

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 621.382.3.061

Н.А. Авдеев, П.Н. Бибило

**НАХОЖДЕНИЕ ЗАДЕРЖЕК РЕГУЛЯРНЫХ СХЕМ
С ПОСЛЕДОВАТЕЛЬНЫМИ СОЕДИНЕНИЯМИ ТРАНЗИСТОРОВ**

Рассматривается задача определения временной задержки регулярной схемы с последовательными соединениями транзисторов. Предлагается определять задержку с помощью моделирования SPICE-описания схемы, представляющей собой сеть транзисторов. Описывается алгоритм сокращения числа тестовых векторов. Показывается, что для уменьшения задержки схемы можно переставлять входные шины.

Введение

Регулярные структуры с последовательными соединениями транзисторов являются удобным средством реализации управляющей логики заказных сверхбольших интегральных схем (СБИС), выполненных по КМОП-технологии. Такие структуры, предназначенные для реализации дизъюнктивных нормальных форм (ДНФ) булевых функций и являющиеся параметризуемыми по числу входных переменных, реализуемых функций и конъюнкций, на которых заданы ДНФ функций, были рассмотрены в работах [1, 2] и названы «селекторами». Модификация селекторов была предложена в [3] и названа регулярной МОП-схемой. Для этой схемы были предложены методы минимизации площади и средства символьного описания структуры и топологии, однако не была решена задача определения задержки такой структуры. Определение задержек логических схем является важной задачей в современных системах автоматизированного проектирования [4]. Методы определения задержки (временной анализ) нерегулярной схемы из библиотечных логических элементов условно разделяются на два типа – динамические и статические [5]. Динамические методы основываются на моделировании работы схемы. Статические методы рассматривают все возможные пути прохождения сигнала в схеме. Для регулярных структур разрабатываются специализированные методы определения задержек. Например, для программируемых логических матриц (ПЛМ) проблема определения задержек рассматривалась в [6], где была предложена формула вычисления задержки в зависимости от размеров ПЛМ (числа входов, выходов, промежуточных шин) и ряда эмпирических коэффициентов. Другой подход к вычислению задержки ПЛМ с параллельными соединениями транзисторов состоит в моделировании пары полностью нагруженных шин – промежуточной шины, к которой имеется максимально возможное число коммутаций с входными шинами, и выходной шины, имеющей максимальное число коммутаций с промежуточными шинами.

В данной работе предлагаются:

- модификация описанной в [3] регулярной структуры с последовательными соединениями транзисторов;
- алгоритм нахождения минимального числа тестирующих векторов, подаваемых на вход схемы и позволяющих найти ее максимальную задержку;
- алгоритм перестановки входных шин структуры, целью которого является уменьшение задержки структуры.

Таким образом, предлагаемый подход к вычислению задержки регулярной схемы относится к классу динамических методов, однако вместо анализа всех возможных путей распространения сигнала рассматривается сокращенное множество наборов входных сигналов, использование которых при моделировании и позволяет определить максимальную задержку схемы.

1. Синхронизируемая РМОП-схема

Изображенную на рис. 1 схему будем называть синхронизируемой регулярной схемой на базе последовательно соединенных n -МОП ячеек (транзисторов) или РМОП-схемой. Схема реализует ДНФ булевых функций

$$\begin{aligned} y_1 &= \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee x_1x_2\overline{x_3}; \\ y_2 &= \overline{x_1}x_3 \vee x_1\overline{x_3}; \\ y_3 &= x_1x_2x_3 \end{aligned} \quad (1)$$

либо инверсий функций. Входные переменные соответствуют строкам, элементарные конъюнкции – столбцам, каждая из функций реализуется независимо от других функций, т. е. если одна и та же элементарная конъюнкция входит в ДНФ двух (и более) функций, то она реализуется каждый раз в составе той функции, где она требуется.

