

МНОГОКРИТЕРИАЛЬНЫЙ АЛГОРИТМ ОТОБРАЖЕНИЯ СХЕМ ЦИФРОВЫХ УСТРОЙСТВ НА ГРАФ ПРОЦЕССОРОВ ПРИ РАСПРЕДЕЛЕННОМ ЛОГИЧЕСКОМ МОДЕЛИРОВАНИИ

Ладыженский Ю.В., Попов Ю.В.

Кафедра ПМИИ, ДонНТУ

ly@cs.dgtu.donetsk.ua

Abstract

Ladyzhensky Y.V., Popoff Y.V. Multi-criterion digital device circuits to a graph of processors mapping algorithm for distributed logic simulation. A multi-criterion digital device circuits partitioning algorithm is developed. Optimization criteria include a total number of communication channels, loading disbalance, coherence degree, total number and size of cycles in a graph of processors. Experimental research of the algorithm is fulfilled.

Введение

Верификация проектов – это важный этап в разработке цифровых электронных устройств. Эффективным способом проверки корректности работы цифрового устройства является моделирование его работы. Моделирование больших проектов требует больших вычислительных ресурсов. Ускорение процесса моделирования достигается использованием программных и аппаратных ускорителей. В аппаратных ускорителях отдельные части схемы либо вся схема моделируется на аппаратном уровне. При программном ускорении схема разрезается на части, и каждая из частей моделируется на отдельном процессоре. Эффективность распределенного логического моделирования в значительной мере зависит от способа отображения схемы на граф процессоров.

Анализ последних исследований показал, что существуют алгоритмы разрезания, использующие различные критерии оптимальности. Один из наиболее известных алгоритмов разрезания – эвристика локального поиска Кернигана и Лина [1]. В [2, 3, 4] рассматриваются вариации этого алгоритма, представлены методы исследования цифровых устройств с программируемой логикой средствами моделирования. Другие подходы к разрезанию схем представлены в [5, 6, 7]. В [8, 9] отмечено, что для повышения эффективности моделирования, при разрезании схемы требуется учитывать равномерность загрузки процессоров (отсутствие дисбаланса нагрузки) и минимизировать число каналов связи. Однако в этих источниках не рассматривается, как именно влияет соотношение дисбаланса нагрузки и числа каналов связи на эффективность моделирования. В [10] показано, что при моделировании одной и той же схемы, разрезанной разными способами, время моделирования схемы сильно изменяется. Установлено, что на время моделирования влияет не только общее число каналов связи и дисбаланс нагрузки, но и степень связности графа процессоров (количество дуг в графе процессоров, при этом одной дуге может соответствовать несколько каналов связи), наличие в графе процессоров циклов, размер этих циклов. Программная система для исследования методов параллельного логического моделирования рассмотрена в [10, 11].

Целью данной статьи является разработка и исследование свойств многокритериального алгоритма разрезания схем на основе классификации критериев оптимальности разрезания.

1. Постановка задачи разрезания схемы на части

Исходная схема – это множество элементов $E = \{e_1, e_2, \dots, e_n\}$, соединенных цепями из множества $V = \{(i, j) \mid \text{есть электрическая цепь } i \rightarrow j\}$. Схему представим в виде графа элементов. Вершина соответствует элементу схемы, дуга соответствует электрической цепи в схеме. Пример схемы и соответствующего ей графа элементов приведен на рис. 1.

