

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 519.7

Ю.В. Поттосин, С.Н. Кардаш

ПОВЫШЕНИЕ БЫСТРОДЕЙСТВИЯ КОМБИНАЦИОННОЙ СХЕМЫ ПУТЕМ КОНВЕЙЕРИЗАЦИИ

Рассматривается вопрос повышения быстродействия устройства без памяти, преобразующего последовательность дискретных сигналов. Поставлена задача разбиения заданной многоуровневой комбинационной схемы на заданное число каскадов, на выходах которых должны быть поставлены регистры для обеспечения конвейерной обработки поступающих сигналов. Для решения этой задачи используется модель, основанная на представлении комбинационной схемы в виде ориентированного графа. В процессе решения задачи определяется частота поступления сигналов на вход схемы, и эта частота по возможности должна быть максимальной.

Введение

Повышению производительности систем обработки информации всегда уделялось большое внимание. Одним из способов повышения производительности является применение структуры конвейерного типа, которая имеет еще название «трубопровод» (пер. с англ. pipeline) [1]. Подобную структуру образуют несколько независимых процессоров, соединенных между собой так, что информация на выходе одного процессора является входной информацией для другого процессора. Процессоры образуют информационный конвейер. Выходной процессор выдает результаты через короткие интервалы времени, хотя действительное время прохождения потока информации через конвейер может быть довольно большим.

Принцип конвейеризации эффективно используется, когда характер обработки информации представляется как последовательность операций, каждая из которых состоит из последовательности этапов [2, 3]. Для того чтобы начать выполнение последующей операции, не надо ждать окончания всего процесса выполнения предыдущей операции. Достаточно, чтобы у предыдущей операции был закончен только первый этап. Если конвейер имеет r последовательных этапов, $(i + k)$ -я операция может проходить $(s - k)$ -й этап, где $1 \leq s$, $s - k \leq r$, в то время когда i -я операция проходит s -й этап.

При построении систем цифровой обработки сигналов в режиме реального времени широкое распространение получил систолический принцип организации вычислений, основанный на сочетании идей реализации параллелизма и конвейеризации вычислений с учетом тенденций развития технологии СБИС [4, 5]. Сама технология ориентирована на уменьшение стоимости изготовления одного элемента на кристалле СБИС. Разрабатывается один унифицированный элемент СБИС, и из элементов одного типа строится цепочка, матрица или другая структура. Элементы работают параллельно, выполняя базовую операцию. После ее выполнения осуществляется синхронная передача выходных данных от одного элемента к соседним по всем локальным связям.

Информация в систолическом процессоре распространяется по конвейеру подобно тому, как пульсирует кровь при сокращении систолы сердца. Аналогия между выполнением базовой операции в элементе структуры и сокращением сердца, а также между пульсациями потока исходных данных, промежуточных и окончательных результатов и кровотоков определила название данного класса процессоров как систолических.

В данной работе предпринята попытка найти способ повышения быстродействия путем конвейеризации многоуровневой комбинационной схемы, построенной на основе СБИС.

1. Постановка задачи

В многоуровневой схеме устройства задержка складывается из задержек элементов самой длинной цепочки. Пусть на вход комбинационной схемы поступает последовательность p наборов

двоичных сигналов. Если T – время задержки схемы, то период смены сигналов не может быть меньше T . Время реакции устройства на данную последовательность в этом случае будет не меньше pT . Разобьем схему на k каскадов (C_1, C_2, \dots, C_k), и если τ_C – время задержки самого медленно действующего каскада, то $T \leq k\tau_C$. На выходы каждого каскада поставим элементы задержки (триггеры D), пропускающие сигналы с выходов каскада по сигналу синхронизации. Этот же сигнал синхронизации определяет период смены сигналов на входе устройства, который должен быть не меньше суммы двух задержек: задержки τ_C и задержки τ_D элемента D ($\tau_{\text{clock}} \geq \tau_C + \tau_D$). Теперь время реакции устройства на упомянутую последовательность длины p равно $(k + p)\tau_{\text{clock}}$.

Пример разделения схемы на три каскада показан на рис. 1, где границы каскадов обозначены штриховыми линиями. В схеме использованы модули, содержащиеся в библиотеке КМОП-элементов проектирования заказных СБИС [6].

