

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ДИСКРЕТНЫХ АВТОМАТОВ

DOI 10.17223/20710410/14/8

УДК 519.7

КОДИРОВАНИЕ СОСТОЯНИЙ ДИСКРЕТНОГО АВТОМАТА, ОРИЕНТИРОВАННОЕ НА УМЕНЬШЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ РЕАЛИЗУЮЩЕЙ СХЕМЫ

Ю. В. Поттосин

*Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь***E-mail:** pott@newman.bas-net.by

Рассматривается задача кодирования состояний дискретного автомата с целью уменьшения интенсивности переключений элементов памяти в реализующей схеме. Предлагается метод решения этой задачи, сочетающий идеи методов «желательных соседств» и «сборки» булева гиперкуба.

Ключевые слова: дискретный автомат, кодирование состояний, энергосбережение.

Введение

В последнее время при проектировании дискретных устройств управления на основе сверхбольших интегральных схем большое внимание уделяется проблеме снижения энергопотребления проектируемой схемы. Это обусловлено стремлением, с одной стороны, увеличить время действия источника энергии в портативных приборах и, с другой стороны, снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой схемой энергии. Как отмечено в работах [1, 2], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна частоте изменения сигналов. Это дает возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться при кодировании состояний и декомпозиции автомата [3–5]. Кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти.

При решении задачи кодирования состояний автомата в работе [3] использован информационный подход, при котором учитываются вероятности состояний. В работе [5] эта задача сводится к укладке графа поведения автомата в полный булев граф, или булев гиперкуб. Наибольшая эффективность метода из работы [5] достигается, когда граф, полученный из графа поведения автомата удалением ориентации и кратности ребер, оказывается подграфом полного булева графа, т. е. он полностью укладывается в гиперкуб. Тогда при каждом переходе автомата только один элемент памяти изменяет свое состояние. В противном случае минимизируется число ребер, не укладываемых в гиперкуб. Алгоритмизация этого метода осуществлена в работе [6], где описан еще один алгоритм кодирования состояний, который основан на использовании матрицы

смежности графа и карты Карно и заключается в построении последовательности конфигураций из ребер и квадратов, образующих подграфы гиперкуба. Под квадратом понимается цикл длины 4 в графе.

Для решения задачи энергосберегающего кодирования состояний автомата можно использовать метод «желательных соседств» [7, 8], направленный на получение наиболее простой схемы. При этом минимизируется величина $W = \sum w_{ij}d_{ij}$, где суммирование ведется по всем парам состояний заданного автомата, а d_{ij} — расстояние по Хэммингу между кодами состояний q_i и q_j . В случае энергосберегающего кодирования в качестве w_{ij} надо выбрать некоторый показатель, связанный с переходами между состояниями q_i и q_j . Как отмечено еще в работе [7], абсолютный минимум величины W получить довольно сложно, да и не всегда он соответствует оптимальному решению конкретной задачи. Поэтому чаще используются некоторые эвристики при размещении состояний автомата в булевом пространстве. В частности, для решения данной задачи можно использовать предлагаемый в работе [9] алгоритм размещения вершин графа, которым может быть гиперкуб, в простой цепи с минимизацией той же суммы W , только в ней w_{ij} — некоторая функция на ребрах графа, а d_{ij} — расстояния в заданной простой цепи.

В настоящей работе предлагается метод кодирования состояний автомата, приводящий к снижению переключательной активности элементов памяти в реализующей схеме. Он отличается от известных методов способом размещения состояний автомата в булевом пространстве внутренних переменных. Используется подход, изложенный в работах [10, 11] и основанный на методе из [7, 8]; размещение состояний в булевом пространстве представляется как процесс построения k -мерного булева гиперкуба, напоминающий сборку некоторой простой механической конструкции. Здесь k — число элементов памяти в проектируемой схеме.