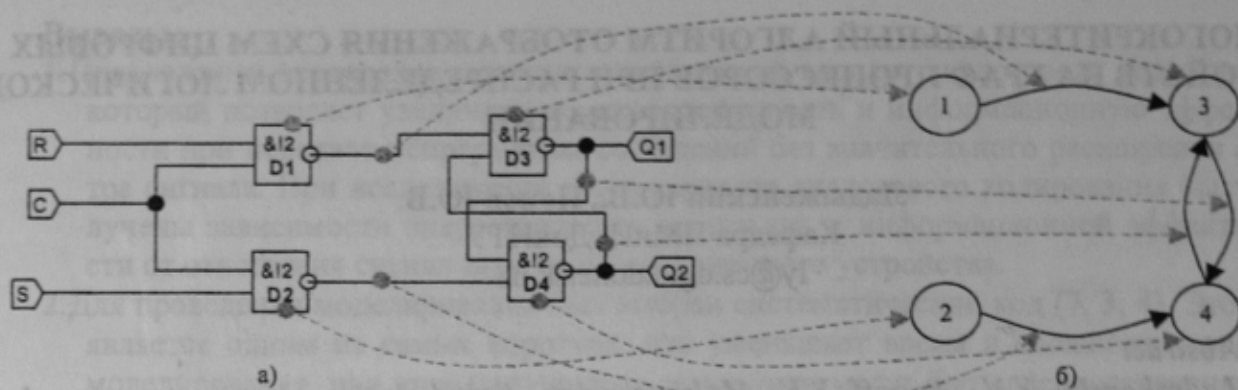


Рисунок 1 – Пример а) схемы и б) графа элементов

В результате разрезания схема отображается на граф процессоров. Граф процессоров задается множеством процессоров $P = \{p_1, p_2, \dots, p_k\}$ и множеством каналов связи $C = \{(i, j) | \text{есть канал связи от } p_i \text{ к } p_j\}$. Списки элементов задаются множеством $M = \{M_1, M_2, \dots, M_k\}$, где M_i - множество элементов на процессоре p_i . Каждому каналу связи соответствует список электрических цепей $C_{i,j} = \{(k, l) | (k, l) \in V, e_k \in M_k, e_l \in M_l\}$. Мощность графа процессоров $|P|$ задана.

Граф процессоров – взвешенный по вершинам и по дугам.

Вес вершины r_i определяется суммарным весом элементов, которые вошли в эту вершину. Для графа процессоров, в котором каждому процессору соответствует ровно один элемент, вес каждой вершины равен весу этого элемента. Вес элемента является относительной оценкой временной сложности элемента. Временная сложность элемента – это количество физического времени, которое требуется для расчета сигналов на выходах элемента при моделировании. Для разрезания имеет значение только соотношение весов элементов.

Вес дуги в графе процессоров $s_{i,j}$ равен количеству разных электрических цепей схемы, которым соответствует эта дуга: $s_{i,j} = |C_{i,j}|$. Для графа на рис. 1 вес всех дуг равен 1, потому что каждая дуга соответствует ровно одной электрической цепи.

Пример разрезанной схемы и соответствующего ей графа процессоров приведен на рис. 2. На рис. 2 а) схема отображена на два процессора P1 и P2. Соответствующий граф процессоров приведен на рис. 2 б). Вес вершин равен 2, потому что на каждый из процессоров отображено по два элемента. Вес дуги равен 2, потому что элементы, отображенные на первый процессор, соединяются с элементами на втором процессоре двумя электрическими цепями. Дуга направлена от выходов элементов первого процессора ко входам элементов второго процессора.

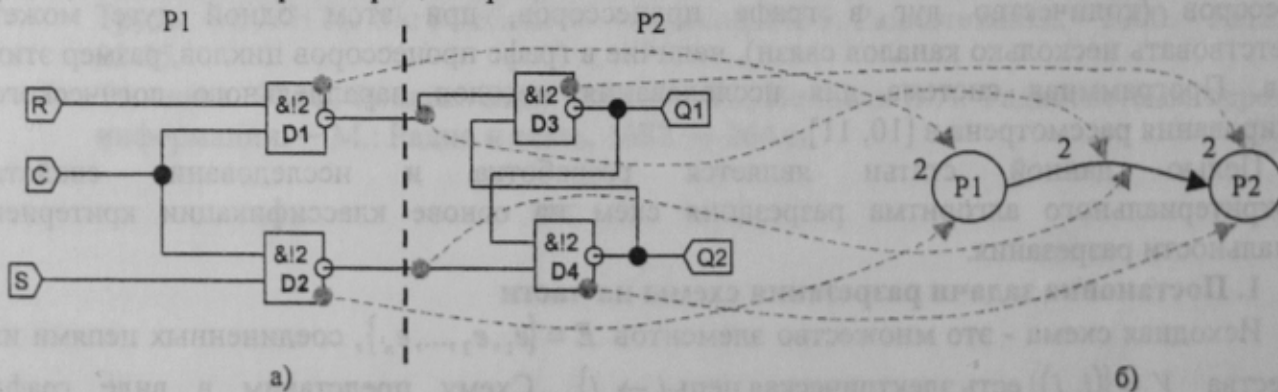


Рисунок 2 – Взвешенный граф процессоров:

а) схема, отображенная на два процессора б) граф процессоров

2. Операция стягивания двух вершин

Стягиванием пары вершин i и j будем называть операцию, в которой: 1) все дуги, соединяющие две стягиваемые вершины, удаляются; 2) все дуги, входящие во вторую вершину, перенаправляются в первую; 3) все дуги, исходящие из второй вершины, перенаправляются из первой; 4) вес первой вершины пересчитывается и становится равным суммарному весу стянутых вершин (рис. 3); 5) вес дуг пересчитывается и становится равным числу разных электрических цепей, соответствующих этим дугам (рис. 3); 6) вторая вершина удаляется.