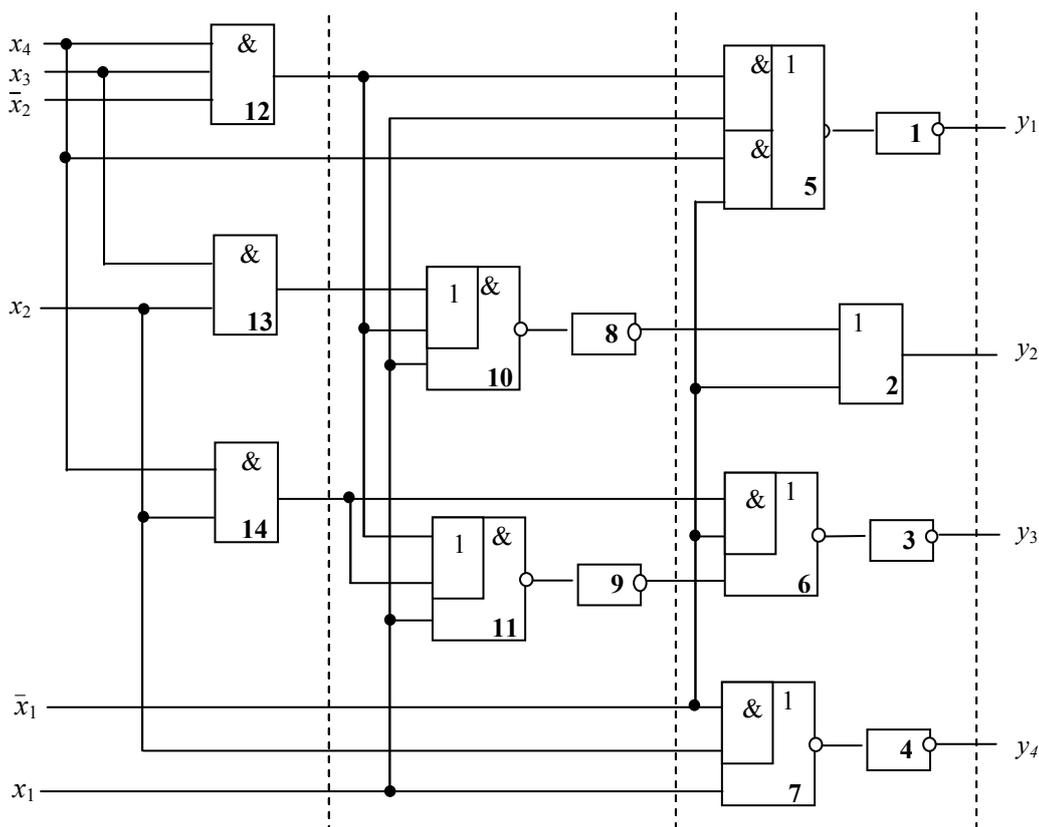


Рис. 1. Схема с выделенными каскадами

Нижняя граница длины p последовательности наборов двоичных сигналов на входе устройства, при которой имеет место ускорение обработки сигналов, определяется неравенством

$$(k + p)\tau_{\text{clock}} < pT, \tag{1}$$

где T – время задержки исходной схемы. Учитывая установленную выше нижнюю границу периода следования сигналов синхронизации $\tau_{\text{clock}} \geq \tau_C + \tau_D$, получим

$$p > \frac{k\tau_{\text{clock}}}{T - \tau_{\text{clock}}} \geq \frac{k(\tau_C + \tau_D)}{T - \tau_C - \tau_D}. \tag{2}$$

Заданную комбинационную схему требуется разбить на требуемое число k каскадов, чтобы обеспечить максимальное быстродействие при описанном конвейерном режиме.

2. Модель комбинационной схемы

Для представления комбинационной схемы удобно использовать в качестве модели ориентированный граф (орграф) $G = (V, A)$ с множеством вершин V и множеством дуг A (рис. 2). Вершины орграфа представляют логические элементы и входные полюсы схемы (индексы у символов вершин совпадают с номерами соответствующих элементов схемы на рис. 1), а дуги показывают направления сигналов от выходов одних элементов к входам других.

