x})$ участвующих в определении ЧБФ $f_1, f_2, \ldots, f_m$ на некотором интервале $\overline{\sigma}$ будем задавать кортежем $\overline{\omega} = \omega_1\omega_2\ldots\omega_m$, $\omega_j \in \{1, 0, -\}$, $j = \overline{1, m}$. Таким образом в записи $\overline{\omega}$ символ $-$ означает, что он может быть заменен на 1 или 0 при соответствующем доопределении данной функции.

Систему из m частичных булевых функций от n переменных (СЧБФ), заданную на l интервалах будем задавать в табличной форме в виде совокупности строк l

$$N\,\overline{\sigma} : \overline{\omega}, \tag{8}$$

где N – номер строки, $\overline{\sigma}$ – интервал из E^n, $\overline{\omega}$ – кортеж значений функций. Совокупности $\{\overline{\sigma}\}$ и $\{\overline{\omega}\}$ из указанного задания будем называть соответственно левой и правой частью таблицы, а $\overline{\sigma}$ и $\overline{\omega}$ – левой и правой частью соответствующей строки. Заметим, что если $\omega_j \in \{1, 0\}$, $j = \overline{1, m}$, то (5) определяет систему ПБФ заданием соответствующей системы дизъюнктивных нормальных форм. Возможность замены - на 0 или 1 в правой части таблицы (5) есть одна из дополнительных степеней свободы при минимизации ПЛМ, отличающих указанный процесс от задачи минимизации БФ (или систем БФ) в классической постановке.

При указанном задании СЧБФ в случае, если некоторые из задающих интервалов пересекаются, а правые части соответствующих строк не совпадают, возникает неоднозначность в определении значений функций на грани, общей для обоих интервалов. Поэтому таблицы (8) строятся с учетом выбранной системы приоритетов значений составляющих функций.

Точнее, на множестве $\{1, 0, -\}$ с помощью несимметричного рефлексивного отношения, трактуемого как <<приоритет ... больше, чем приоритет ... >> или << ... подавляет ... >>, вводится тот или иной (возможно частичный) порядок. Указанное отношение будем записывать как $>$ (правильнее было бы $\geqslant$, однако принятое обозначение уже стало традиционным и не приводит к затруднениям, поскольку используется лишь при обнаружении коллизии значений функций – см. ниже).

Если для некоторых интервалов $\overline{\sigma}^1$ и $\overline{\sigma}^2$, участвующих в задании данной функции имеет место $\overline{\sigma} = \overline{\sigma}^1 \cap \overline{\sigma}^2 \neq \varnothing$, $K_{\overline{\sigma}^1}(\tilde{x}) = \omega_1$, $K_{\overline{\sigma}^1}(\tilde{x}) = \omega_2$, и $\omega_1 \neq \omega_2$, то говорят, что имеет место коллизия значений функций. При этом считается, что $K_{\overline{\sigma}}(\tilde{x}) = \omega_1$ (ω_2), когда $\omega_1 > \omega_2$ $(\omega_2 > \omega_1)$, и говорят, что принята интерпретация коллизий в соответствии с данным введенным частичным порядком на множестве значений функции. Появление в описанной выше ситуации несравнимых значений ω_1 и ω_2 (противоречие в определении функции) означает ошибку в задании таблицы. Заметим, что появление ошибочных таблиц вполне возможно на практике, и, поэтому, каждая таблица должна быть проверена на отсутствие противоречий, если нет уверенности в обратном. Факт несравнимости значений ω_1 и ω_2 можно записать $\omega_1 = \omega_2$, понимая под этим, что приоритеты соответствующих значений равны.

Системы приоритетов значений удобно задавать пятерками символов вида

$$a\,o\,b\,o\,c,$$

где $a, b, c \in \{1, 0, -\}$, $o \in \{>, =\}$. Нетрудно подсчитать, что всего возможно 10 различных систем приоритетов.