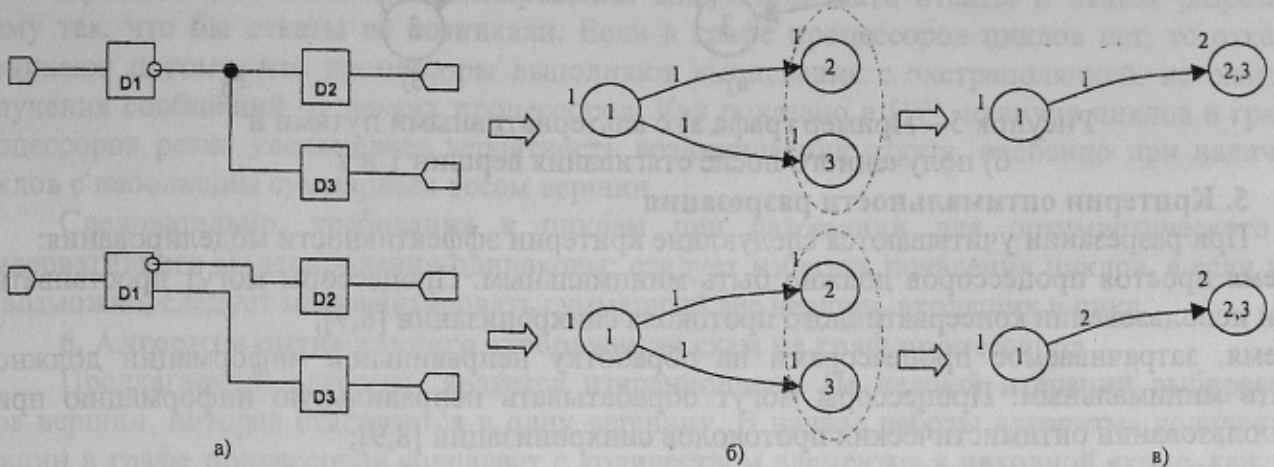


Рисунок 3 – Пересчет весов вершин и дуг после стягивания двух вершин:

а) исходная схема, б) граф процессоров, в) граф после стягивания двух вершин

3. Расстояние между вершинами

Под расстоянием между двумя элементами будем понимать физическое время моделирования, которое требуется для того, что бы изменение сигналов, произошедшее на входах первого элемента привело к изменениям на выходах второго элемента.

Стоимость пути между парой вершин равна суммарному весу всех вершин, входящих в путь, включая исходную и конечную вершины. Для отображения имеют значение только кратчайшие пути, поскольку первые изменения на выходах последнего элемента произойдут через минимальное время. Пусть, $T = \{t_{i,j}\}$ - матрица кратчайших путей между всеми парами вершин. Пример графа и матрицы кратчайших путей T приведен на рис. 4.

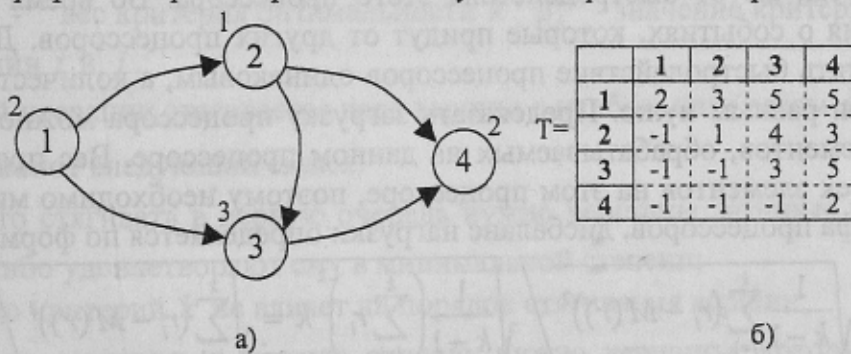


Рисунок 4 -- а) граф и б) матрица кратчайших путей

4. Альтернативные пути

Рассмотрим граф, в котором есть путь, содержащий в себе более чем две (начальная и конечная) вершины. На рис. 5, а от вершины 1 к вершине 2 есть путь через вершину 3. Стягивание вершин, между которыми есть такой путь, приведет к образованию нового цикла.

Альтернативным будем называть путь, в котором, кроме начальной и конечной, содержатся другие вершины. Альтернативным такой путь называется потому, что если между двумя заданными вершинами есть дуга, то альтернативный путь не будет

кратчайшим. Матрицу наличия альтернативных путей обозначим $H = \{h_{i,j} \mid \text{истина, если есть альтернативный путь } i \rightarrow j; \text{ ложь иначе}\}.$

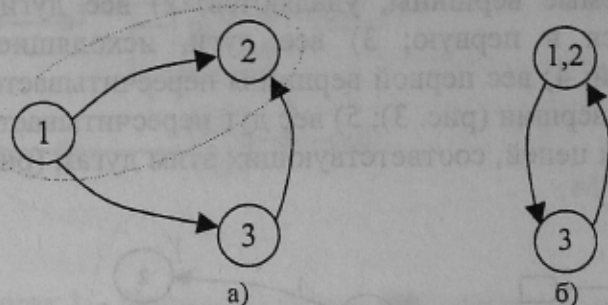


Рисунок 5 – Пример графа а) с альтернативными путями и б) полученного после стягивания вершин 1 и 2

5. Критерии оптимальности разрезания

При разрезании учитываются следующие критерии эффективности моделирования:

- 1) время простоя процессоров должно быть минимальным. Процессоры могут простаивать при использовании консервативного протокола синхронизации [8,9];
- 2) время, затрачиваемое процессорами на обработку неправильной информации должно быть минимальным. Процессоры могут обрабатывать неправильную информацию при использовании оптимистических протоколов синхронизации [8,9];
- 3) суммарный объем сообщений, передаваемых по сети, должен быть минимальным.