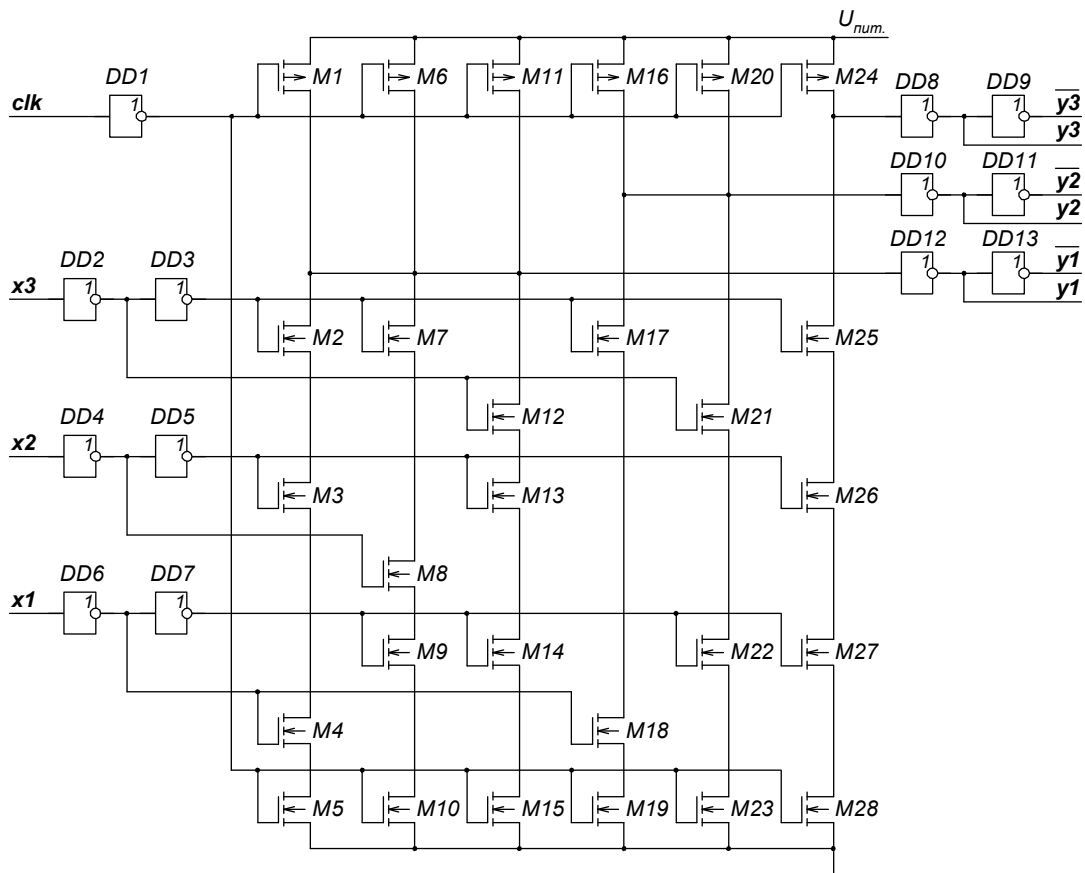


Рис. 1. РМОП-схема, реализующая систему функций (1)

В общем случае РМОП-схема реализует систему

$$\begin{aligned} f^1(x_1, \dots, x_n) &= k_1^1 \vee k_2^1 \vee \dots \vee k_{p_1}^1; \\ f^2(x_1, \dots, x_n) &= k_1^2 \vee k_2^2 \vee \dots \vee k_{p_2}^2; \\ &\dots \\ f^m(x_1, \dots, x_n) &= k_1^m \vee k_2^m \vee \dots \vee k_{p_m}^m \end{aligned} \quad (2)$$

ДНФ m булевых функций, зависящих от n переменных x_1, \dots, x_n . Через k_j^i в (2) обозначена элементарная конъюнкция литералов переменных x_1, \dots, x_n , входящая в ДНФ функции $f^i(x_1, \dots, x_n)$, $i = 1, \dots, m$. Литералом переменной x_q называется x_q либо \bar{x}_q , $q=1, \dots, n$. Каждая из функций f^i может быть реализована в прямой f^i либо инверсной \bar{f}^i форме.

В выходном буфере на выходе каждой из функций находится пара последовательно соединенных инверторов. В отличие от рассмотренной ранее схемы [3, 7] РМОП-схема содержит дополнительный входной синхросигнал clk , по которому осуществляется управление ее работой. Каждый такт (временной интервал) работы схемы разбивается на два временных интервала: первый (длительности τ_1) соответствует предзаряду схемы, во втором (длительности τ_2) вычисляется значение функций для установившихся значений входных переменных. Значения входных сигналов (переменных) должны быть установлены к моменту окончания первого временного интервала (предзаряда).

2. Постановка задачи

Вычисление задержки РМОП-схемы важно для определения длительностей тактов синхросигнала, к которому привязывается функционирование РМОП-схемы. Задержка схемы (параметр τ_2) может быть установлена путем моделирования. Для моделирования требуется построить *тест*, состоящий из множества (последовательности) тестирующих векторов. *Тестирующий вектор* – это булев вектор значений входных переменных, подаваемых на вход Spice-модели (сети транзисторов), для которого можно получить в результате моделирования информацию о времени τ_2 срабатывания схемы. Иногда в понятие тестирующего вектора включается и вектор ожидаемых реакций модели на данное тестирующее воздействие. Задержка РМОП-схемы определяется по максимальной задержке срабатывания схемы для любой комбинации значений входных сигналов. На практике при моделировании важно минимизировать число тестирующих векторов.

Задача 1. Найти минимальное число тестирующих векторов, позволяющих определить задержку РМОП-схемы.