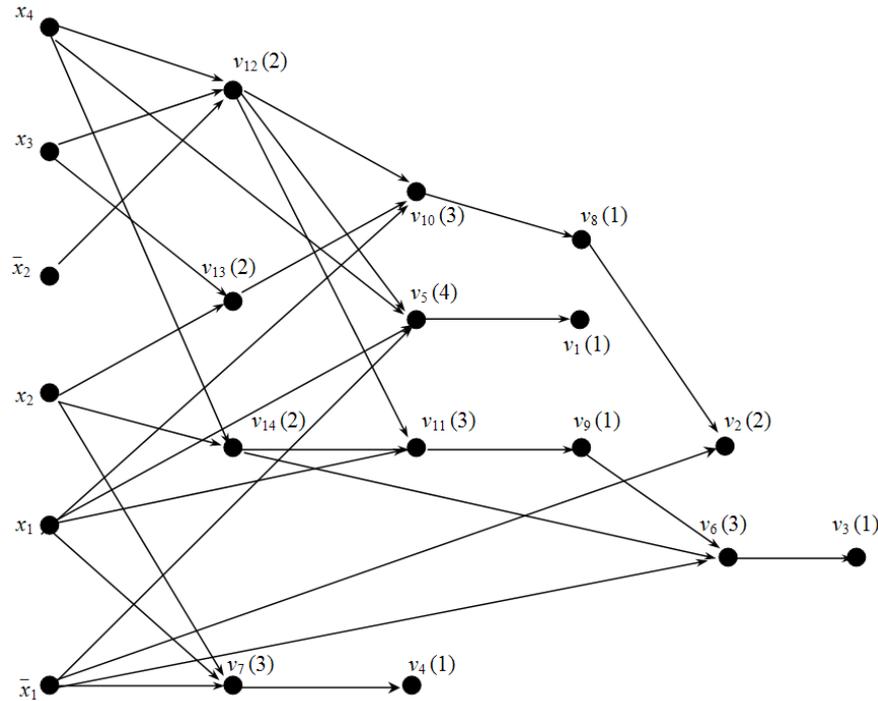


Рис. 2. Орграф G

Орграф G не содержит контуров. Каждой его вершине $v \in V$ приписан вес $\tau(v)$, представляющий задержку соответствующего элемента. Весами могут быть целые числа, пропорциональные временам задержки элементов. Вершины, соответствующие входам схемы, имеют вес, равный нулю. Пусть вершины орграфа на рис. 2 указаны в скобках. При этом считаем, что чем сложнее булева функция, реализуемая логическим элементом, тем большую задержку имеет данный элемент.

3. Разбиение схемы на слои

Сформируем последовательность слоев L_1, L_2, \dots, L_m , представляющую собой упорядоченное разбиение множества вершин V орграфа G с таким свойством, что если вершина v принадлежит полуокрестности исхода $N^+(u)$ вершины u , то эти вершины находятся в разных слоях и слой, содержащий вершину u , предшествует в этой последовательности слою с вершиной v (не обязательно непосредственно). Если длины путей от входов схемы к ее выходам различны, то данное разбиение не является единственным. Следует выбрать такой вариант разбиения на слои, чтобы сумма весов всех слоев была по возможности минимальной. Под весом слоя понимаем максимум весов вершин, принадлежащих этому слою.

Можно выделить два типа вершин орграфа G . К одному типу отнесем вершины, которые лежат на самых длинных путях в орграфе G . Они строго распределяются по слоям и не могут менять свое положение. Назовем их *неподвижными*. Положение в слоях других вершин, которые назовем *подвижными*, можно менять в определенных пределах, скажем, от слоя L_l до слоя L_r ($l < r$). Эти пределы устанавливаются довольно легко. Достаточно выполнить приведенный

ниже алгоритм 1 для заданного орграфа G и орграфа G^c , полученного из G изменением направлений всех дуг. В алгоритме 1 приняты следующие обозначения: $N^-(v)$ – полуокрестность захода, $N^+(v)$ – полуокрестность исхода вершины v , L_i – i -й слой, m – число слоев.

Алгоритм 1

- 1) $L_1 := \{v / N^-(v) = \emptyset\}$, $i := 1$;
- 2) $i := i + 1$, $L_i := \bigcup_{v \in L_{i-1}} N^+(v)$. Если $L_i \neq \emptyset$, перейти к 2,
иначе $j := i$, $m := i := i - 1$;
- 3) $i := j := j - 1$. Если $j = 1$, перейти к 5,
иначе
- 4) $i := i - 1$. Если $i = 1$, перейти к 3,
иначе $L_i := L_i \setminus L_j$, перейти к 4;
- 5) конец.

После выполнения алгоритма 1 над орграфом G^c необходимо порядок в полученной последовательности слоев поменять на обратный. Неподвижные вершины окажутся в одних и тех же слоях. Для подвижной вершины «левым» слоем L_l является тот, который получен при выполнении алгоритма 1 над орграфом G , «правым» слоем L_r – тот, который получен при выполнении алгоритма 1 над орграфом G^c .

