

П.Н. Бибило, Н.А. Кириенко

Применение конвейеризации для увеличения быстродействия логических схем

Представлены алгоритм и программа конвейеризации комбинационных нерегулярных логических схем с целью повышения их быстродействия. Приведены результаты исследования по определению характеристик конвейеризованных схем и выводы об усложнении конвейеризованной схемы и потребляемого ею тока.

An algorithm and a programme for pipelining of circuits of logic elements to increase the speed of logic circuits are described. The results of the pipelining circuits research are presented. The conclusions about significant increase in complexity and power consumption of pipelining circuit are made.

Подано алгоритм і програму конвеєризації комбінаційних нерегулярних логічних схем з метою підвищення їх швидкодії. Подано результати дослідження з визначення характеристик конвеєризованих схем і висновки про ускладнення конвеєризованої схеми і спожитого нею струму.

Введение. Сегодня широко применяются промышленные синтезаторы логических схем по алгоритмическим описаниям их поведения на языках высокого уровня, таких как *Verilog* и *VHDL*. Например, для заказных СБИС широко известным синтезатором логических схем есть *LeonardoSpectrum* (фирма *Mentor Graphics*) [1], позволяющий получать логические схемы в целевой библиотеке пользователя (проектировщика). При этом проектировщик может задавать различные технологические ограничения, требования к быстродействию, критерии оптимизации и т.д. Для некоторых специальных применений требуются логические схемы, обеспечивающие максимальное быстродействие, определяемое используемой библиотекой логических элементов, однако такие опции синтеза в синтезаторах, как правило, отсутствуют. В данной статье достижение максимальных показателей быстродействия комбинационных (без элементов памяти) нерегулярных логических схем предлагается осуществлять с помощью конвейеризации.

Получение конвейеризованных логических схем

Получение конвейеризованных логических схем осуществляется в два этапа.

Этап 1. Синтез нерегулярных комбинационных схем с помощью синтезаторов (*LeonardoSpectrum* либо других) и получение струк-

турных *VHDL*-описаний полученных схем в базе одновыходных логических элементов.

Этап 2. Преобразование полученных описаний (нетлистов) логических схем в конвейеризованные структуры, для которых функционирование осуществляется по систолическому принципу: все сигналы из выходов одного уровня (блока) конвейера поступают одновременно на вход следующего блока конвейера.

Для выполнения этапа 2 разработана программа *KONV*, проведено ее экспериментальное исследование. Исходные данные для программы *KONV* – нетлисты нерегулярных логических схем из одновыходных библиотечных элементов и описание триггера, который должен быть использован для конвейеризации. Результатирующими данными являются нетлисты конвейеризованных схем. Пример исходной логической схемы показан на рис. 1, результат конвейеризации – на рис. 2.

Для преобразования описания схемы в описание конвейеризованной структуры выполняется следующее.

- Проводится разбиение элементов схемы на каскады.
- Для уменьшения нагрузочной способности (увеличения быстродействия) проводится дублирование логических элементов так, чтобы сигнал с выхода любого логического элемента поступал на вход только одного логиче-

ского элемента следующего каскада. Так, для рассматриваемого примера логической схемы элемент *dd3* (рис. 1) будет дополнительно два раза продублирован элементами такого же типа *NA*, в результирующей схеме добавленные элементы имеют имена *dd3L1L*, *dd3L2L*.

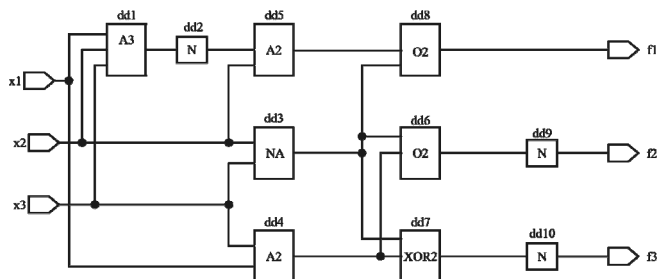


Рис. 1. Логическая схема до конвейеризации

• Устанавливаются триггеры на выходах логических элементов и на линиях соединений, проходящих через каскады, так, чтобы сигналы с выходов элементов (и соединительных линий) каскада поступали на входы триггеров. Заметим, что на все триггеры подается один и тот же синхросигнал, что позволяет синхронизировать работу всех триггеров схемы. На рис. 2

общий синхросигнал для всех триггеров схемы не показан.

• Удаляются инверторы в схеме, если используемый для конвейеризации триггер имеет инверсный выход. При исключении инвертора используются инверсный выход того триггера, у которого сигнал с прямого выхода подается на инвертор. На рис. 2 новые связи элементов после исключения инверторов показаны пунктирными линиями.