Для минимизации времени простоя при консервативном моделировании и времени, потраченного на неправильные вычисления при оптимистическом моделировании, необходимо, чтобы процессоры были равномерно загружены вычислениями. Если при моделировании один из процессоров загружен намного больше, чем другой, то при консервативном моделировании менее загруженный процессор будет ждать событий от более загруженного процессора. При оптимистическом моделировании менее загруженный процессор будет производить больше неправильных вычислений, а так же передавать по сети больше неправильных сообщений.

Время, затрачиваемое процессором на обработку всех событий, зависит от входного воздействия, от событий, полученных от других процессоров, от схемы, моделируемой на этом процессоре, а так же от быстродействия этого процессора. Во время разрезания не известна информация о событиях, которые придут от других процессоров. Для упрощения расчетов будем считать быстродействие процессоров одинаковым, а количество событий во входном воздействии равным нулю. Предсказать загрузку процессора можно на основании суммарного веса элементов, обрабатываемых на данном процессоре. Вес процессора равен суммарному весу всех элементов на этом процессоре, поэтому необходимо минимизировать дисбаланс весов графа процессоров, дисбаланс нагрузки определяется по формуле:

$$\text{Дисбаланс}(r) = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (r_i - M(r))^2} / \sqrt{\frac{1}{k-1} \left(\sum_{i=1}^k r_i\right)^2} k = \sqrt{\sum_{i=1}^k (r_i - M(r))^2} / \left(\sum_{i=1}^k r_i\right)^2, \quad (1)$$

где $M(r)$ - средний вес процессоров; $\frac{1}{k-1} \sum_{i=1}^k (r_i - M(r))^2$ - дисперсия весов процессоров; $\frac{1}{k-1} \left(\sum_{i=1}^k r_i\right)^2 k$ - наибольшее значение дисперсии. Наибольшая дисперсия весов

процессоров получается, если все элементы находятся на одном процессоре. $\text{Дисбаланс}(r)$ изменяется от 0 в наилучшем случае (когда веса всех процессоров равны между собой) до 1 в наихудшем случае (когда все элементы находятся на одном процессоре).

При консервативном моделировании процессоры могут простаивать в результате ожидания выхода из тупика [8,9]. Для минимизации времени такого простоя, необходимо либо разрезать схему так, что бы тупиковые ситуации не возникали, либо что бы их количество было минимальным. Тупики не возникают, если в графе процессоров нет циклов. Процессоры, входящие в циклы с большим суммарным весом обрабатывают, больше событий перед переходом в состояние тупика, чем процессоры, входящие в цикл с меньшим суммарным весом.

При оптимистическом моделировании могут возникать откаты и нельзя разрезать схему так, что бы откаты не возникали. Если в графе процессоров циклов нет, то откаты возникают потому, что процессоры выполняют вычисления с экстраполяцией, не ожидая получения сообщений от других процессоров. Как показано в [10], наличие циклов в графе процессоров резко увеличивает вероятность возникновения отката, особенно при наличии циклов с небольшим суммарным весом вершин.

Следовательно, требования к циклам при разрезании для оптимистического и консервативного моделирования одинаковы: следует избегать появления циклов, а если это не возможно, следует максимизировать суммарный вес вершин, входящих в цикл.

6. Алгоритм оптимального отображения схем на граф процессоров

Предлагаемый алгоритм является итерационным. На каждой итерации выбирается пара вершин, которая стягивается в одну вершину. В начале работы алгоритма количество вершин в графе процессоров совпадает с количеством элементов в исходной схеме, каждой вершине графа процессоров соответствует один элемент исходной схемы, каждой дуге в графе процессоров соответствует электрическая цепь в исходной схеме: $P := E, C := V, M_i := \{e_i\}$. После стягивания очередной пары вершин мощность множества вершин в графе процессоров уменьшается на единицу. Процесс стягивания вершин продолжается до тех пор, пока мощность множества вершин в графе процессоров не станет равной заданному числу процессоров.