3. Построение тестов для РМОП-схем

Тест для каждой из функций f^i , $i = 1, \dots, m$, строится независимо от других функций, реализуемых на данной структуре.

Рассмотрим построение теста для РМОП-схемы, реализующей отрицание ДНФ функции $f = (a \vee b \vee c \vee d \vee e \vee q)$, которая зависит от шести аргументов x_1, x_2, \dots, x_6 и задана матрицей

$$T = \begin{array}{cccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & \\ \hline 0 & - & 0 & - & 1 & 0 & a \\ - & 1 & 0 & 0 & - & 0 & b \\ 0 & - & - & 0 & 1 & 1 & c \\ 1 & 1 & 0 & 0 & 1 & - & d \\ 0 & - & 0 & 0 & - & 1 & e \\ - & - & 0 & - & - & 0 & q \end{array} .$$

Строки матрицы T (троичные векторы) задают элементарные конъюнкции a, \dots, q . Например, троичный вектор (0-0-10) задает конъюнкцию $a = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 x_5 x_6$, входящую в ДНФ функции f . Каждая строка j матрицы T представляет собой интервал c_j булева пространства V^x . Интервалы c_j взаимно однозначно соответствуют элементарным конъюнкциям k_j функции f .

Интервал c_j задает разбиение r_j булева пространства V^x на два класса $M_{c_j}^0, M_{c_j}^1$. Булево пространство V^x строится над вектором $x = (x_1, \dots, x_n)$ размерности n . Обозначим через $\pi^i, i=1, \dots, s$, непустые классы произведения разбиений r_j .

Видно, что класс π^i представляет собой ДНФ, которую обозначим через D_i . Алгоритм нахождения классов π^i , сводящийся к операциям перемножения ДНФ, умножения одной ДНФ на инверсию другой ДНФ, сравнения ДНФ на равенство, представлен в работе [8]. Применяя данный алгоритм, получим представление классов π^i в виде ДНФ D_i , которые зададим троичными матрицами (табл. 1), символ \neg означает знак логической операции «инверсия».

Таблица 1
Задание классов π^i троичными матрицами

Класс π^i	Троичная матрица (ДНФ)
π^1	$\neg(a \vee b \vee c \vee d \vee e \vee q)$
π^2	000-10 010110
π^3	-10000
π^4	0-1011
π^5	110011
π^6	010010
π^7	110010
π^8	0-0011
π^9	0-0001
π^{10}	000-00 1-01-0 1000-0 010100

Построим матрицу V задания интервалов c_j (элементарных конъюнкций a, b, c, d, e, q) в виде дизъюнкции ДНФ классов π^i :

$$V = \begin{array}{c} \begin{array}{cccccccccc} \pi^1 & \pi^2 & \pi^3 & \pi^4 & \pi^5 & \pi^6 & \pi^7 & \pi^8 & \pi^9 & \pi^{10} \end{array} \\ \left| \begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right| \begin{array}{l} a \\ b \\ c \\ d \\ e \\ q \end{array} \end{array}.$$

Иначе говоря, матрица V задает отношение вхождения классов π^i в интервалы c_j . Каждый интервал c_j равен дизъюнкции ДНФ тех классов π^i , которым соответствуют единицы в

строке, помеченной символом c_j . В свою очередь, каждый класс π^i задает ДНФ, полученную путем пересечения множества R^i тех интервалов, которым соответствуют единицы в столбце, помеченном π^i . Назовем интервалы (элементарные конъюнкции) из множества R^i конъюнкциями, образующими класс π^i . Например, классу π^2 , образованному пересечением интервалов b и q , соответствует множество $R^2 = \{b, q\}$, поэтому будем говорить, что конъюнкции b, q образуют класс π^2 .

Заметим, что для нахождения классов π^i возможен и другой подход, при котором сначала по матрице T строится матрица V без «явного» нахождения классов π^i (это возможно, например, с помощью алгоритма, описанного в работе [9]), а затем с использованием матрицы V находятся классы π^i в виде ДНФ D_i .

Будем говорить, что вектор $\underline{x}^* \in V^{\mathbb{Z}}$ входит в класс π^i , если и только если справедлива импликация $\underline{x}^* \rightarrow D_i$, где D_i – это ДНФ, задающая класс π^i .

Утверждение 1. *Любой тестирующий вектор $\underline{x}^* \in V^{\mathbb{Z}}$ входит в один и только в один из классов $\pi^i, i = 1, \dots, s$.*

Справедливость данного утверждения следует из того, что классы π^i являются непересекающимися и задают разбиение всего булева пространства $V^{\mathbb{Z}}$. Например, для тестирующих векторов (000010), (000110), (010110) из класса π^2 всегда «сработают» только две шины, на которых реализуются элементарные конъюнкции a, q .