Систему $a = b = c$ (приоритеты всех значений равны) обычно называют инвариантной. В таблицах, построенных по данной системе, строки, соответствующие пересекающимся интервалам должны иметь совпадающие правые части. Название системы связано с тем, что алгоритмы преобразования СЧБФ (например, поглощения и склеивания интервалов и т.д.) и реализации элементарных операций с таблицами, разработанные для других систем приоритетов, будут применимы и для инвариантных систем. По этой причине инвариантная система используется, например, для того, чтобы обеспечить возможность выполнения алгоритмов, предназначенных для работы с таблицами, записанными в различных системах приоритетов и избежать при этом преобразовании таблиц из одной системы в другую (ясно, что такое преобразование есть переборная задача). С другой стороны легко видеть, что таблицы, построенные по инвариантной системе требуют наибольшего числа задающих интервалов, по сравнению с таблицами, использующие другие системы приоритетов.

Три системы вида $a > b = c$ (значение a подавляет значения b и c, приоритеты которых равны) не интересны с точки зрения задания функций таблицами, не используются и, по-видимому, никогда не будут использоваться на практике. Причина заключается в наличии двух значений с наименьшим приоритетом. Наличие лишь одного такого значения (для определенности, c), как во всех других типах систем, исключая инвариантную, дает возможность не включать в таблицу строки вида $\overline{\sigma} : cc \dots c$, что часто приводит к существенному сокращению числа задающих интервалов. Заметим, что принятый порядок на множестве задающих интервалов мог бы быть записан как $- > 1 = 0$.

Из систем приоритетов типа $a = b > c$ и $a > b > c$ практически исключительно используются $1 = 0 > -$ (``листопадовская", примененная, например, в системе синтеза ПЛМ <<Листопад>> [4]) и $1 > - > 0$ (``со слабым нулем"). Последняя система удобна, в частности тем, что соответствующие таблицы, как и все таблицы, построенные по системам типа $a > b > c$, не требуют проверки на наличие противоречий в определении значений. Недостатком системы со слабым нулем является невозможность быстрого получения инверсных значений функций. Действительно, в этом случае некоторые нули той или иной функции будут подавляться ее неопределенными или единичными значениями и не будут представлены в таблице явно, так же как и интервалы $\overline{\sigma}$ с нулевыми значениями всех функций (строки $\overline{\sigma} : 00 \dots 0$).

Алгоритм нахождения разложения Э. Гильберта использует две основных операции: определение мажоранты и взятия отрицания от функции. Его реализация для таблично заданных функций имеет свои особенности: если алгоритм определения мажоранты функции элементарен для любой системы приоритетов значений, то инвертирование функции в системе приоритетов $a > b > c$ может вызвать трудности в силу трудоемкости явного определения соответствующих задающих интервалов и неоднозначности представления результата.

Для получения совокупности нулевых интервалов функции $\varphi = \bar{f}$ необходимо найти множество $E^n \setminus \{N_f^1 \cap N_f^-\}$ в явном виде. Это можно сделать, если имеется возможность выполнять операцию ``вычитания интервалов", т.е. явного определения множества $\overline{\sigma}^1 \setminus \overline{\sigma}^2$. Результат такой операции может быть записан в различной форме. Мы предлагаем записывать его в виде совокупности p интервалов возрастающей размерности $r, r+1, \ldots, r+p-1$, где p – мощность множества $\{j : \sigma_j^1 = -, \sigma_j^1 \& \sigma_j^2 \neq -, j = \overline{1,n}\}$, а r – размерность интервала $\overline{\sigma}^1 \& \overline{\sigma}^2$. Наш опыт показывает, что такое представление обеспечивает как компактность представления с одной стороны, так и необходимое быстродействие алгоритма с другой. Единичные интервалы функции φ получаются как результат вычитания всех строк вида $\overline{\sigma}$: 1 и $\overline{\sigma}$: – таблицы функции f из строки $- \ldots -$: 1.

4 Применение разложения Э. Гильберта к задаче синтеза схем

Описанный алгоритм нахождения функций Э.Гильберта, преобразованной для СЧБФ был запрограммирован и в качестве одного из методов был включен в состав системы LORD автоматического многоуровневого синтеза комбинационно-логических схем цифровых блоков интегральных микросхем в составе. Система LORD создана авторами в НИИ молекулярной электроники МЭП СССР как одна из компонент САПР БИС <<Arc/ws>> [5].