Оптимальность разрезания зависит от порядка стягивания вершин. Для каждой пары вычисляется вес $w_{i,j}$:

$$w_{i,j} = \sum_k \alpha_k \beta_k^{i,j}$$

где α_k - вес критерия оптимальности k ; $\beta_k^{i,j}$ - значение критерия оптимальности k для пары вершин i и j .

На каждой итерации стягивается пара вершин с наибольшим весом $w_{i,j}$.

Веса α_k имеют следующий смысл:

- если $\alpha_k < 0$, то стягивать в первую очередь нужно вершины, которые не удовлетворяют критерию k либо удовлетворяют ему в минимальной степени;
- если $\alpha_k = 0$, то критерий k не влияет на порядок стягивания вершин;
- если $\alpha_k > 0$, то стягивать в первую очередь нужно вершины, которые удовлетворяют критерию k в максимальной степени;
- если $|\alpha_k| > |\alpha_l|$, то критерий k важнее критерия l .

Рассмотрим критерии оптимальности для выбора пары вершин:

- 1) $\beta_1^{i,j}$ - сумма весов пары вершин: $\beta_1^{i,j} = r_i + r_j$. Для минимизации дисбаланса нагрузки следует в первую очередь стягивать вершины с минимальным весом, поэтому α_1 должно быть отрицательным. На рис. 6 выбор пары вершин (1,3) (рис. 6,а) оказался лучше выбора пары вершин (1,2) (рис. 6,б), потому что в первом случае $Дисбаланс(r) = 0$, а во втором $Дисбаланс(r) \approx 0.52$;

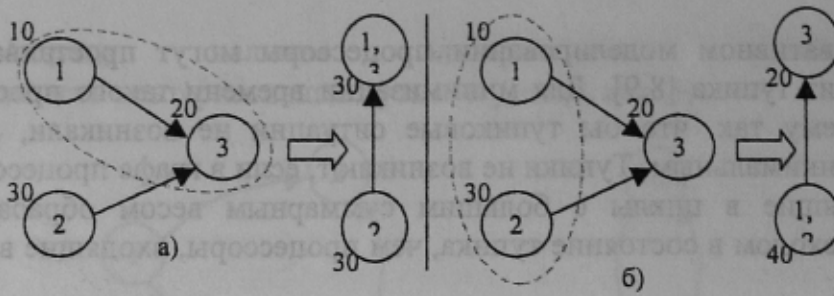


Рисунок 6 – Балансировка нагрузки: для а) дисбаланс нагрузки меньше, чем для б)

2) $\beta_2^{i,j}$ - сумма весов соединяющих дуг: $\beta_2^{i,j} = s_{i,j} + s_{j,i}$. Дуг может не быть, может быть дуга в одном направлении или две дуги в обоих направлениях. Что бы после разрезания получилось как можно меньше каналов связей, нужно склеивать в первую очередь вершины, которые соединяются дугами максимальной стоимости. Для этого α_2 должно быть положительным;

3) $\beta_3^{i,j}$ - признак отсутствия пути между парой вершин: $\beta_3^{i,j} = \begin{cases} 1, t_{i,j} = \infty \& t_{j,i} = \infty \\ 0, \text{ иначе} \end{cases}$. Если вершины взаимно не достижимы, то части схемы, соответствующие этим вершинам, работают независимо друг от друга. Независимые вершины нужно стягивать в последнюю очередь, поэтому α_3 должно быть отрицательным;

4) $\beta_4^{i,j}$ - признак наличия путей в обоих направлениях: $\beta_4^{i,j} = \begin{cases} 1, t_{i,j} < \infty \& t_{j,i} < \infty \\ 0, \text{ иначе} \end{cases}$. Вершины, которые входят в одну сильную компоненту, следует стягивать в первую очередь, поэтому α_4 должно быть положительным;

5) $\beta_5^{i,j}$ - признак наличия альтернативного пути: $\beta_5^{i,j} = \begin{cases} 1, h_{i,j} \vee h_{j,i} \\ 0, \text{ иначе} \end{cases}$. $h_{i,j}$ - признак наличия альтернативного пути от вершины i к вершине j . Если между парой вершин существует альтернативный путь, то стягивание этих вершин приведет к появлению циклов. Наличие циклов нежелательно, поэтому α_5 должно быть отрицательным;

6) $\beta_6^{i,j}$ - стоимость кратчайшего пути между парой вершин в случае, если вершины не принадлежат одной сильной компоненте и одна из вершин достижима из другой:

$$\beta_6^{i,j} = \begin{cases} t_{i,j}, t_{i,j} < \infty \& t_{j,i} = \infty \\ t_{j,i}, t_{j,i} < \infty \& t_{i,j} = \infty \\ 0, \text{ иначе} \end{cases}$$

Чем больше расстояние между процессорами, тем меньше вычисления на одном процессоре зависят от вычислений на другом процессоре. В первую очередь следует стягивать вершины, которые находятся ближе всего, поэтому α_6 должно быть отрицательным.