Утверждение 2. *Тест для нахождения задержки РМОП-схемы, реализующей функцию f , составляют s булевых векторов – представителей классов π^1, \dots, π^s .*

Представитель класса π^i – булев вектор \underline{x}^* – определяется по его максимальному весу $\delta(\underline{x}^*)$. Вес $\delta(\underline{x}^*)$ булева вектора \underline{x}^* подсчитывается как сумма весов тех строк троичной матрицы T , которые не принадлежат множеству строк R^i , соответствующих классу π^i . В свою очередь, вес строки t , не принадлежащей множеству R^i , подсчитывается как число неортогональных компонент в паре векторов (t, \underline{x}^*) , при этом рассмотрение компонент (и соответственно подсчет) начинается с последней компоненты (справа налево) и продолжается до первой ортогональной пары компонент. Например, для $\underline{x}^* = (000010)$ из класса π^2 веса строк b, c, d, e равны 3, 0, 3, 0 соответственно. Строки a, q не рассматриваются, так как они входят в множество R^2 . Рассмотрим строку b . Ее вес равен трем, так как для пары строк (b, \underline{x}^*) в векторах $\underline{x}^* = (000010)$, $b = (-100-0)$ имеются только три неортогональные компоненты (они выделены жирным шрифтом), если рассматривать их справа налево. Аналогично вес строки c равен 0, так как в векторах $\underline{x}^* = (000010)$, $c = (0--011)$ последние компоненты являются ортогональными. Вес $\delta(\underline{x}^*)$ вектора $\underline{x}^* = (000010)$ равен 6, и этот вес является максимальным для наборов из класса π^2 , поэтому вектор $\underline{x}^* = (000010)$ является представителем класса π^2 в тесте.

Понятие веса для тестирующего вектора \underline{x}_q^* отражает дополнительную задержку РМОП-схемы на «несрабатывающих» шинах: весу соответствует число дополнительных емкостей открытых транзисторов на «несрабатывающих» шинах. Основную задержку РМОП-схемы для

тестирующего набора \underline{x}_q^* вносят «срабатывающие» шины, т. е. шины, на которых реализованы конъюнкции из множества R^i .

В рассматриваемом примере множество всех тестирующих векторов, на которых функция принимает значение 0, и соответствующие задержки (в наносекундах) РМОП-схемы, реализующей функцию f , заданы в первой части табл. 2, т. е. при перестановке $x_1x_2x_3x_4x_5x_6$ входных шин. В табл. 2 через t_{IN} обозначена задержка входного буфера, t_M – задержка матричной подсхемы, t_{OUT} – задержка выходного буфера, τ_2 – суммарная задержка РМОП-схемы:

$$\tau_2 = t_{IN} + t_M + t_{OUT}.$$

Жирным шрифтом в табл. 2 выделены значения τ_2 для тестовых векторов, являющихся представителями классов π^i .

Таблица 2

Результаты моделирования РМОП-схемы

Класс π^i	R^i	Задержка при перестановке $x_1x_2x_3x_4x_5x_6$ входных шин						Задержка при перестановке $x_5x_4x_3x_2x_6x_1$ входных шин					
		Вектор \underline{x}^*	t_{IN}	t_M	t_{OUT}	τ_2	Вес $\delta(\underline{x}^*)$	Вектор \underline{x}^*	t_{IN}	t_M	t_{OUT}	τ_2	Вес $\delta(\underline{x}^*)$
π^2	a, q	000010	0,79	0,98	0,85	2,61	6	100000	0,80	1,00	0,75	2,54	3
	a, q	000110	0,80	0,97	0,72	2,49	2	110000	0,80	1,00	0,74	2,54	3
	a, q	010110	0,80	0,97	0,71	2,48	2	110100	0,80	1,01	0,81	2,62	5
π^3	b, q	010000	0,78	0,94	0,69	2,40	1	000100	0,80	1,01	0,81	2,62	5
	b, q	110000	0,74	0,91	0,68	2,33	1	000101	0,80	0,96	0,76	2,51	4
π^4	c	001011	0,80	2,23	1,29	4,32	4	101010	0,80	2,20	1,21	4,21	3
	c	011011	0,80	2,24	1,29	4,32	4	101110	0,80	2,17	1,23	4,19	3
π^5	d	110011	0,80	2,83	1,63	5,27	6	100111	0,80	2,43	1,07	4,29	0
π^6	a, b, q	010010	0,80	0,81	0,65	2,26	4	100100	0,80	0,81	0,64	2,25	2
π^7	b, d, q	110010	0,80	0,86	0,69	2,35	3	100101	0,80	0,83	0,58	2,21	0
π^8	c, e	000011	0,80	1,23	0,85	2,88	3	100010	0,80	1,22	0,77	2,79	1
	c, e	010011	0,80	1,23	0,88	2,90	4	100110	0,80	1,22	0,77	2,79	1
π^9	e	000001	0,80	2,05	1,04	3,88	1	000010	0,80	2,17	1,28	4,24	4
	e	010001	0,80	2,05	1,05	3,89	1	000110	0,80	2,17	1,28	4,25	4
π^{10}	q	000000	0,80	1,39	1,00	3,19	4	000000	0,80	1,51	1,09	3,40	6
	q	000100	0,80	1,34	0,88	3,02	2	010000	0,80	1,46	1,08	3,34	6
	q	010100	0,79	1,34	0,84	2,97	2	010100	0,80	1,49	1,18	3,48	8
	q	100000	0,80	1,36	0,98	3,15	4	000001	0,80	1,35	0,85	3,00	2
	q	100010	0,80	1,47	1,27	3,54	9	100001	0,73	1,35	0,86	2,94	2
	q	100100	0,80	1,35	0,85	3,00	2	010001	0,80	1,35	0,84	2,99	2
	q	100110	0,80	1,42	1,03	3,25	5	110001	0,80	1,35	0,86	3,01	2
	q	110100	0,80	1,35	0,84	2,99	2	010101	0,80	1,38	1,10	3,28	6
q	110110	0,80	1,42	1,03	3,25	5	110101	0,80	1,38	1,11	3,29	6	