На рис. 2 и 3 показаны распределения вершин по слоям в результате применения алгоритма 1 к орграфу G (см. рис. 2) и к орграфу G^c (рис. 3). На обоих рисунках слои представлены вертикальными рядами вершин. Для орграфа G получены слои $L_1 = \{x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, x_4\}$, $L_2 = \{v_7, v_{12}, v_{13}, v_{14}\}$, $L_3 = \{v_4, v_5, v_{10}, v_{11}\}$, $L_4 = \{v_1, v_8, v_9\}$, $L_5 = \{v_2, v_6\}$, $L_6 = \{v_3\}$, а для орграфа G^c после изменения порядка слоев – $L_1 = \{x_2, \bar{x}_2, x_3, x_4\}$, $L_2 = \{x_1, v_{12}, v_{14}\}$, $L_3 = \{v_{11}, v_{13}\}$, $L_4 = \{\bar{x}_1, v_9, v_{10}\}$, $L_5 = \{v_5, v_6, v_7, v_8\}$, $L_6 = \{v_1, v_2, v_3, v_4\}$. Неподвижными являются вершины $\bar{x}_2, x_2, x_3, x_4, v_3, v_6, v_9, v_{11}, v_{12}$ и v_{14} . С неподвижными вершинами образовалось шесть слоев $L_1 = \{x_2, \bar{x}_2, x_3, x_4\}$, $L_2 = \{v_{12}, v_{14}\}$, $L_3 = \{v_{11}\}$, $L_4 = \{v_9\}$, $L_5 = \{v_6\}$ и $L_6 = \{v_3\}$ соответственно с весами 0, 2, 3, 1, 3 и 1.

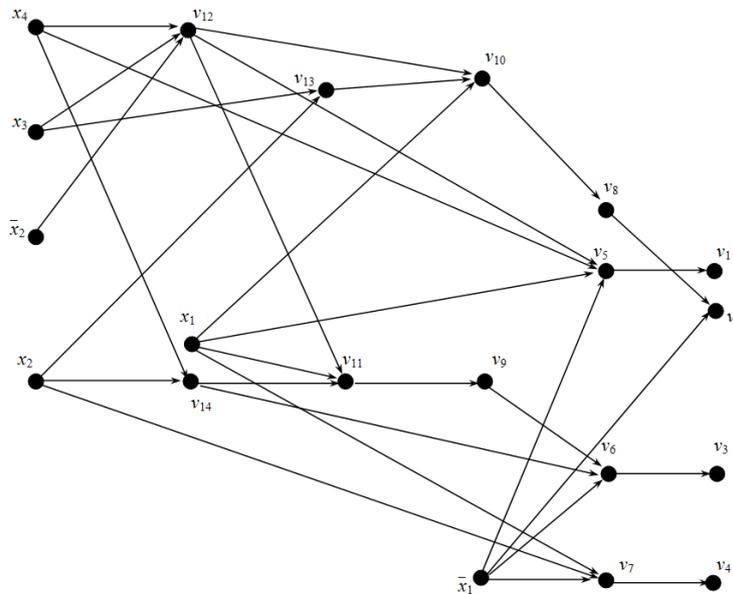


Рис. 3. Распределение по слоям вершин орграфа G , полученное применением алгоритма 1 к орграфу G^c

Остальные вершины являются подвижными, и для каждой из них, как показано выше, существует непустое множество слоев, где вершина может находиться. Вершина x_1 может находиться в слоях L_1 и L_2 , вершина \bar{x}_1 – в слоях L_1, L_2, L_3 и L_4 , вершина v_1 – в сло-

ях L_4 , L_5 и L_6 , вершина v_2 – в слоях L_5 и L_6 , вершина v_4 – в слоях L_3 , L_4 , L_5 и L_6 , вершина v_5 – в слоях L_3 , L_4 и L_5 , вершина v_7 – в слоях L_2 , L_3 , L_4 и L_5 , вершина v_8 – в слоях L_4 и L_5 , вершина v_{10} – в слоях L_3 и L_4 , вершина v_{13} – в слоях L_2 и L_3 . Однако положение одних вершин зависит от положения других. В частности, вершины, связанные дугой, не могут находиться в одном слое.