Ниже приведен пример работы алгоритма автоматического синтеза в произвольной логике схемы цифрового блока по его функциональному (поведенческому) описанию. Монотонные функции Э.Гильберта реализованы в виде сумм логических произведений.

Таблица 1 представляет файл kluch.fdt, задающий поведенческое описание работы некоторого разрабатываемого устройства (в данном случае – ключа видеоконтроллера). Работа устройства описывается системой двух частичных булевых функций от восьми переменных (строки kinp и kout), которая задается в табличной форме. Для описания логики работы данного устройства разработчик задал значения соответствующих функций на 36-и интервалах (строка klin) 8-мерного единичного булева куба. Интервалы и значения функций на них указаны 36-ю строками таблицы; каждая строка предваряется порядковым номером. Принятая интерпретация коллизий значений функций на интервалах указана в строке Interpretation. Она указывает на равные приоритеты значений 1, 0 и – (инвариантная кодировка). Это, в частности, означает, что непустое пересечение могут иметь лишь те интервалы левой части таблицы, которые имеют идентичные правые части (например, интервалы 1 и 2). Строка Phase показывает, что все аргументы и все функции заданы в прямой форме, т.е. указаны их значения, а не отрицания.

Таблица 2 есть листинг файла kluch.rm, представляющий табличное задание системы из двух ДНФ, являющихся возможными доопределениями исходных функций из файла kluch.fdt. Интерпретация коллизий значений правых частей таблицы производится в соответствии с приоритетом $1 > - > 0$. Число задающих интервалов сократилось до двух.

Таблица 3 представляет собой листинг файла kluch.sht, описывающего схемное решение логики разрабатываемого ключа контроллера в виде списка цепей (netlist). Строки списка цепей имеют вид

<имя_цепи> = <имя_ключа> (<список_цепей>); .

Имена цепей состоят из буквы и нескольких (не менее одной) цифр, разделенных точками. Входы схемы имеют имена от x.1 до x.8, выходы – y.0 и y.1, все остальные цепи – внутренние. Мнемоника букв имен: nc означает, что данная цепь есть выход элемента NAND, m – данная цепь реализует монотонную функцию (одну из функций Э.Гильберта), z – прочие цепи.

В схеме использовались ключи (функциональные элементы) с именами NAND (``многоместный" штрих Шеффера, т.е. отрицание конъюнкций входов) и NOT (отрицание). Список цепей, заключенный в скобки после имени ключа есть перечисленные через запятую имена цепей, являющихся входами для данного ключа (список имен аргументов функционального элемента).

Мы видим, что для схемной реализации ключа видеоконтроллера, заданного Таблицей 1 размером $(8 + 2) \times 36$ потребовалось всего 18 функциональных элементов. Учитывая реализацию данной схемы в КМОП-технологии, где (в простейшем случае) приходится по 2 транзистора на каждый вход элемента, функция ключа видеоконтроллера может быть реализована с использованием только 40-а транзисторов.