Замечание. Разработана программа, включенная в систему Custom Logic [7] и позволяющая получать Spice-модели РМОП-схем по их структурным описаниям – троичным матрицам. Построенные Spice-модели используются при схемотехническом моделировании, целями которого является определение задержек РМОП-схемы и проверка правильности их функционирования. В табл. 2 указаны задержки, полученные путем схемотехнического моделирования РМОП-схемы в программе ACCUSIM фирмы Mentor Graphics. Для некоторых тестирующих

векторов из одного класса, имеющих одинаковый вес, задержки могут незначительно различаться, что обусловлено погрешностью вычислений, которые производит программа ACCUSIM. Например, в классе π^{10} векторы 000100, 010100 имеют одинаковый вес 2, однако для вектора 000100 задержка составляет 3,02 нс, а для вектора 010100 – 2,97 нс.

Таким образом, задержка РМОП-схемы составляет 5,27 нс и определяется по максимальной задержке, выбираемой среди задержек для всех тестирующих векторов – представителей классов π^i . Для тестирующих векторов из класса π^1 задержка равна τ_1 .

Если же РМОП-схема реализует m функций, то ее задержку будем определять по задержке выходного сигнала той функции f^i , значение которой устанавливается позже установления значений всех других функций. После того как построены тестирующие векторы, легко определяются ожидаемые значения функций на данных векторах. Определение значения функции $f^i(\underline{x}^*)$ для тестирующего вектора \underline{x}^* сводится к проверке импликации $\underline{x}^* \rightarrow f^i(\underline{x})$. Если импликация $\underline{x}^* \rightarrow f^i(\underline{x})$ справедлива (равна 1), то $f^i(\underline{x}^*)=1$, в противном случае $f^i(\underline{x}^*)=0$.

4. Минимизация числа тестирующих векторов

Итак, число тестирующих векторов равно числу классов (табл. 3). Однако число классов может быть значительным, поэтому схемотехническое моделирование Spice-модели РМОП-схемы может быть трудоемким. Поэтому на практике целесообразно минимизировать число тестирующих векторов. Заметим, что некоторые из классов можно исключить из рассмотрения. Для этого их надо научиться сравнивать, принимая во внимание конъюнкции, образующие классы, и веса $\delta(\underline{x}^*)$ представителей классов. Далее представителя класса будем называть *тестирующим вектором класса*.

Таблица 3
Задержки РМОП-схемы для тестирующих векторов

Класс π^i	Тестирующий вектор класса π^i	Задержка τ_2 , нс	Результирующий тест
π^2	010110	2,61	
π^3	010000	2,40	
π^4	001011	4,32	
π^5	110011	5,27	110011
π^6	010010	2,26	
π^7	110010	2,35	
π^8	010011	2,90	
π^9	010001	3,89	
π^{10}	100010	3,54	100010

Длиной $\lambda(k)$ элементарной конъюнкции k назовем число литералов, входящих в данную конъюнкцию. Например, конъюнкции a, b имеют длину $\lambda(a) = \lambda(b) = 4$, конъюнкция q – длину $\lambda(q) = 2$.