1. Построение булева гиперкуба

Формально задачу можно поставить следующим образом. Пусть задан автомат M , состояния которого, образующие множество $Q = \{q_1, q_2, \dots, q_\gamma\}$, должны быть закодированы булевыми векторами так, чтобы минимизировать сумму $W = \sum w_{ij}d_{ij}$, где суммирование ведется по всем парам состояний заданного автомата; d_{ij} — расстояние по Хэммингу между кодами состояний q_i и q_j ; $w_{ij} = w(q_i, q_j)$; w — функция, принимающая вещественные значения на множестве пар состояний из Q . Смысл функции w и ее определение зависят от конкретной задачи. В работах [10, 11] она выбрана из соображений получения наиболее простой схемы автомата. В случае энергосберегающего кодирования функция w должна отражать интенсивность переключений элементов памяти. Так же, как в [6], в качестве w возьмем целочисленную функцию, значения которой для пар q_i, q_j пропорциональны вероятностям перехода между состояниями q_i и q_j , неважно, в каком направлении. Эту вероятность можно вычислить методом Чэпмена — Колмогорова [12], который подробно описан также в [6].

Критерием качества размещения состояний в вершинах булева гиперкуба можно считать введенную в [6] величину, названную дефектом отображения (ребер графа переходов на ребра гиперкуба), которая в терминах излагаемого метода вычисляется по формуле $D = \sum w_{ij}(d_{ij} - 1)$, где суммирование берется по всем парам состояний, соответствующим парам вершин в гиперкубе. Очевидно, чем меньше значение D , тем лучше результат размещения, и дефект отображения равен нулю, если всем парам состояний, связанным переходами, соответствуют ребра гиперкуба. Тогда при любом переходе из состояния в состояние переключается ровно один элемент памяти.

Предлагаемый метод является эвристическим, т. е. не гарантирует получения абсолютного минимума суммы W . Он основан на том соображении, что для того, чтобы величина W не сильно отличалась от минимума, надо, чтобы состояния q_i и q_j , для которых значение w_{ij} велико, кодировались по возможности близкими кодами.

Исходными данными для построения n -мерного гиперкуба являются значения функции w и число состояний γ заданного автомата M . Полагается, что это число минимальное или по какой-то причине его не надо минимизировать. Если γ не является целой степенью двойки, его надо путем введения фиктивных состояний увеличить до 2^n , где $n = \lceil \log_2 \gamma \rceil$. Считаем, что $w_{kl} = 0$, если хотя бы одно из состояний q_k или q_l является фиктивным. В начале процесса вершины, которые представляют состояния автомата M и должны быть вершинами гиперкуба, образуют пустой граф (без ребер).

Построение n -мерного гиперкуба представляется как последовательность n шагов. На s -м шаге рассматривается множество $(s-1)$ -мерных гиперкубов, они объединяются в пары, и из каждой пары получается один s -мерный гиперкуб путем соответствующего добавления ребер. При этом, по возможности, для соединения ребрами выбираются те вершины q_i и q_j , которым соответствуют наибольшие из оставшихся значения w_{ij} . В результате выполнения n шагов получим искомый n -мерный гиперкуб. Вершинам приписываем n -компонентные булевы векторы с соблюдением отношения соседства, представленного ребрами гиперкуба.

На первом шаге из γ изолированных вершин, или 0-мерных гиперкубов, строятся 1-мерные гиперкубы в виде $\gamma/2$ попарно несмежных ребер. На последнем n -м шаге из двух $(n-1)$ -мерных гиперкубов собирается один n -мерный гиперкуб путем добавления 2^{n-1} ребер.

Рассмотрим более подробно образование s -мерных гиперкубов на s -м шаге. Для компьютерной реализации метода важным моментом является способ представления гиперкубов. Любой k -мерный гиперкуб, являющийся подграфом n -мерного гиперкуба, можно представить последовательностью S из 2^k номеров вершин, взятых из множества $\{1, 2, \dots, 2^n\}$. Ребра заданы неявно: считается, что две вершины связаны ребром тогда и только тогда, когда занимаемые ими места в последовательности S соответствуют местам соседних кодов в равной по длине последовательности кодов Грея.

Перед выполнением любого шага число гиперкубов всегда четно, точнее, для s -го шага оно равно 2^{n-s+1} . Текущая ситуация характерна наличием некоторого множества s -мерных гиперкубов C_s (перед выполнением шага оно пусто) и некоторого остатка в виде множества $(s-1)$ -мерных гиперкубов C_{s-1} . Перебираются все пары гиперкубов из множества C_{s-1} и выбирается одна из них в соответствии с критерием, указанным ниже. К этой паре добавляются ребра, чтобы образовать s -мерный гиперкуб, который вводится в множество C_s . Выбранная пара исключается из C_{s-1} . Выполнение шага заканчивается, когда C_{s-1} оказывается пустым.