7) $\beta_7^{i,j}$ - стоимость кратчайшего пути для вершин из одной сильной компоненты:

$$\beta_7^{i,j} = \begin{cases} \min(t_{i,j}, t_{j,i}), t_{i,j} < \infty \& t_{j,i} < \infty \\ 0, \text{ иначе} \end{cases}$$

Смысл $\beta_7^{i,j}$ такой же, как и $\beta_6^{i,j}$, но его добавление позволяет задавать разную стоимость единицы кратчайшего пути для вершин, входящих или не входящих в одну сильную

компоненту. Сначала следует стягивать те вершины из одной сильной компоненты, расстояние между которыми минимально, поэтому a_7 должно быть отрицательным;

8) $\beta_8^{i,j}$ - стоимость кратчайшего цикла, в который входит заданная пара вершин:

$$\beta_8^{i,j} = \begin{cases} t_{i,j} + t_{j,i}, t_{i,j} < \infty \ \& \ t_{j,i} < \infty \\ 0, \text{ иначе} \end{cases}$$

Если нельзя избежать циклов в графе процессоров, то следует стремиться к максимизации суммарного веса вершин, входящих в цикл. В первую очередь следует стягивать вершины, которые входят в самые легкие циклы, поэтому α_8 должно быть отрицательным (рис. 7).

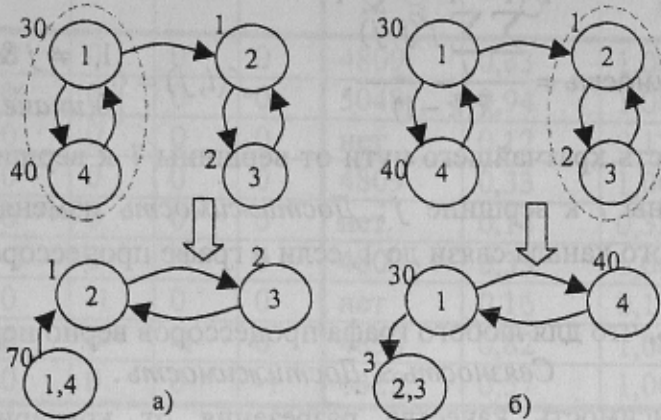


Рисунок 7 – Минимизация числа легких циклов:

для а) суммарный вес вершин, входящих в цикл, меньше, чем для б)

7. Экспериментальное исследование алгоритма

Предложенный алгоритм учитывает комбинации нескольких критериев оптимальности разрезания схем.

Все эксперименты проводились для схемы s38417 из набора схем [12]. В этой схеме содержится 23815 вершин. Уменьшение размерности перед применением алгоритма оптимального разрезания производилось путем стягивания соседних вершин и вершин, входящих в сильные компоненты.

Зависимость времени работы алгоритма разрезания от количества узлов в графе процессоров представлена на рис. 8.



Рисунок 8 – Зависимость времени работы от количества узлов в графе процессоров

На каждой итерации алгоритма требуется определение кратчайших путей между всеми парами вершин. В реализации алгоритма для этого используется алгоритм Флойда [13]. Сложность алгоритма Флойда, как известно, составляет $O(n^3)$. Всего на третьем этапе выполняется $n-k$ шагов, где n – количество вершин в графе процессоров после выполнения первых двух этапов, k – количество процессоров в результате, $n \gg k$. Таким образом, сложность алгоритма составляет $O(n^4)$.

Для оценки качества разрезания введем понятие степени связности и степени достижимости графа процессоров. Степень связности графа процессоров:

$$\text{Связность} = \frac{|C|}{k(k-1)}$$

где C - множество дуг (каналов связи) в графе процессоров; $|C|$ - общее число дуг в графе процессоров; $k(k-1)$ - максимальное количество дуг в графе процессоров; *Связность* изменяется от 0, если в графе процессоров нет ни одного канала связи до 1, если в графе процессоров есть канал связи от каждого процессора к каждому.

Степень достижимости графа процессоров:

$$\text{Достижимость} = \frac{\sum_i \sum_j f(i, j)}{k(k-1)}, \quad f(i, j) = \begin{cases} 1, i \neq j \ \& \ t_{i,j} < \infty \\ 0, \text{иначе} \end{cases}$$

где $t_{i,j}$ - стоимость кратчайшего пути от вершины i к вершине j ; $f(i, j)$ - признак наличия пути от вершины i к вершине j ; *Достижимость* изменяется от 0, если в графе процессоров нет ни одного канала связи до 1, если в графе процессоров есть путь от каждого процессора к каждому.

Следует отметить, что для любого графа процессоров верно неравенство:

$$\text{Связность} \leq \text{Достижимость}.$$

Рассмотрим зависимость качества разрезания от критериев разрезания. Перед применением алгоритма оптимального разрезания размерность схемы s38417 из набора схем [12] была уменьшена до 351 вершин путем стягивания соседних вершин. Зависимость качества разрезания от веса критериев оптимальности разрезания при учете только одного критерия оптимальности представлена в табл. 1.