Рассмотрим классы π^2 и π^3 . Каждый из этих классов образован двумя конъюнкциями: класс π^2 – конъюнкциями a, q , класс π^3 – конъюнкциями b, q . Задержки, которые вносят «срабатывающие» конъюнкции, будут одинаковы для любых тестовых векторов из данных классов. Однако тестовый вектор 000010 класса π^2 имеет вес 6, а тестовый вектор 010000 класса π^3 – вес 1, поэтому можно сказать, что задержка тестирующего вектора класса π^2 бу-

дет больше (вес соответствует числу дополнительных емкостей транзисторов на «не срабатывающих» конъюнкциях). Можно говорить, что задержка класса π^3 меньше задержки класса π^2 и тестирующий вектор из класса π^3 можно исключить из рассмотрения.

Введем более строго понятие отношения « \leq » между классами. Позже будет показано, что данное отношение является отношением частичного порядка.

Упорядочим конъюнкции k каждого из множеств R^i по убыванию (точнее говоря, по не возрастанию) длин $\lambda(k)$ конъюнкций k . Обозначим через $|R^i|$ мощность множества R^i . Рассмотрим два класса π^i и π^j . Поставим во взаимно однозначное соответствие конъюнкции с одинаковыми номерами, т. е. конъюнкции $k_p^i \in R^i$, имеющей номер p , поставим в соответствие конъюнкцию $k_p^j \in R^j$ с тем же номером p .

Определение. Будем говорить, что классы π^i , π^j находятся в отношении $\pi^i \leq \pi^j$, если и только если выполняются следующие условия:

$$A. \quad |R^i| \geq |R^j|, \quad (3)$$

т. е. мощность (число элементов) множества R^i больше либо равна мощности множества R^j .

$$B. \quad \lambda(k_p^i) \leq \lambda(k_p^j), \quad (4)$$

где $k_p^i \in R^i$, $k_p^j \in R^j$, т. е. длина конъюнкции с номером p из множества R^i меньше либо равна длине конъюнкции с тем же номером p из множества R^j .

$$B. \quad \delta(x_q^{*i}) \leq \delta(x_q^{*j}), \quad (5)$$

т. е. вес $\delta(x_q^{*i})$ тестирующего вектора x_q^{*i} класса π^i меньше либо равен весу $\delta(x_q^{*j})$ тестирующего вектора x_q^{*j} класса π^j .

Если одновременно выполняются отношения $\pi^i \leq \pi^j$ и $\pi^j \leq \pi^i$, то будем считать, что классы являются равными (неразличимыми по задержке). Если же не выполняется ни одно из отношений $\pi^i \leq \pi^j$, $\pi^j \leq \pi^i$, то классы π^i , π^j будем считать несравнимыми.

Легко проверить, что $\pi^3 \leq \pi^2$. Действительно, упорядочим конъюнкции класса π^2 по убыванию длин конъюнкций: 1) a ; 2) q . Упорядочим также конъюнкции класса π^3 по убыванию длин конъюнкций: 1) b ; 2) q . Условия А, Б, В выполняются, поэтому класс π^3 меньше класса π^2 .

Проверим отношение \leq между классами π^6 , π^7 . Порядок конъюнкций в классе π^6 : a, b, q ; порядок конъюнкций в классе π^7 : d, b, q . Условия А, Б выполняются, однако условие В не выполняется, так как вес $\delta(x_q^{*6})$ тестирующего вектора $x_q^{*6}=(010010)$ класса π^6 равен 4, а вес $\delta(x_q^{*7})$ тестирующего вектора $x_q^{*7}=(110010)$ класса π^7 – 3. Отношение $\pi^6 \leq \pi^7$ не выполняется, отношение $\pi^7 \leq \pi^6$ также не выполняется, поэтому классы π^6 , π^7 являются несравнимыми.

Утверждение 3. *Отношение \leq является отношением частичного порядка.*

Зададим отношение \leq ориентированным графом G : вершинам графа будут соответствовать классы π^i , а дуге, исходящей из вершины π^i и заходящей в вершину π^j , будет соответствовать отношение $\pi^i \leq \pi^j$ между классами π^i, π^j .

Для рассматриваемого примера граф G частичного порядка показан на рис. 2.

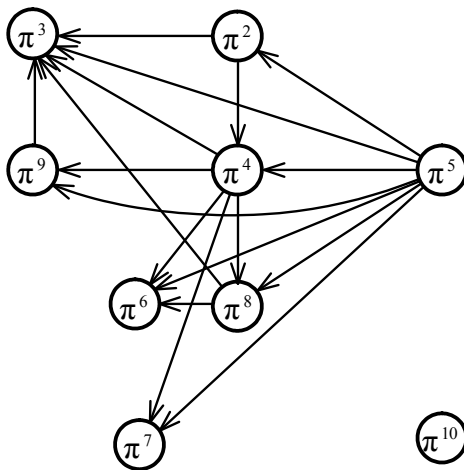


Рис. 2. Граф G

Утверждение 4. В тест, формируемый для оценки задержки РМОП-схемы, могут не включаться тестовые векторы тех классов, которым соответствуют вершины графа G , имеющие хотя бы одну заходящую дугу.