2. Объединение гиперкубов в пары

Для двух $(s-1)$ -мерных гиперкубов, представленных последовательностями S' и S'' , подсчитывается сумма $\sum w_{ij}$, где суммирование ведется по всем парам i и j номеров вершин, занимающих одноименные места в S' и S'' . Величина, представляемая указанной суммой, меняется с перестановкой вершин в одной из последовательностей, например в S'' . Конечно, только те последовательности могут быть приняты к рассмотрению, которые сохраняют отношение соседства между вершинами.

Выбирается та пара гиперкубов, для которой величина $\sum w_{ij}$ максимальна, и из них строится s -мерный гиперкуб соединением ребрами вершин, номера которых за-

нимают одноименные места в последовательностях S' и S'' (после соответствующей перестановки). Последовательность, представляющая построенный гиперкуб, получается соединением последовательностей S' и S'' , одна из которых меняет свой порядок на обратный.

Пусть, например, для восьми состояний получены два 2-мерных гиперкуба, которые представляются следующими последовательностями, связанными с кодом Грея: q_1, q_2, q_5, q_3 и q_4, q_6, q_8, q_7 . Первый гиперкуб имеет ребра q_1q_2, q_2q_5, q_3q_5 и q_1q_3 ; второй — q_4q_6, q_6q_8, q_7q_8 и q_4q_7 . При таком расположении вершин построенный с помощью добавления ребер q_1q_4, q_2q_6, q_5q_8 и q_3q_7 3-мерный гиперкуб представится последовательностью $q_1, q_2, q_5, q_3, q_7, q_8, q_6, q_4$. Если взять перестановку q_8, q_6, q_4, q_7 состояний второй последовательности, то получим $q_1, q_2, q_5, q_3, q_7, q_4, q_6, q_8$.

3. Сложность перебора вариантов объединения гиперкубов в пары

Рассмотрим перестановку, задаваемую оператором E_k , в результате действия которого все соседние по k -му измерению вершины меняются местами. Очевидные свойства операторов такого типа, $E_i E_k = E_k E_i$ и $E_i E_i = 1$, позволяют в каждом варианте пары гиперкубов перебирать еще 2^{s-1} различных вариантов, порождаемых перестановками вершин в одном из гиперкубов рассматриваемой пары, получаемыми путем применения всевозможных сочетаний операторов $E_1, E_2, \dots, E_{s-1}$.

Количество перебираемых вариантов пар гиперкубов на s -м шаге выражается следующей формулой:

$$L'_s = \sum_{i=1}^{2^{n-s}} (2^{n-s} - i + 1)(2^{n-s+1} - 2i + 1) = 2^{n-s-1}((2^{2(n-s+1)} - 1)/3 + 2^{n-s}).$$

Принимая во внимание перестановки, определяемые всевозможными сочетаниями операторов E_k ($k = 1, 2, \dots, s-1$) для каждой пары $(s-1)$ -мерных гиперкубов, получим количество перебираемых вариантов на s -м шаге:

$$L_s = 2^{s-1} L'_s = 2^{n-2}((2^{2(n-s+1)} - 1)/3 + 2^{n-s}).$$

Полный перебор всех возможных перестановок вершин $(s-1)$ -мерного гиперкуба с сохранением отношения соседства на s -м шаге, предусмотренный в работе [10] и допустимый при небольшом числе состояний автомата, резко увеличивает трудоемкость получения решения при увеличении числа состояний (число вариантов при этом составит $L'_s(s-1)! \cdot 2^{s-1}$). Поэтому данный случай рассматривать здесь не будем.

Окончательно получим число вариантов, перебираемых указанным способом в течение всего процесса построения n -мерного булева гиперкуба:

$$L = \sum_{s=1}^n L_s = 2^{n-2}((2^{2(n+1)} - 3n - 13)/9 + 2^n).$$

Поскольку 2^n в данном случае является числом состояний автомата, количество вариантов перебора при сборке n -мерного гиперкуба оценивается как полином степени 3 от числа состояний. Процесс выбора пары $(s-1)$ -мерных гиперкубов на s -м шаге может быть сокращен, если величина Σw_{ij} достигла верхней границы, которую легко подсчитать, взяв сумму наибольших значений w_{ij} для пар вершин, еще не связанных ребрами.