Для окончательного распределения вершин по слоям так, чтобы сумма весов слоев была по возможности минимальной, предлагается следующий способ. Удалив из орграфа G неподвижные вершины вместе с инцидентными им ребрами, получим оргграф H , в каждой компоненте которого выделим вершину с максимальным весом. Эту вершину поместим в один из допустимых для нее слоев с максимальным весом. Границы положения вершин при этом изменятся, и некоторые вершины из подвижных перейдут в неподвижные. Дальнейшее распределение по слоям можно вести для каждой компоненты орграфа H описанным выше способом.

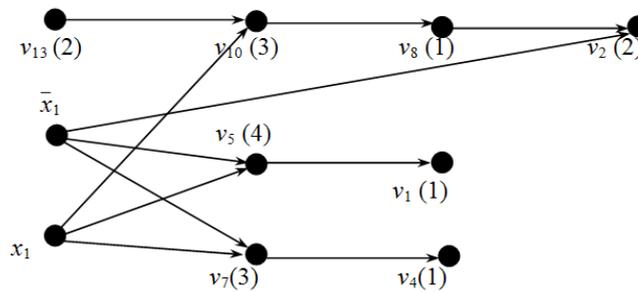


Рис. 4. Оргграф H , полученный из орграфа G

Для рассматриваемого примера оргграф H , состоящий из одной компоненты, показан на рис. 4. Вершины x_1 и \bar{x}_1 с нулевым весом могут быть помещены в любой слой. Вершину v_5 с максимальным весом согласно указанному способу поместим в слой L_3 . Окончательно получим $L_1 = \{x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, x_4\}$, $L_2 = \{v_{12}, v_{13}, v_{14}\}$, $L_3 = \{v_5, v_7, v_{10}, v_{11}\}$, $L_4 = \{v_1, v_4, v_8, v_9\}$, $L_5 = \{v_2, v_6\}$ и $L_6 = \{v_3\}$.

4. Выравнивание путей в орграфе

Приведем все пути в графе к единой длине путем добавления новых вершин с нулевым весом. В результате исходный оргграф $G = (V, A)$ преобразуется в оргграф $G' = (V', A')$. В оргграф G добавляются новые вершины так, чтобы выполнялось $N^+(v) \subseteq L_{i+1}$ для любых $v \in L_i$ и $i = 1, 2, \dots, m-1$ при сохранении достижимости вершин. Под сохранением достижимости понимаем то, что во всяком пути из вершины $v \in V$ к вершине $u \in V$ в орграфе G' встречаются те же вершины из V в том же порядке, что и в соответствующем пути в орграфе G . Новые вершины снабжаются нулевым весом. Для этого используется алгоритм 2. Преобразованный таким образом оргграф изображен на рис. 5, где добавленные вершины обозначены светлыми кружками.

Алгоритм 2

- 1) $i := 0, j := 0; U := \emptyset;$
- 2) $i := i + 1$. Если $i = m$, перейти к 4,
иначе $L := L_i;$
- 3) если $L = \emptyset$, перейти к 2,
иначе выбрать $v \in L, L := L \setminus \{v\}, A := N^+(v) \setminus L_{i+1}$. Если $A = \emptyset$, перейти к 3,
иначе $j := j + 1, U := U \cup \{u_j\}, N^+(v) := (N^+(v) \cap L_{i+1}) \cup \{u_j\}, N^+(u_j) := A,$
 $L_{i+1} := L_{i+1} \cup \{u_j\}$, перейти к 3;
- 4) конец.