Logical schema type: Boolean
Description type: table_txt

Interpretation: 1=0=-
kinp = 8
kout = 2
klin = 36

1	0	0	-	-	-	1	-	-	:	0	0
2	0	0	-	-	-	-	1	-	:	0	0
3	1	1	-	-	-	1	-	-	:	0	0
4	1	1	-	-	-	-	1	-	:	0	0
5	0	1	1	0	-	0	0	0	:	1	0
6	0	1	1	0	-	0	0	1	:	0	0
7	0	1	1	0	0	0	1	0	:	0	0
8	0	1	1	0	-	0	1	1	:	0	0
9	0	1	1	0	0	1	0	0	:	1	0
10	0	1	1	0	-	1	0	1	:	0	0
11	0	1	1	0	0	1	1	0	:	0	0
12	0	1	1	0	-	1	1	1	:	0	0
13	0	1	1	0	1	0	0	0	:	1	0
14	0	1	1	0	-	0	0	1	:	0	0
15	0	1	1	0	1	0	1	0	:	0	0
16	0	1	1	0	-	0	1	1	:	0	0
17	0	1	1	0	1	1	0	0	:	1	0
18	0	1	1	0	-	1	0	1	:	0	0
19	0	1	1	0	1	1	1	0	:	0	0
20	0	1	1	0	-	1	1	1	:	0	0
21	0	1	0	1	0	0	0	0	:	0	1
22	0	1	0	1	-	0	0	1	:	0	0
23	0	1	0	1	0	0	1	0	:	0	1
24	0	1	0	1	-	0	1	1	:	0	0
25	0	1	0	1	0	1	0	0	:	0	0
26	0	1	0	1	-	1	0	1	:	0	0
27	0	1	0	1	0	1	1	0	:	0	0
28	0	1	0	1	-	1	1	1	:	0	0
29	0	1	0	1	1	0	0	0	:	0	1
30	0	1	0	1	-	0	0	1	:	0	0
31	0	1	0	1	1	0	1	0	:	0	1
32	0	1	0	1	-	0	1	1	:	0	0
33	0	1	0	1	1	1	0	0	:	0	0
34	0	1	0	1	-	1	0	1	:	0	0
35	0	1	0	1	1	1	1	0	:	0	0
36	0	1	0	1	-	1	1	1	:	0	0
Phase :	+	+	+	+	+	+	+	+	:	+	+

Таб. 1.

Logical schema type: Boolean
Description type: table_txt

Interpretation: 1>-> 0
kinp = 8
kout = 2
klin = 2

1	0	1	0	1	-	0	-	0	:	0	1
2	0	1	1	0	-	-	0	0	:	1	0
Phase :	+	+	+	+	+	+	+	+	:	+	+

Таб. 2.

Logical schema type: Boolean
Description type: form_txt

1 nc.0 = NAND (x.1, x.3);
2 nc.1 = NAND (x.1, x.2); 3 nc.2 = NAND (x.0, x.1,x.3, x.6);
4 nc.3 = NAND (x.0, x.1, x.2, x.5);
5 nc.4 = NAND (x.1,x.3, x.7);
6 nc.5 = NAND (x.1, x.3, x.5);
7 nc.6 = NAND (x.1, x.2,x.7);
8 nc.7 = NAND (x.1, x.2, x.6);
9 m.0 = NOT (nc.1);
10 m.1 = NOT (nc.0);
11 m.2 = NAND (nc.3, nc.6, nc.7);
12 m.3 = NAND (nc.2,nc.4, nc.5);
13 z.0.1 = NOT (m.2);
14 z.0.0 = NAND (z.0.1, m.0);
15 y.0 = NOT (z.0.0);
16 z.1.1 = NOT (m.3);
17 z.1.0 = NAND (z.1.1,m.1);
18 y.1 = NOT (z.1.0);

Таб. 3.

Литература

[1] *Gilbert E.N.* Lattice theoretic properties of frontal switching functions // J. Math. Phys. 33, No. 1, 1954, pp. 57-67. (Русск. пер.: *Э.Н.Гильберт.* Теоретико-структурные свойства замыкающих переключательных функций. // Кибернетический сборник, 1, 1960. – С. 175-188).

[2] *Закревский А.Д.* Логический синтез каскадных схем. / М.: Наука, 1981.

[3] *Дискретная* математика и математические вопросы кибернетики. Т. I. / Под ред. С.В. Яблонского и О.Б. Лупанова. – М.: Наука, 1974.

[4] *Бобошко Ю.Г.* Об одном подходе к алгоритмам минимизации не полностью определенных булевых функций и его применение к кодированию программируемых логических матриц. // Микроэлектроника и полупроводн. приборы, 1979, 4. – С. 33-38.

[5] *Авдеев Ю.В., Гаврилов С.В., Гуров С.И. и др.* САПР заказных БИС на открытых вычислительных системах // <<Электронная техника>>. Сер. 3. <<Микроэлектроника>>, 1, 1992. – С. 12-21.