Очевидно, что после введения в схему большого числа триггеров, синхронизируемых одним синхросигналом, может возникать проблема расфазировки тактовых импульсов. Для решения этой проблемы применяются известные подходы, описанные, например, в работе [2].

Быстродействие результирующей конвейеризованной логической схемы определяется задержкой самого «медленного» элемента, имеющегося в схеме, эта задержка и определит длину периода синхросигнала для триггеров. Синхронизация общая, поэтому в каждом такте на вход начального блока может быть подана следующая комбинация входных сигналов.

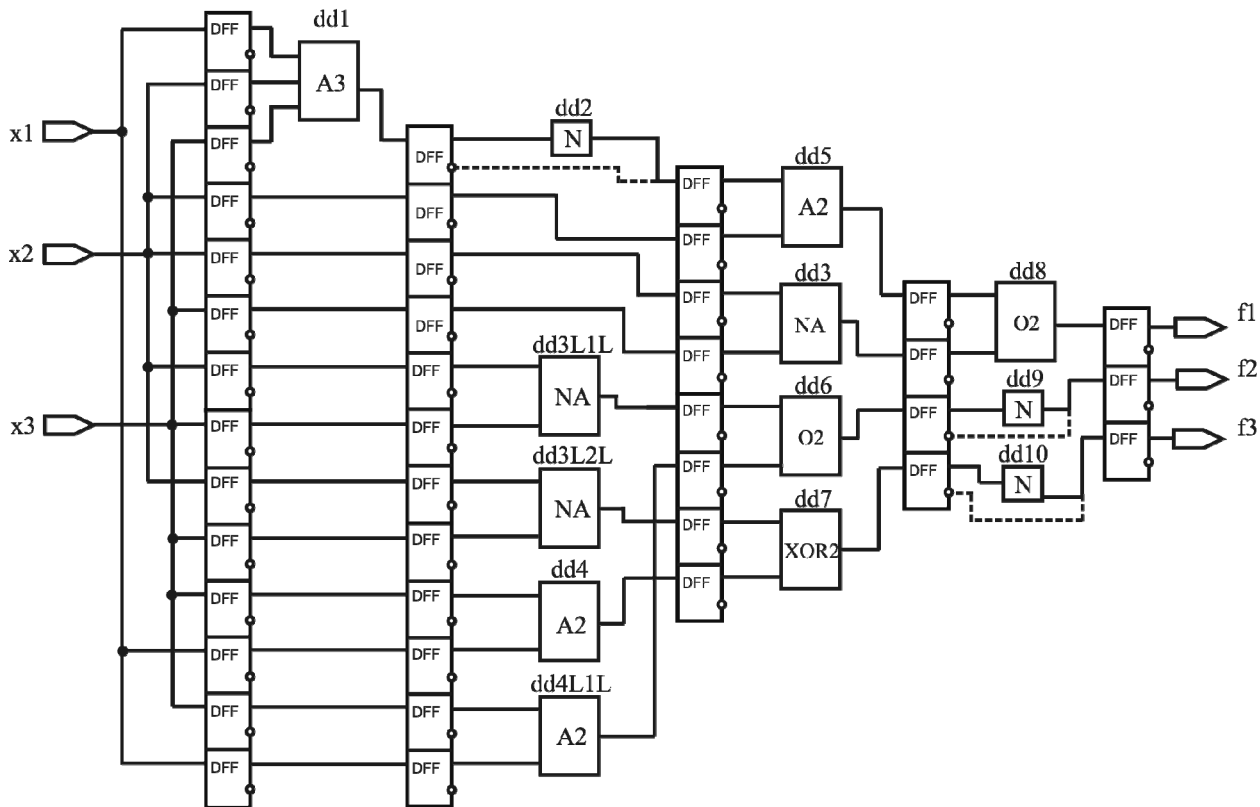


Рис. 2. Логическая схема после конвейеризации (все синхронные *D*-триггеры *DFF* имеют общий синхросигнал)

Таким образом, конвейер позволяет выдавать в каждом такте выходные сигналы. В начале функционирования, естественно, имеется задержка, определяемая числом каскадов схемы (глубиной конвейера). На рис. 2 число регистров равно пяти, через пять тактов синхросигнала схема выдаст на выходах значения для первого набора входных сигналов.

Экспериментальное исследование конвейеризованных логических схем

Авторами статьи проведено экспериментальное исследование программы *KONV* на потоке практических примеров. Схемы для экспериментов выбирались из набора тестовых примеров [3], в качестве библиотеки логических элементов использовалась библиотека КМОП-элементов, представленная в [4], в качестве триггеров выбраны синхронные *D*-триггеры. Цель экспериментов – определение характеристик схем после проведения конвейеризации, а именно: насколько увеличится сложность схемы и потребляемый ток. Исследовались также методы синтеза (синтезаторы) с целью выбора более предпочтительного, т.е. позволяющего получать конвейеризованные схемы с лучшими характеристиками.