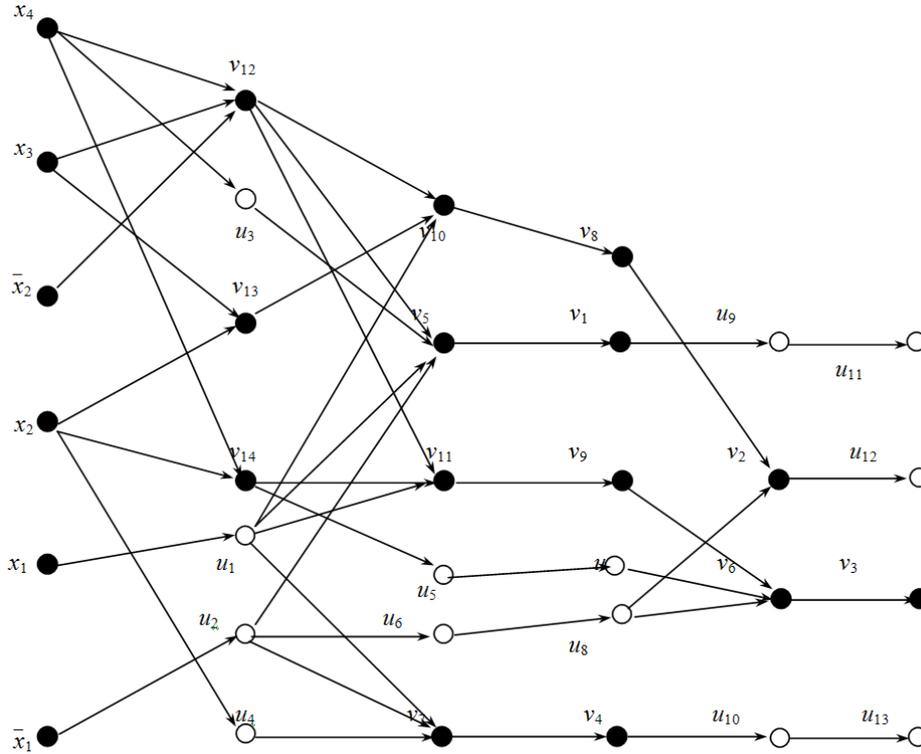


Рис. 5. Орграф G' с выравненными длинами путей

5. Разбиение схемы на каскады

Каждому из слоев соответствует множество значений веса, приписанных вершинам, которые принадлежат данному слою. Максимальное значение веса в этом слое представляет собой задержку прохождения сигнала в слое. Заданную комбинационную схему надо разбить на заданное число каскадов с минимизацией задержки в самом медленно действующем каскаде. Каждый каскад представляет собой упорядоченное множество слоев. Возникает следующая задача.

Дана последовательность положительных чисел (a_1, a_2, \dots, a_n) . Назовем отрезком последовательности ее часть вида $(a_p, a_{p+1}, \dots, a_q)$, где $1 \leq p < q \leq n$. Требуется разбить заданную последовательность на заданное число k отрезков B_1, B_2, \dots, B_k , где $B_i = (a_{n_{i-1}+1}, \dots, a_{n_i})$, $i = 1, 2, \dots, k$, $n_0 = 0$, $n_1, n_2, \dots, n_k = n$, которым соответствуют S_1, S_2, \dots, S_k , где $S_i = \sum_{a_j \in B_i} a_j$, так, чтобы величина $\max\{S_1, S_2, \dots, S_k\}$ была минимальной.

Элементы B_i данной последовательности соответствуют слоям в заданной схеме, а каждый из них равен максимуму веса в соответствующем слое.

Предлагается следующий метод получения решения данной задачи, близкого к оптимальному. Сначала определяется нижняя граница максимальной задержки в каскаде как

$$b = (\sum_{j=1}^n a_j) / n. \text{ Очередной } i\text{-й отрезок формируется путем накопления суммы } S_i = \sum_{a_j \in B_i} a_j, \text{ кото-}$$

рая сравнивается с b . При $S_i > b$ после добавления очередного a_j к S_i , если $b - (S_i - a_j) > S_i - b$ (недобор до b больше перебора), то крайним правым элементом в отрезке B_i устанавливается a_j , в противном случае — a_{j-1} . Та же процедура повторяется для оставшейся неразделенной части последовательности.

Ниже представлен алгоритм 3, получающий решение для случая, когда $(\sum_{j=1}^k S_j) / k > a_i$,

для любого $i = 1, 2, \dots, n$. Решение этой задачи представим в виде последовательности n_1, n_2, \dots, n_k , где n_i – номер крайнего правого элемента в i -м отрезке. В данном алгоритме τ – максимальная задержка в каскаде, S – среднее значение S_i , B – текущее значение S_i .

Алгоритм 3

- 1) $i := 0, R := 0, \tau := 0, l := k, n_k := n$;
- 2) если $i = n$, перейти к 3,
иначе $i := i + 1, R := R + a_i$, перейти к 2;
- 3) $S := R / l, i := 0, B := 0, j := 1$;
- 4) $i := i + 1$. Если $i = n$, то перейти к 6,
иначе $C := B, B := B + a_i$. Если $S > B$, перейти к 4,
иначе $D := S - C, E := B - S$. Если $D < E$, то $j := j + 1, n_j := i - 1, B := a_i$, перейти к 5,
иначе $j := j + 1, n_j := i, C := B, B := 0$;
- 5) $l := l - 1, R := R - C, S := R / l$. Если $\tau < C$, то $\tau := C$, перейти к 4,
иначе перейти к 4;
- 6) конец.

