ПРОЦЕССОР ДЛЯ РЕШЕНИЯ ЗАДАЧИ О МАКСИМАЛЬНОМ НЕЗАВИСИМОМ МНОЖЕСТВЕ ВЕРШИН ГРАФА

Ладыженский Ю.В., Куркчи В.А.
Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики и информатики
E-mail: ly@cs.dgtu.donetsk.ua, azur1982@mail.ru

Abstract.
Ladyzhensky Y., Kourktchi V. The processing unit for the maximal independent set problem. *A new processing unit is provided which search for maximal independent set of the given graph using the polynomial algorithm. A simulation model of processing unit is described. The results of simulation of processing unit are discussed.*

Введение

Задача поиска наибольшего независимого множества вершин задача имеет приложения в инженерии и экономике [1, 2]. Она возникает при планировании параллельных вычислений, в компьютерном зрении, при планировании радиосетей, составлении маршрутов, проектировании БИС и ПЛИС [9].

Пусть задан граф G(V,E), где V={1,2,…,n} – множество вершин графа, $E\subset V\times V$ – множество ребер графа. Требуется найти наибольшее подмножество I вершин, такое что любые две вершины $i,j\in I$ несмежны и добавление любой другой вершины в множество I нарушит условие его независимости [3]. Приближенным решением задачи является любое максимальное независимое множество.

Задача о наибольшем независимом множестве является NP-полной [4]. Это значительно затрудняет ее решение. Если требуется решить данную задачу на графах с несколькими тысячами или десятками тысяч вершин, то при отсутствии полиномиального алгоритма это сделать невозможно. Существуют эвристические [5] и метаэвристические [5, 6] приближенные методы для поиска наибольшего независимого множества. Некоторые из них позволяют быстро получить грубые приближения к решению, другие дают более точные приближения к решению.

В таких алгоритмах, как генетические или поиск табу, для повышения точности находимого решения многократно используется локальный поиск [6]. Поэтому, для повышения производительности методов поиска независимых множеств целесообразно разрабатывать специализированные проблемно-ориентированные процессоры.

Полиномиальный алгоритм

Задачу поиска наибольшего независимого множества можно сформулировать как задачу нелинейной оптимизации целевой функции полинома n-ого порядка [7]:

$$\max_{\bar{x}\in[0,1]^n} F(\bar{x}) = \sum_{i=1}^{n}\left((1-x_i)\prod_{j\in\Gamma(i)} x_j\right),$$

где x_i – переменная, сопоставленная с вершиной i заданного графа. Поставленная задача решается приближенным методом, состоящим из следующих этапов:

1. Произвольно выбирается начальное приближение $X=\{0,1\}^n$.
2. Для j=1,2,…,n: если $\sum_{(j,k)\in E} x_k<1$, тогда $x_j:=1$, иначе $x_j:=0$.
3. Для j=1,2,…,n: если $\sum_{(j,k)\in E} x_k<1$, тогда $x_j:=1$, иначе $x_j:=0$.
4. Максимальное независимое множество $I=\{i\in V|x_i=1\}$.

На шаге 3 достигается независимость множества, а повторение операции аналогичной на шаге 4 обеспечивает его максимальность.

В [8] проведено экспериментальное исследование полиномиального алгоритма в сравнении с другими известными алгоритмами. Показано, что алгоритм позволяет получать решения хорошего качества и прост для аппаратной реализации.

Структура процессора

Структурная схема спецпроцессора представлена на рис. 1. Устройство состоит из четырех блоков: блока управления БУ, матрицы смежности МС, массива независимого множества МНМ и вычислительного блока ВБ.

Блок управления (см. рис. 2) осуществляет выборку текущей строки матрицы смежности заданного графа и текущего элемента массива независимого множества. Генератор тактовых импульсов ГИ, счетчик Сч1 и дешифратор Д обеспечивают единичный сигнал на одном выходе и нулевые сигналы на всех остальных выходах. Фактически эти три элемента выполняют цикл из пп. 2 и 3 алгоритма.

Двухразрядный счетчик Сч2, на вход которого подается выход дешифратора с наибольшим номером, обеспечивает выполнение цикла дважды, после чего на выходе старшего разряда счетчика формируется единичный сигнал, который подается на вход сброса триггера Т, что останавливает работу устройства.