В проведенных экспериментах синтез на этапе 1 выполнялся двумя различными синтезаторами. Схема синтезировалась двумя способами – в синтезаторе *LeonardoSpectrum* и в системе (синтезаторе) *СиВер* [5]. Синтез в *СиВер* включал два шага: сначала осуществлялась предварительная технологически независимая оптимизация на основе *BDD* (*Binary Decision Diagram* – диаграмма двоичного выбора), выполняемая с помощью программы [6], затем оптимизированные представления покрывались логическими элементами. Измерение энергопотребления логических схем осуществлялось с помощью схемотехнического моделирования описаний схем на транзисторном уровне в системе *Accusim* (фирма *Mentor Graphics*).

Результаты экспериментов представлены в таблице, где L – число элементов в схеме до конвейеризации; $L_{\text{дубл}}$ – число элементов схемы после дублирования элементов; $L_{\text{рез}}$ – число элементов (включая триггеры) в конвейеризованной схеме; p – число каскадов (уровней) схе-

мы; R – суммарное число транзисторов во всех элементах схемы до конвейеризации; $R_{\text{рез}}$ – суммарное число транзисторов во всех элементах схемы, включая триггеры, после конвейеризации; A – средний потребляемый исходной схемой ток в миллиамперах (мА); $A_{\text{рез}}$ – средний потребляемый конвейеризованной схемой ток (мА).

Оценка энергопотребления конвейеризованных логических схем

Имя схемы	L	$L_{\text{дубл}}$	$L_{\text{рез}}$	p	R	$A, \text{мА}$	$R_{\text{рез}}$	$A_{\text{рез}}, \text{мА}$
Синтезатор LeonardoSpectrum								
<i>add6</i>	1176	2662	24664	21	6592	4,94	715564	780
<i>b12</i>	81	129	1144	10	426	0,44	33050	37,9
<i>dist</i>	301	655	4136	14	1588	0,93	114652	102
<i>life</i>	68	117	884	13	352	0,35	25106	27,83
<i>mlp4</i>	461	1024	6697	15	2392	1,05	185998	157
<i>root</i>	155	340	2186	14	796	0,68	60720	62,25
<i>ryy6</i>	11	11	51	4	74	0,08	1354	1,68
<i>z5xp</i>	410	924	5573	14	2168	1,19	153758	135
<i>z9sym</i>	158	340	2675	19	854	0,77	76294	69,6
Среднее значение					1693,56	1,16	151832,89	152,58
Синтезатор СиВер								
<i>add6</i>	285	402	3551	26	1238	2,75	102428	85,12
<i>b12</i>	186	239	1424	14	776	1,21	38840	42,49
<i>dist</i>	585	811	3770	16	2554	6,31	97948	100
<i>life</i>	104	131	581	17	434	0,88	14940	15,95
<i>mlp4</i>	516	746	3908	16	2294	5,03	104172	114
<i>root</i>	266	351	1570	15	1106	2,13	40378	43,41
<i>ryy6</i>	61	69	559	29	234	0,46	15948	13,75
<i>z5xp1</i>	221	306	1641	13	956	1,78	43918	46,68
<i>z9sym</i>	138	178	725	17	580	1,23	18232	20,49
Среднее значение					1130,22	2,42	52978,22	53,54

Число добавленных в схему комбинационных элементов равно $L_{\text{дубл}} - L$, число добавленных триггеров равно $L_{\text{рез}} - L_{\text{дубл}}$. Для схем, представленных на рис. 1 и 2, $L = 10$, $L_{\text{дубл}} = 13$, $L_{\text{рез}} = 54$. Лучшие решения (схемы с меньшим числом транзисторов и потребляющие меньший ток) обозначены в таблице жирным шрифтом.

Закключение. Проведенный эксперимент позволяет сделать следующие выводы.

Для данного потока примеров схем синтезатор *СиВер* в сравнении с синтезатором *LeonardoSpectrum* позволил получить схемы, характеризующиеся «в среднем» меньшим числом транзисторов и, как следствие, меньшим потребляемым током.

Основное число транзисторов в конвейеризованных схемах приходится на добавленные *D*-триггеры.

Так как число каскадов в схемах может достигать двух и более десятков, то основной вывод по эксперименту заключается в том, что повышение быстродействия схемы в несколько десятков раз приводит при использовании конвейеризации к увеличению сложности схемы и потребляемого тока также в десятки раз. К примеру, рассмотрим результаты конвейеризации схемы *z9sum*, полученной с помощью синтезатора СиВер и имеющей 17 каскадов. Все элементы схемы *z9sum* содержат до конвейеризации 580 транзисторов, потребляемый ток составляет 1,23 миллиампера. После конвейеризации во всех элементах схемы (включая добавленные триггеры) *z9sum* имеется 18232 транзисторов (увеличение примерно в 30 раз), а величина потребляемого тока составляет 20,49 миллиампера (увеличение почти в 17 раз).