В рассматриваемом примере последовательность чисел (0, 2, 4, 1, 3, 1) представляет задержки в слоях. Допустим, эту последовательность надо разбить на три отрезка ($k = 3, n = 6$). Результатом выполнения алгоритма 3 является последовательность $(n_1, n_2, n_3) = (2, 3, 6)$, а максимальная задержка в каскаде – 5.

Триггеры в схеме должны стоять после второго, третьего и шестого слоев, на выходе каждого из крайних правых элементов каждого каскада. Соответствующим образом преобразованный орграф показан на рис. 6, где вершины, соответствующие триггерам, помечены символами d_i . Вершины, представляющие фиктивные элементы с нулевой задержкой, должны быть удалены из орграфа.

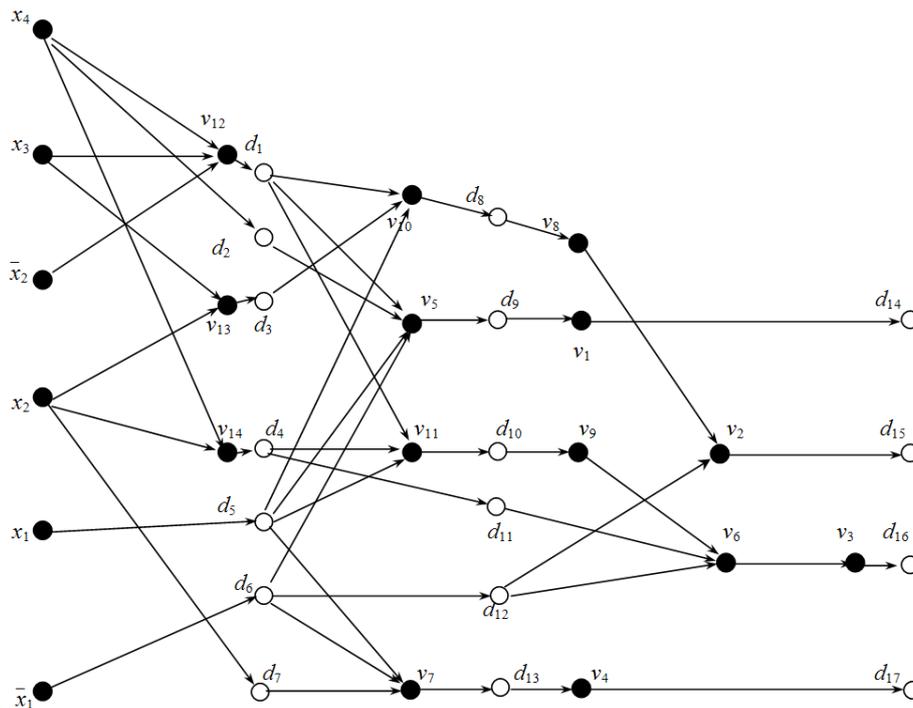


Рис. 6. Орграф с вершинами, соответствующими триггерам

Результат преобразования схемы, представленной на рис. 1, в схему, работающую в режиме конвейеризации, изображен на рис. 7, где вместо инверторов используются инверсные выходы триггеров.