4. Примеры

Процесс сборки гиперкуба продемонстрируем на примере из [6]. Воспользуемся результатами вычислений функции w_{ij} , приведенными в данной работе, и представим значения w_{ij} в табл. 1 (значение w_{ij} пропорционально вероятности перехода между состояниями q_i и q_j), где строки и столбцы соответствуют состояниям автомата. К автомату с шестью состояниями добавляем два фиктивных состояния q_7 и q_8 , чтобы придать числу состояний значение целой степени двойки. Нулевые значения w_{i7} и w_{i8} в табл. 1 не представлены.

Т а б л и ц а 1
Значения w_{ij}

q_2	q_3	q_4	q_5	q_6	
18	11	0	7	1	q_1
	0	2	0	7	q_2
		7	15	2	q_3
			8	8	q_4
				0	q_5

Максимальное значение имеет $w_{12} = 18$. Поэтому в первую очередь соединяем ребром вершины q_1 и q_2 . Затем ребрами соединяются вершины q_3 и q_5 , q_4 и q_6 , q_7 и q_8 . Таким образом, на первом шаге получаем четыре одномерных гиперкуба, изображенных на рис. 1.

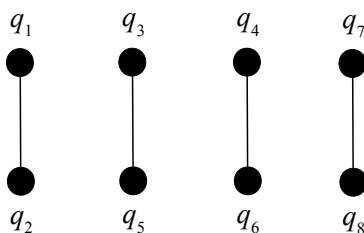


Рис. 1. Одномерные гиперкубы

Теперь требуется сформировать двумерный гиперкуб, добавив два ребра $q_i q_j$ и $q_k q_l$ так, чтобы величина $w_{ij} + w_{kl}$ была максимальной среди всех возможно добавляемых ребер. Все варианты присоединения ребер с соответствующей величиной $w_{ij} + w_{kl}$ приведены в табл. 2. Ребра, инцидентные вершинам q_7 и q_8 , не рассматриваются.

Т а б л и ц а 2
Варианты присоединения ребер на втором шаге

Ребра	$q_1 q_3, q_2 q_5$	$q_1 q_5, q_2 q_3$	$q_1 q_4, q_2 q_6$	$q_1 q_6, q_2 q_4$	$q_3 q_4, q_5 q_6$	$q_3 q_6, q_4 q_5$
$w_{ij} + w_{kl}$	11	7	7	3	7	10

Выбрав первый вариант, получим, как показано на рис. 2, один двумерный гиперкуб на вершинах q_1, q_2, q_3 и q_5 и второй гиперкуб путем добавления оставшихся ребер.

Для окончания работы требуется добавить еще четыре ребра, чтобы получить трехмерный гиперкуб. Варианты добавления ребер представлены в табл. 3. Среди них только два ребра могут иметь ненулевое значение w_{ij} . Поэтому рассматриваются только пары ребер.

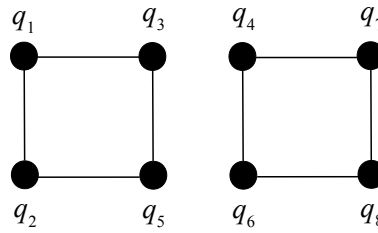


Рис. 2. Результат второго шага сборки гиперкуба

Таблица 3

Варианты присоединения ребер на третьем шаге

Ребра	q_1q_6, q_2q_4	q_1q_4, q_2q_6	q_2q_6, q_4q_5	q_2q_4, q_5q_6	q_3q_4, q_5q_6	q_3q_6, q_4q_5	q_1q_6, q_2q_4	q_1q_4, q_3q_6
$w_{ij} + w_{kl}$	3	7	15	2	7	10	8	2

Выбрав третий вариант, получим результат, показанный на рис. 3. Согласно отношению соседства, представленному графом на рис. 3, шесть состояний заданного автомата могут быть закодированы следующим образом: $q_1 - 000$, $q_2 - 001$, $q_3 - 010$, $q_4 - 111$, $q_5 - 011$, $q_6 - 110$.