Повышение быстродействия во многих случаях – определяющий фактор, ради которого приходится жертвовать ухудшением других параметров схемы. Разработанная программа конвейеризации позволит проектировщику на этапе логического проектирования оценить ап-

паратурные затраты для достижения лучшего быстродействия схем. Актуален и вопрос о том, какие методы предварительной оптимизации и какие методы синтеза наиболее предпочтительны для получения экономичных результирующих конвейеризованных структур, характеризующихся меньшим энергопотреблением.

1. Бибилло П.Н. Системы проектирования интегральных схем на основе языка *VHDL*. *StateCAD, ModelSim, LeonardoSpectrum*. – М.: СОЛОН-Пресс, 2005. – 384 с.
2. Рабау Ж.М., Чандракасан А., Николитч Б. Цифровые интегральные схемы. – М.: Вильямс, 2007. – 912 с.
3. Berkeley PLA test set. – <http://www1.cs.columbia.edu/~cs4861/sis/espresso-examples/ex/>
4. Бибилло П.Н., Кириенко Н.А. Оценка энергопотребления логических КМОП-схем по их переключательной активности // Микроэлектроника. – 2012. – № 1 – С. 65–77.
5. Бибилло П.Н., Кардаш С.Н., Романов В.И. СиВер – система синтеза и верификации комбинационных логических схем. // Информатика. – 2006. – № 4. – С. 79 – 87.
6. Бибилло П.Н., Леончик П.В. Алгоритм построения диаграммы двоичного выбора для системы полностью определенных булевых функций. // УСиМ. – 2009. – № 6. – С. 42–49.

Поступила 04.06.2013

Тел. для справок: +37 517 284-2084, 284-2076 (Минск)
E-mail: bibilo@newman.bas-net.by, kir@newman.bas-net.by

© П.Н. Бибилло, Н.А. Кириенко, 2013

Окончание статьи А.Гаиш Якуба и др.

слению такого же дерева для пяти произвольных точек, что и будет представлено в дальнейших работах.

1. Агаи Аг Гаиш Якуб, Донец Г.А. О некоторых аспектах классической задачи Штейнера // УСиМ. – 2013. – № 5. – С. 27–31.
2. Brandstädt A., Kratsch D. On the restriction of some NP-complete graph problems to permutation graphs // Proc. of the FCT'85 Conf. – Berlin, 1985. – P. 53–62.
3. Garey M.R., Johnson D.S., Stockmeyer L. Some simplified NP-complete graph problems // Theor. Comp. Sci. – 1976. – 1. – P. 237–267.
4. Котов В.М. О задаче Штейнера // Вестн. Бел. гос. ун-та. – Минск, 1982. – 10 с. Деп в ВИНТИ 29.07.82 г., № 4070–82.

Поступила 25.01.2013

Тел. для справок: +38 044 526-2188 (Киев)
E-mail: j_donets@mail.ru

© Агаи Аг Гаиш Якуб, Г.А. Донец, 2013

В этом случае точка P_4 принадлежит лунке, соединяющей точку O и вершину треугольника P_{i+2} . Как показано ранее, в этом случае дерево Штейнера единственное. Во втором случае

$$\begin{aligned} |P_i P_{i+1}|^2 &\geq |P_i P_4|^2 + |P_{i+1} P_4|^2 + |P_i P_4| \cdot |P_{i+1} P_4|, \\ |P_{i+1} P_{i+2}|^2 &\geq |P_{i+1} P_4|^2 + |P_{i+2} P_4|^2 + |P_{i+1} P_4| \cdot |P_{i+2} P_4|, \\ |P_{i+2} P_{i+3}|^2 &\leq |P_{i+2} P_4|^2 + |P_{i+3} P_4|^2 + |P_{i+2} P_4| \cdot |P_{i+3} P_4|. \end{aligned} \quad (16)$$

Здесь точка P_4 находится внутри фигуры, ограниченной двумя лунками и стороной треугольника $P_{i+2} P_{i+3}$. Ранее показано, что здесь возможны два альтернативных дерева Штейнера: первое состоит из отрезка $P_{i+3} P_4$ и построения Штейнера внутри $\Delta P_{i+1} P_{i+2} P_4$, а второе – из отрезка $P_4 P_{i+2}$ и построения Штейнера внутри $\Delta P_i P_{i+1} P_4$.

Заключение. Умея вычислять дерево Штейнера для четырех произвольных точек, можно переходить к вычи-