Для графа G только две вершины π^5, π^{10} не имеют заходящих дуг, поэтому только два тестирующих вектора (110011) и (100010) классов π^5, π^{10} включаются в результирующий тест (см. табл. 3).

Решение задачи 1 минимизации числа тестовых векторов для РМОП-схемы (при заданной перестановке входных шин) сводится к построению графа G и нахождению множества вершин, имеющих только исходящие дуги, а тестовые векторы, являющиеся представителями данных классов, будут составлять минимальный по числу векторов тест, позволяющий определить задержку РМОП-схемы.

5. Перестановка входных шин

В примере задержка РМОП-схемы составляет 5,27 нс. Задержку, вносимую срабатывающими шинами, уменьшить нельзя при любой перестановке входных шин, однако перестановка входных шин влияет на число дополнительных емкостей на несрабатывающих шинах, а следовательно, влияет на общую задержку РМОП-схемы. Поэтому целесообразно так переставить входные шины РМОП-схемы, чтобы уменьшить число открывающихся транзисторов для тестирующего вектора, для которого задержка является наибольшей. В рассматриваемом примере таким вектором является тестирующий вектор (110011) из класса π^5 .

Задача 2. Задана троичная матрица T и булев вектор $\underline{x}^* = (x_1^*, \dots, x_n^*)$. Требуется так переставить столбцы матрицы T , чтобы вес $\delta(\underline{x}^*)$ булева вектора \underline{x}^* был минимальным.

Рассмотрим произвольный столбец x_i матрицы T . Пусть i -я компонента тестирующего вектора \underline{x}^* равна α , т. е. $x_i^* = \alpha$, $\alpha \in \{0,1\}$. Если троичный столбец x_i матрицы T содержит только неопределенные элементы « \rightarrow » и определенные элементы $\bar{\alpha}$, то x_i можно переставить на крайнюю правую позицию, при этом данный столбец не увеличит вес тестирующего вектора. Если же в столбце x_i значения всех компонент принадлежат множеству $\{\alpha, -\}$, то данный

столбец следует размещать в крайнюю левую позицию. Назовем такие столбцы *безусловно размещаемыми*.

Будем решать задачу 2 для троичной матрицы T , из которой удалена строка d , и тестирующего вектора 110011 класса π^5 :

$$T = \begin{array}{c|cccccc|l} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & \\ \hline \mathbf{0} & - & 0 & - & 1 & \mathbf{0} & & a \\ - & 1 & 0 & 0 & - & \mathbf{0} & & b \\ \mathbf{0} & - & - & 0 & 1 & 1 & & c \\ \mathbf{0} & - & 0 & 0 & - & 1 & & e \\ - & - & 0 & - & - & \mathbf{0} & & q \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & & \text{Тестирующий} \\ & & & & & & & \text{вектор класса } \pi^5 \end{array} .$$

Заметим, что при подсчете веса тестирующего набора срабатывающая строка (в данном случае строка d) во внимание не принимается. Рассмотрим столбец x_1 , все определенные элементы данного столбца равны 0, а значение $x_1^* = 1$, поэтому столбец x_1 , безусловно, размещается справа. После размещения данного столбца в крайнюю правую позицию значение веса тестирующего вектора не увеличивается. Столбцы x_5 , x_4 , x_3 , x_2 следует поместить слева, так как они не содержат компонент, ортогональных соответствующим компонентам тестирующего вектора. Жирным шрифтом выделены компоненты столбцов, ортогональных значениям компонент тестирующего вектора, курсивом – неортогональных. Целью алгоритма перестановки столбцов является такое их размещение, чтобы в каждой строке хотя бы один выделенный жирным шрифтом элемент оказался правее элемента, выделенного курсивом, точнее говоря, чтобы в каждой строке (начиная справа) первым определенным (0, 1) элементом оказался элемент, выделенный жирным шрифтом. Переставив столбец x_1 в крайнюю правую позицию, легко видеть, что строки a , c , e в дальнейшем можно не рассматривать, так как требуемая ортогональность достигнута уже по переменной x_1 . Матрица T редуцируется путем вычеркивания размещенных столбцов и тех строк, в которых достигнута ортогональность, – в нашем примере это безусловно размещаемые столбцы: x_1 (идет вправо) и x_5 , x_4 , x_3 , x_2 (идут влево). Неразмещенным оказался единственный столбец x_6 , который автоматически попадает на вторую справа позицию и обеспечивает ортогональность в оставшихся строках b , q . Матрица T после перестановки столбцов имеет вид

$$T = \begin{array}{c|cccccc|l} & x_5 & x_4 & x_3 & x_2 & x_6 & x_1 & \\ \hline 1 & - & 0 & - & 0 & \mathbf{0} & & a \\ - & 0 & 0 & 1 & \mathbf{0} & - & & b \\ 1 & 0 & - & - & 1 & \mathbf{0} & & c \\ - & 0 & 0 & - & 1 & \mathbf{0} & & e \\ - & - & 0 & - & \mathbf{0} & - & & q \\ \hline 1 & 0 & 0 & 1 & 1 & 1 & & \text{Тестирующий} \\ & & & & & & & \text{вектор класса } \pi^5 \end{array} .$$