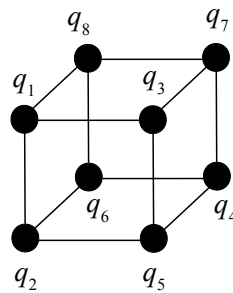


Рис. 3. Окончательный результат сборки

Получим дефект отображения $D = \sum w_{ij}(d_{ij} - 1)$. В рассмотренном примере к парам состояний, связанным переходами и соответствующим парам несвязных вершин в гиперкубе, относятся (q_1, q_5) , (q_1, q_6) , (q_2, q_4) , (q_3, q_6) и (q_3, q_4) . Соответствующими весами являются 7, 1, 2, 2 и 7, а расстояниями — 2, 2, 2, 3 и 2. В итоге получаем $D = 21$, что значительно меньше значения, полученного для того же примера в работе [6].

Для сравнения предлагаемого метода с методом квадратов, описанным в [6], рассмотрим другой из приведенных там примеров — автомат с 11 состояниями. В данном случае $w_{ij} = 1$, если состояния q_i и q_j связаны переходом, и $w_{ij} = 0$ в противном случае, поскольку вероятности переходов здесь не учитываются. Значения w_{ij} на парах состояний автомата удобно задать в виде табл. 4.

Для доведения числа состояний до целой степени двойки вводим пять фиктивных состояний и определим $w_{ij} = 0$, если хотя бы одно из q_i и q_j является фиктивным. Одномерные гиперкубы, получаемые на первом шаге, изображены на рис. 4, где ребра, у которых оба конца соответствуют фиктивным состояниям, не присутствуют.

Используя тот же способ, что и в предыдущем примере, вводим ребра q_1q_6 , q_3q_7 , q_2q_9 , q_4q_8 , q_5q_{11} и $q_{10}q_{12}$; получим три двумерных гиперкуба (четвертый гиперкуб с нулевыми весами ребер не представляет интереса), изображенных на рис. 5. При переборе

Т а б л и ц а 4

Значения w_{ij}

q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	
0	1	0	0	1	0	1	0	0	0	q_1
	0	1	0	1	0	0	1	0	1	q_2
		0	0	1	1	0	1	0	0	q_3
			1	0	0	1	0	0	0	q_4
				0	0	0	0	1	1	q_5
					1	0	0	1	0	q_6
						0	1	0	0	q_7
							1	0	0	q_8
								0	0	q_9
									1	q_{10}

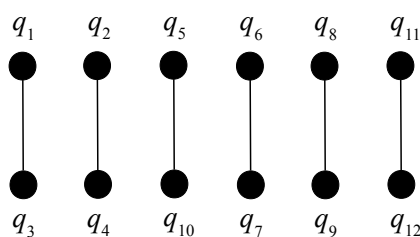


Рис. 4. Одномерные гиперкубы

вариантов введения ребер следует учитывать верхнюю границу суммы их весов, на данном шаге равную 2.

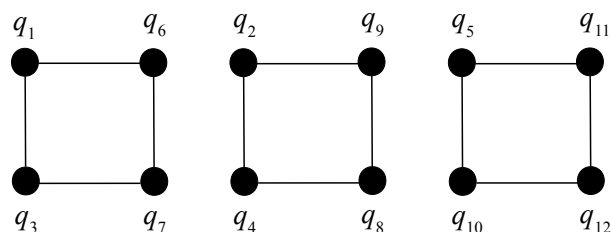


Рис. 5. Двумерные гиперкубы

Следующим шагом является построение двух трехмерных гиперкубов путем введения двух четверок ребер. При этом в данном примере достаточно рассматривать только такую четверку, в которой каждое ребро имеет не более одного конца, соответствующего фиктивному состоянию.

Все рассматриваемые варианты присоединения ребер с соответствующей величиной суммы их весов приведены в табл. 5, которая имеет три секции, соответствующие трем парам гиперкубов. Первая строка соответствует исходному положению гиперкубов, последующие строки — перестановкам, определяемым операторами E_1, E_2 и их сочетанию $E_1 E_2$. Выбирается четвертый вариант с максимальной суммой весов, равной 2. Результатом выполнения этого шага является трехмерный гиперкуб, изображенный на рис. 6. Во втором гиперкубе, также изображенном на рис. 6, одна из гиперграней полностью состоит из вершин, соответствующих фиктивным состояниям, и для сборки этого гиперкуба не надо перебирать ребра.