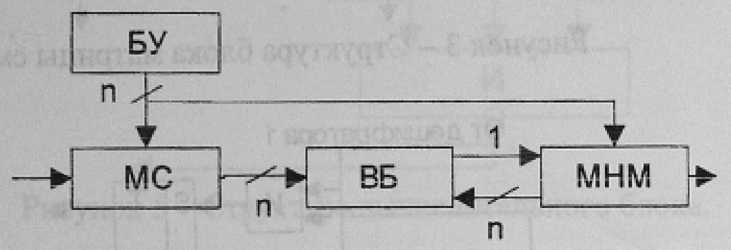

Рисунок 1 – Структура процессора для решения задачи о максимальном независимом множестве

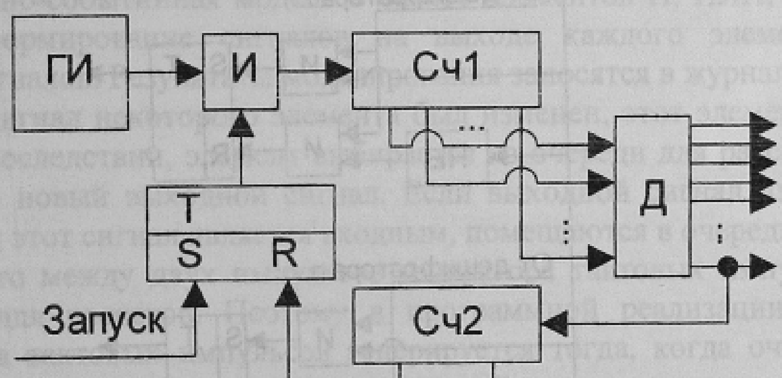

Рисунок 2 – Структура блока управления

Матрица смежности в устройстве (см. рис. 3) построена так, чтобы на каждом такте работы устройства на выходе ее формировались сигналы, соответствующие одной строке матрицы смежности графа [9]. Номер строки на выходе матрицы смежности соответствует номеру выхода блока управления с единичным сигналом.

Массив независимого множества (см. рис. 4) выполнен на RS-триггерах и предназначен для хранения текущего независимого множества и формирования в нем максимального независимого множества. Каждый RS-триггер k этого блока хранит информацию о принадлежности вершины графа с номером k текущему множеству. Данный блок записывает результат вычислений вычислительного блока в триггер, номер которого совпадает с номером единичного сигнала на выходе блока управления.