В каждой строке матрицы T справа первым идет выделенный жирным шрифтом определенный элемент, поэтому после перестановки столбцов вес тестирующего вектора (100111) класса π^5 равен нулю. В данном примере полученная перестановка оказалась построенной после первого шага предлагаемого алгоритма перестановки столбцов – алгоритма решения задачи 2.

Алгоритм является жадным и состоит в итеративном выполнении на каждом шаге следующих трех действий:

- 1) размещение вправо и влево безусловно размещаемых столбцов;
- 2) выбор на очередную позицию справа столбца, который обеспечивает ортогональность компонент столбца с компонентой тестирующего вектора в максимальном числе строк, в случае отсутствия безусловно размещаемых столбцов;
- 3) сокращение матрицы путем вычеркивания строк, в которых достигнута ортогональность.

Алгоритм заканчивает свою работу, когда все столбцы будут рассмотрены и назначены на выбранные позиции.

Результаты схемотехнического моделирования РМОП-схемы после перестановки столбцов приведены в правой части табл. 2. Значение максимальной задержки уменьшилось до 4,29 нс, таким образом перестановка столбцов позволила уменьшить задержку. Следует заметить, что у некоторых тестирующих векторов вес изменился. Поэтому после перестановки столбцов следует снова оценить задержку РМОП-схемы, так как предлагаемый алгоритм преследует цель уменьшения задержки схемы только для одного фиксированного тестирующего вектора.

Заключение

Предложенные средства построения моделей РМОП-схем и формирования тестовых воздействий позволяют определять правильность функционирования и временные задержки регулярных схем с последовательными соединениями транзисторов на этапе схемотехнического проектирования, что важно для определения тактовой частоты функционирования проектируемой СБИС.

Список литературы

1. Mead, C. Introduction to VLSI Systems / C. Mead, L. Conway. – Addison–Wesley, 1980. – 396 p.
2. Хейес, Дж. Обобщенная теория переключательных схем и ее применение для проектирования СБИС / Дж. Хейес // ТИИЭР. – 1982. – Т. 70, № 10. – С. 5–18.
3. Бибило, П.Н. Символьная топология матричных макроэлементов СБИС в кремниевом компиляторе SCAS. Ч. 1 / П.Н. Бибило // Микроэлектроника. – 1998. – № 2. – С. 134–143.
4. Бибило, П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П.Н. Бибило. – М.: СОЛОН-Пресс, 2005. – 384 с.
5. Стемповский, А.Л. Методы повышения эффективности временного анализа СБИС / А.Л. Стемповский, С.В. Гаврилов, А.Л. Глебов // Информационные технологии. – 2006. – № 6. – С. 2–12.
6. Papachristou, C.A. A Design Scheme for PLA-Based Control Tables with Reduced Area and Time-Delay Cost / C.A. Papachristou, A.L. Pandya // IEEE Transaction on Computer-Aided Design. – 1990. – Vol. 9, № 5. – P. 453–472.
7. Система «Custom Logic» автоматизированного проектирования управляющей логики заказных цифровых СБИС / П.Н. Бибило [и др.] // Микроэлектроника. – 2004. – Т. 32, № 5. – С. 379–398.
8. Бибило, П.Н. Синтез комбинационных схем методами функциональной декомпозиции // П.Н. Бибило, С.В. Енин. – Минск: Наука и техника, 1987. – 189 с.
9. Гольдберг, Е.И. Алгоритм построения существенных подмножеств множества интервалов / Е.И. Гольдберг // Автоматизация логического проектирования дискретных систем. – Минск: Ин-т техн. кибернетики АН БССР, 1991. – С. 8–18.

Поступила 09.10.07

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: bibilo@newman.bas-net.by*

N.A. Avdeev, P.N. Bibilo

**DELAY DETERMINATION OF REGULAR CIRCUITS
WITH SERIAL CONNECTION OF TRANSISTORS**

A problem of the delay determination of a regular circuit with serial connection of transistors is considered. The delay is suggested to be determined by SPICE simulation, the number of test vectors is minimized. An algorithm for rearranging the input wires of the circuits is proposed that provides decreasing the delay in a number of cases.