Таблица 5

Варианты присоединения ребер на третьем шаге

Ребра	Σw_{ij}	Ребра	Σw_{ij}	Ребра	Σw_{ij}
$q_1q_2, q_6q_9, q_3q_4, q_7q_8$	0	$q_1q_5, q_6q_{11}, q_7q_{12}, q_3q_{10}$	0	$q_2q_5, q_9q_{11}, q_8q_{12}, q_4q_{10}$	0
$q_1q_9, q_2q_6, q_3q_8, q_4q_7$	1	$q_1q_{11}, q_5q_6, q_7q_{10}, q_3q_{12}$	0	$q_2q_{11}, q_5q_9, q_8q_{10}, q_4q_{12}$	1
$q_1q_4, q_6q_8, q_7q_9, q_2q_3$	1	$q_1q_{10}, q_6q_{12}, q_7q_{11}, q_3q_5$	0	$q_2q_{10}, q_9q_{12}, q_8q_{11}, q_4q_5$	1
$q_1q_8, q_4q_6, q_2q_7, q_3q_9$	2	$q_1q_{12}, q_6q_{10}, q_5q_7, q_3q_{11}$	1	$q_2q_{12}, q_9q_{10}, q_5q_8, q_4q_{11}$	0

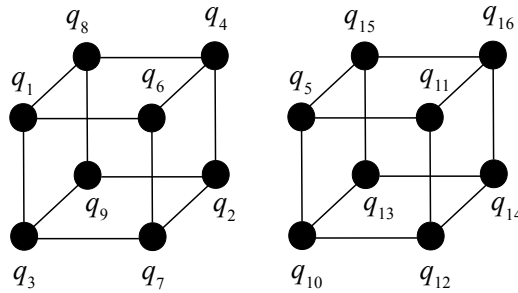


Рис. 6. Результат формирования трехмерных гиперкубов

Перебор вариантов соединения полученных трехмерных гиперкубов для получения окончательного решения в виде четырехмерного гиперкуба представлен в табл. 6. На последнем шаге качество сборки удобно оценивать по дефекту отображения ребер D , значения которого показаны в последней строке табл. 6. Наименьшее значение $D = 6$ получено в результате применения оператора E_2 .

Таблица 6

Варианты последнего шага сборки гиперкуба

Код Грея	Исходное положение		Применение операторов E_i						
			E_1	E_2	E_3	E_1E_2	E_1E_3	E_2E_3	$E_1E_2E_3$
000	q_1	q_5	q_{15}	q_{11}	q_{10}	q_{16}	q_{13}	q_{12}	q_{14}
001	q_3	q_{10}	q_{13}	q_{12}	q_5	q_{14}	q_{15}	q_{11}	q_{16}
011	q_7	q_{12}	q_{14}	q_{10}	q_{11}	q_{13}	q_{16}	q_5	q_{15}
010	q_6	q_{11}	q_{16}	q_5	q_{12}	q_{15}	q_{14}	q_{10}	q_{13}
110	q_4	q_{16}	q_{11}	q_{15}	q_{14}	q_5	q_{12}	q_{13}	q_{10}
111	q_2	q_{14}	q_{12}	q_{13}	q_{16}	q_{10}	q_{11}	q_{15}	q_5
101	q_9	q_{13}	q_{10}	q_{14}	q_{15}	q_{12}	q_5	q_{16}	q_{11}
100	q_8	q_{15}	q_5	q_{16}	q_{13}	q_{11}	q_{10}	q_{14}	q_{12}
D	10	9	6	8	8	8	8	8	8

Окончательное решение в виде четырехмерного гиперкуба получаем, соединив ребрами $q_1q_{11}, q_2q_{13}, q_3q_{12}, q_4q_{15}, q_5q_6, q_7q_{10}, q_8q_{16}$ и q_9q_{14} трехмерные гиперкубы, показанные на рис. 6. Это решение приведено на рис. 7. Коды состояний автомата можно получить по табл. 6. Для этого к кодам Грея, соответствующим состояниям из второго столбца, надо приписать слева 0, а к кодам Грея, соответствующим состояниям из столбца E_2 , приписать слева 1. Тогда получим следующее кодирование для 11 состояний: $q_1 - 0000$, $q_2 - 0111$, $q_3 - 0001$, $q_4 - 0110$, $q_5 - 1001$, $q_6 - 0010$, $q_7 - 0011$, $q_8 - 0100$, $q_9 - 0101$, $q_{10} - 1011$, $q_{11} - 1000$.