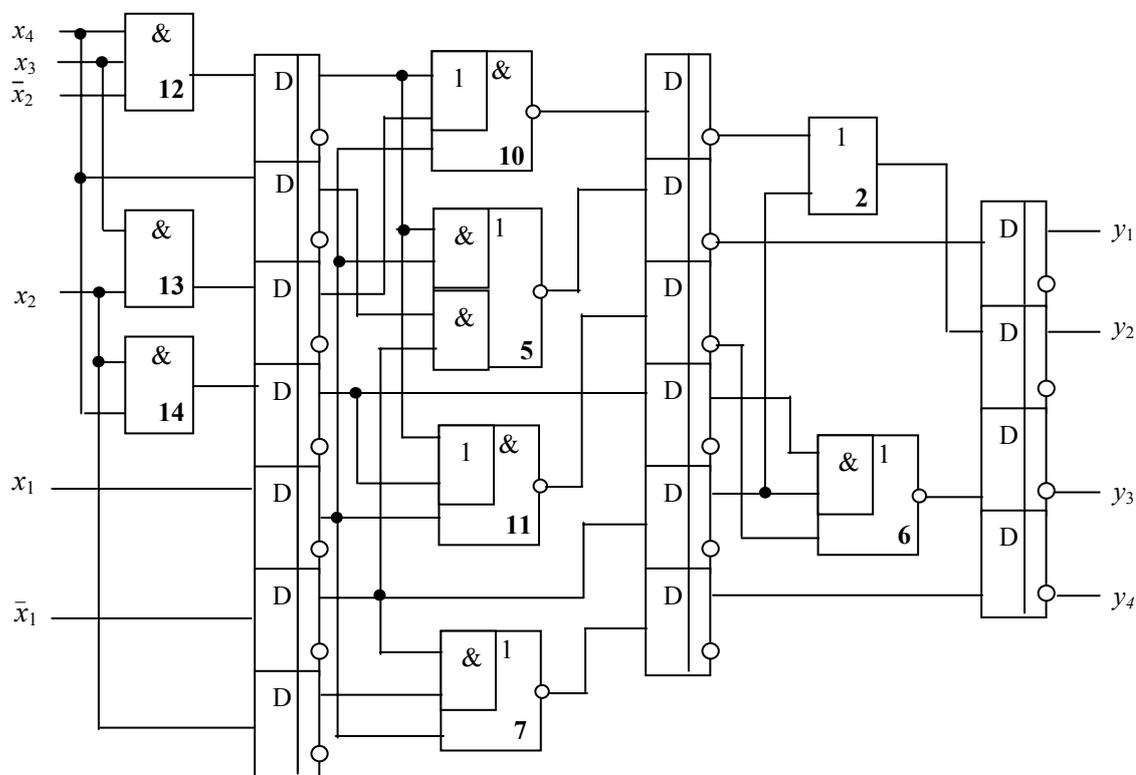


Рис. 7. Схема, работающая в конвейерном режиме

Допустим, что задержка триггера составляет четыре условные единицы, используемые при оценке задержки в каскадах схемы из приведенного примера. Задержка самого медленного каскада равна пяти. Следовательно, длительность такта синхронизации должна быть не менее девяти. Согласно формуле (2) ускорение обработки сигналов может быть достигнуто при длине входной последовательности не менее 14. Полная реакция исходной схемы на такую последовательность будет через 154 единицы времени, тогда как на такую же последовательность конвейеризованная схема выдаст реакцию через 153 единицы времени. Из формулы (1) видно, что разница между этими величинами будет увеличиваться с увеличением длины входной последовательности.

Заключение

Предлагаемый подход к повышению быстродействия комбинационной схемы рассчитан на применение его к готовым схемам и решает данную задачу на схемотехническом уровне. Возможно, на функциональном уровне можно добиться более значительного эффекта. Однако в этом случае необходимо заново проектировать схему исходя из ее функционального описания.

Список литературы

1. Каган, Б.М. Цифровые вычислительные машины и системы / Б.М. Каган, М.М. Каневский. – М. : Энергия, 1973. – 680 с.
2. Воеводин, В.В. Математические модели и методы в параллельных процессах / В.В. Воеводин. – М. : Наука. Гл. ред. физ.-мат. лит., 1986. – 296 с.
3. Капитонова, Ю.В. Математическая теория проектирования вычислительных систем / Ю.В. Капитонова, А.А. Летичевский. – М. : Наука. Гл. ред. физ.-мат. лит., 1988. – 296 с.

4. Кухарев, Г.А. Систолические процессоры для обработки сигналов / Г.А. Кухарев, А.Ю. Тропченко, В.П. Шмерко. – Минск : Беларусь, 1988. – 127 с.
5. Кухарев, Г.А. Алгоритмы и систолические процессоры для обработки многозначных данных / Г.А. Кухарев, В.П. Шмерко, Е.Н. Зайцева. – Минск : Навука і тэхніка, 1990. – 296 с.
6. Лукошко, Г. КМОП-базовые матричные кристаллы серии К1574 / Г. Лукошко, Е. Коннов // Радиолобитель. – 1997. – № 9. – С. 39–40.

Поступила 28.11.12

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

Yu.V. Pottosin, S.N. Kardash

INCREASING COMBINATIONAL CIRCUIT PERFORMANCE VIA PIPELINING

The question of increasing performance of a device with no memory, which develops a sequence of discrete signals, is considered. A problem is set to divide a given multilevel combinational circuit into a given number of cascades with registers providing pipeline-wise development of incoming signals. To solve this problem we use a model based on representation of combinational circuit as a directed graph. In the process of solving this problem, the frequency of incoming signals is established. This frequency must be as high as possible.