Кратчайший цикл – наименьшая суммарная стоимость процессоров, которые входят в один цикл. Дисбаланс нагрузки определяется по формуле (1). Количество цепей – общее количество электрических цепей, которые соединяют выходы элементов на одних процессорах со входами элементов на других процессорах. Будем называть эти цепи разрезанными электрическими цепями (р-цепи). Эта величина может служить мерой количества передаваемых по сети сообщений о событиях.

Первая строка в табл. 1 показывает результат разрезания, когда вершины стягиваются в таком порядке, как они записаны в файле описания схемы. Строки, в которых $\alpha_3 = -1$ (нет пути между парой вершин) и $\alpha_3 = -1$ (есть альтернативный путь между парой вершин) совпали с первой строкой потому, что соответствующие критерии выполняют предотвращающую функцию: не стягивать независимые вершины и вершины, стягивание которых может привести к образованию новых циклов. Эти критерии влияют на результат только в комбинации с другими критериями.

Зависимость количества разрезанных электрических цепей и дисбаланса нагрузки от значения α_2 (стоимость дуг между стягиваемыми вершинами) при $\alpha_1 = -1$ представлена на рис. 9. При увеличении α_2 уменьшается число разрезанных электрических цепей, но при этом увеличивается дисбаланс нагрузки.

Зависимость степени связности графа и дисбаланса нагрузки от значения α_4 (входят ли стягиваемые вершины в одну сильную компоненту) при $\alpha_1 = -1$ представлена на рис. 10. При увеличении α_4 уменьшается степень связности графа процессоров, но при этом увеличивается дисбаланс нагрузки. При $\alpha_4 = 0 \dots 10000$ суммарное число разрезанных электрических цепей (не показано на рис. 10) оставалось практически неизменным в

пределах от 1700 до 1900, при $\alpha_4=20000$ оно уменьшилось до 1194, а при $\alpha_4=30000$ оно уменьшилось до 54.

Таблица 1 – Зависимость качества разрезания от веса критериев оптимальности

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	Кратчайший цикл	Связность	Достижимость	Дисбаланс	Количество р-цепей
0	0	0	0	0	0	0	0	4809	0,33	1,00	0,56	757
-1	0	0	0	0	0	0	0	5048	0,94	1,00	0,05	1886
0	1	0	0	0	0	0	0	нет	0,12	0,12	0,87	7
0	0	-1	0	0	0	0	0	4809	0,33	1,00	0,56	757
0	0	0	1	0	0	0	0	нет	0,14	0,35	0,87	19
0	0	0	0	-1	0	0	0	4809	0,33	1,00	0,56	757
0	0	0	0	0	-1	0	0	нет	0,16	0,17	0,86	36
0	0	0	0	0	0	-1	0	4898	0,82	1,00	0,06	1330
0	0	0	0	0	0	0	-1	6625	0,57	1,00	0,20	1122

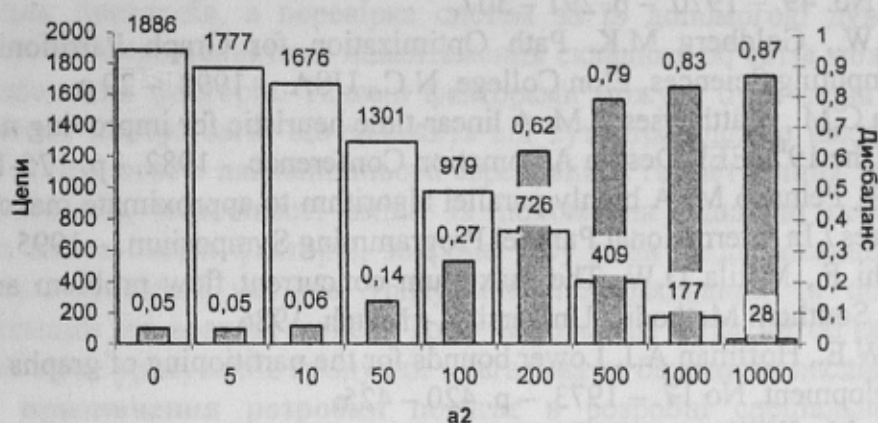


Рисунок 9 – Зависимость числа разрезанных электрических цепей и дисбаланса нагрузки от значения α_2 при $\alpha_1=-1$