Заметим, что дефект отображения D , характеризующий качество решения, здесь оказался равным дефекту отображения, полученному для того же примера в [6]. Если

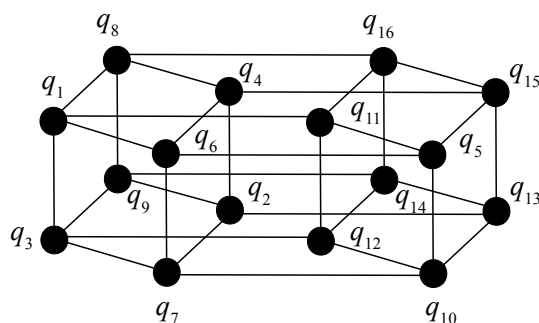


Рис. 7. Результат сборки четырехмерного гиперкуба

взять способ построения гиперкуба из работы [6], при котором выбирается вариант соединения четырьмя ребрами грани $q_5q_{11}q_{12}q_{10}$ с одной из граней куба с вершинами $q_1, q_2, q_3, q_4, q_6, q_7, q_8$ и q_9 , то удастся получить $D = 4$, т. е. на 2 меньше, чем в [6], но этот способ намного более трудоемкий.

Заключение

Предлагаемый метод кодирования состояний автомата рассчитан на использование его в автоматизированной системе логического проектирования. Минимизация интенсивности переключений не противоречит минимизации количества элементов в схеме. Предлагаемый метод допускает частичное совместное решение этих двух задач на этапе кодирования состояний. Решение примеров показывает, что предлагаемый подход не хуже известных и может быть использован для энергосберегающего кодирования состояний автомата.

Автор выражает благодарность рецензенту за ценные замечания, позволившие устранить ошибки и повысить качество изложения.

ЛИТЕРАТУРА

1. *Мурога С.* Системное проектирование сверхбольших интегральных схем. В 2 кн. Кн. 1. М.: Мир, 1985. 288 с.
2. *Pedram M.* Power minimization in IC design: Principles and applications // ACM Trans. Design Automat. Electron. Syst. 1996. V. 1. P. 3–56.
3. *Kashirova L., Keevallik A., and Meshkov M.* State assignment of finite state machine for decrease of power dissipation // Second Inter. Conf. Computer-Aided Design Discrete Devices — CAD DD'97. Minsk, Republic of Belarus, November 12–14, 1997. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 1997. P. 60–67.
4. *Sudnitson A.* Partition search for FSM low power synthesis // Fourth Inter. Conf. Computer-Aided Design of Discrete Devices — CAD DD'2001. Minsk, Republic of Belarus, November 14–16, 2001. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 2001. P. 44–49.
5. *Закревский А. Д.* Об оптимальном размещении графа в булевом пространстве // Вестник Томского государственного университета. Приложение. 2005. № 14. С. 13–17.
6. *Закревский А. Д.* Алгоритмы энергосберегающего кодирования состояний автомата // Информатика. 2011. № 1(29). С. 68–78.
7. *Armstrong D. B.* A programmed algorithm for assigning internal codes for sequential machines // IRE Trans., EC-11. 1962. No. 4. P. 466–472.

8. *Armstrong D. B.* On the efficient assignment of internal codes to sequential machines // IRE Trans, EC-11. 1962. No. 5. P. 611–622.
9. *Оранов А. М.* Размещение множества вершин взвешенного графа в простой цепи // Логическое проектирование дискретных устройств. Минск: Ин-т техн. кибернетики АН БССР, 1984. С. 54–61.
10. *Закревский А. Д., Поттосин Ю. В., Черемисинова Л. Д.* Логические основы проектирования дискретных устройств. М.: Физматлит, 2007. 592 с.
11. *Pottosin Yu. V.* «Assembling» a Boolean hypercube: an approach to state assignment of finite state machines // Second Inter. Conf. Computer-Aided Design of Discrete Devices — CAD DD'97. Minsk, Republic of Belarus, November 12–14, 1997. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 1997. P. 54–59.
12. *Macii E., Pedram M., and Somenzi F.* High-level power modeling, estimation and optimization // IEEE Trans. Computer-Aided Design Integrated Circuits and Systems. 1998. V. 17. No. 11. P. 1061–1079.