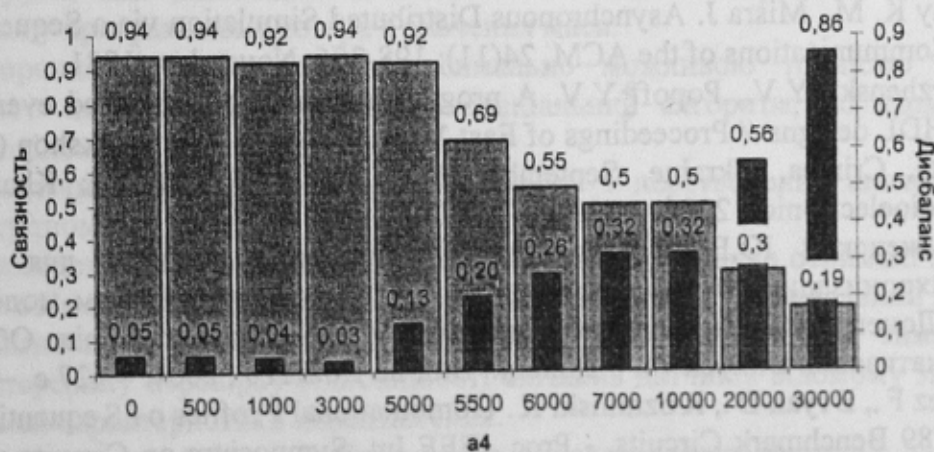


Рисунок 10 – Зависимость степени связность графа процессоров и дисбаланса нагрузки от значения α_4 при $\alpha_1=-1$.

Выводы

Сформулированы основные критерии оптимальности разрезания. Предложен многокритериальный алгоритм для разрезания схем на части. Предложены критерии для оценки оптимальности результата разрезания. В дополнение к классическим критериям оптимальности, включающим в себя одинаковость размера каждой части схемы и минимизацию числа каналов связи, предложено ввести новые критерии оптимальности, учитывающие наличие и некоторые свойства циклов в графе процессоров. Предложенный алгоритм разрезания схемы допускает разницу в размере частей схемы в пользу уменьшения числа каналов связи и в пользу удаления циклов из графа процессоров либо максимизации суммарного веса вершин, входящих в циклы.

Реализация рассмотренного в статье многокритериального алгоритма разрезания схем может быть использована для исследования эффективности работы алгоритмов распределенного логического моделирования при моделировании одной и той же схемы, разрезанной разными способами.

Перспективным направлением дальнейшей работы является исследование особенностей влияния весов критериев оптимальности разрезания на результат разрезания и на эффективность распределенного моделирования схемы.

Литература

1. Kernigan B.W., Lin S. An efficient procedure for partitioning graphs // Bell System Technical Journal, No. 49. – 1970. – p. 291 – 307.
2. Berry J.W., Goldberg M.K. Path Optimization for Graph Partitioning Problems. – Department of Computing Sciences, Elon College, N.C., USA. – 1998. – 29 p.
3. Feduccia C.M., Mattheyses R.M. A linear-time heuristic for improving network partitions / In proceedings of the 19th IEEE Design Automation Conference. - 1982. - p. 175-181.
4. Homer S., Peinado M. A highly parallel algorithm to approximate maxcut on distributed memory architectures / In International Parallel Programming Symposium. – 1995.
5. Shahrokhi F., Matila D.W. The maximum concurrent flow problem and sparsest cuts. Technical report. – Southern Methodist University. – Match, 1986.
6. Donath W.E., Hoffman A.J. Lower bounds for the partitioning of graphs / IBM Journal of Research and Development, No 17. – 1973. – p. 420 – 425.
7. Goemans M., Williamson D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming / Journal of the ACM, submitted.
8. Ferscha Alois. Parallel and Distributed Simulation of Discrete Event Systems. In Hardbound of Parallel and Distributed Computing. McGraw-Hill, 1995.
9. Chandy K. M., Misra J. Asynchronous Distributed Simulation via a Sequence of Parallel Computations. Communications of the ACM, 24(11): 198-206, November, 1981.
10. Ladyzhensky Y.V., Popoff Y.V. A program system for distributed event-driven logic simulation of VHDL designs // Proceedings of East-West Design & Test Workshop (EWDTW'04). – Yalta, Alushta, Crimea, Ukraine, September 23-26, 2004. – Kharkov: Kharkov National University of Radioelectronics, 2004. – 289 p. – p. 203 - 209.
11. Ладыженский Ю.В., Попов Ю.В. Программная система для исследования протоколов синхронизации при распределенном событийном логическом моделировании // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випуск 74. – Донецьк: Вид-во ДонНТУ, 2004. – 427 с. – с. 201 - 209.
12. Brglez F., Bryan D., Kozminski K. Combinational Profiles of Sequential Benchmark Circuits. ISCAS'89 Benchmark Circuits. - Proc. IEEE Int. Symposium on Circuits and Systems. - May 1989. - pp. 1929-1934.
13. Дискретная математика для программистов / Ф.А. Новиков. – СПб: Питер, 2001. – 304 с.: илл.