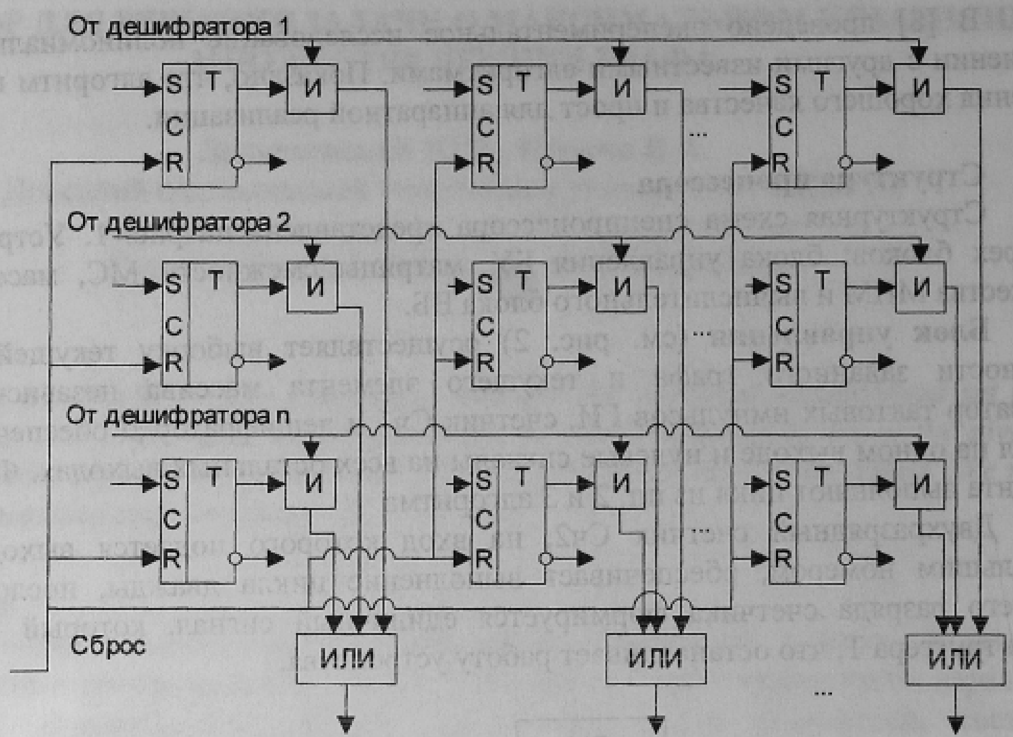

Рисунок 3 – Структура блока матрицы смежности

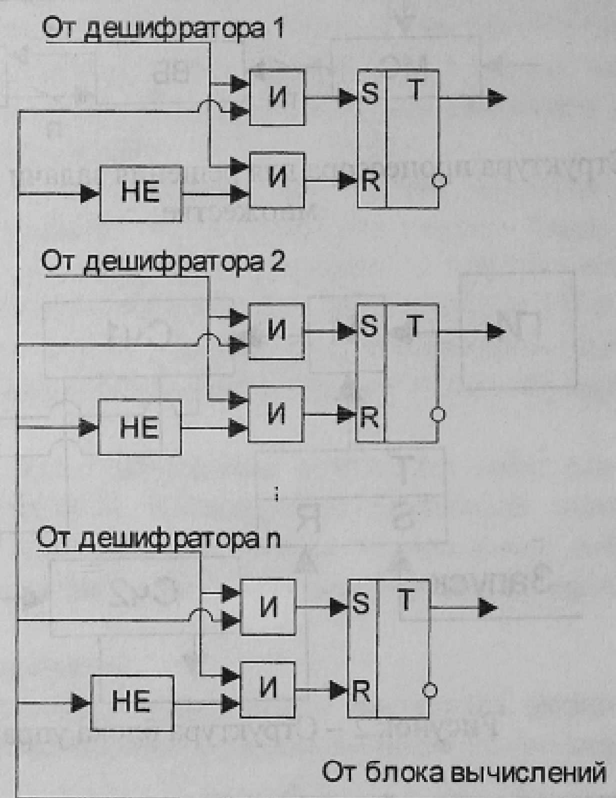

Рисунок 4 – Структура блока массива независимого множества

Блок вычислений на основе текущей строки матрицы смежности и текущего состояния массива независимого множества вычисляет элементы сумм шагов 2 и 3 алгоритма. Если хотя бы один из элементов суммы равен единице, то значение суммы будет больше или равно единице. Результат проверки этого условия формируется в виде единичного или нулевого сигнала на выходе блока вычислений и сохраняется в массиве независимого множества.

Процессор находит максимальное независимое множество за 2n тактов работы.

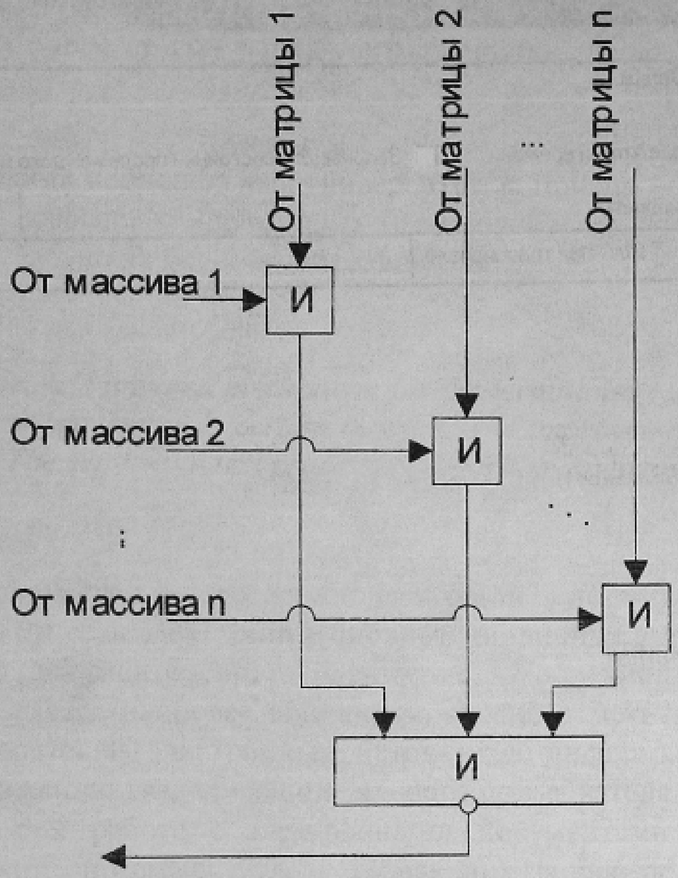

Рисунок 5 – Структура вычислительного блока.

Исследование процессора методом логического моделирования

Для исследования свойств предложенного процессора разработана его программная имитационная дискретно-событийная модель в базисе элементов И, ИЛИ, НЕ, RS-триггер. Модель описывает формирование сигналов на выходе каждого элемента вследствие изменения входных сигналов. Результаты моделирования заносятся в журнал (см. рис. 6).

Если входной сигнал некоторого элемента был изменен, этот элемент помещается в очередь элементов. Впоследствии, элемент выбирается из очереди для рассмотрения. После этого вычисляется его новый выходной сигнал. Если выходной сигнал изменился, то все элементы, для которых этот сигнал является входным, помещаются в очередь.

Мы считаем, что между двух импульсов генератора тактовых импульсов проходит достаточно времени для расчетов. Поэтому в программной реализации модели новый импульс от генератора тактовых импульсов генерируется тогда, когда очередь элементов становиться пустой, то есть никакие изменения состояний уже невозможны.

Проведено моделирование на графах с числом вершин от 5 до 50. Моделирование подтвердило правильность разработанных алгоритмов работы процессора и оценки его быстродействия.

Выводы

Разработан процессор для решения задачи о максимальном независимом множестве вершин графа на основе полиномиального алгоритма. На процессор получен патент Украины на полезную модель [10].

Разработана и программно реализована имитационная модель процессора. Исследования имитационной модели подтвердили работоспособность процессора.

Процессор можно использовать для решения задачи о максимальном независимом множестве в вычислительных системах и сетях, в метаэвристических алгоритмах, основанных на многократном локальном поиске.

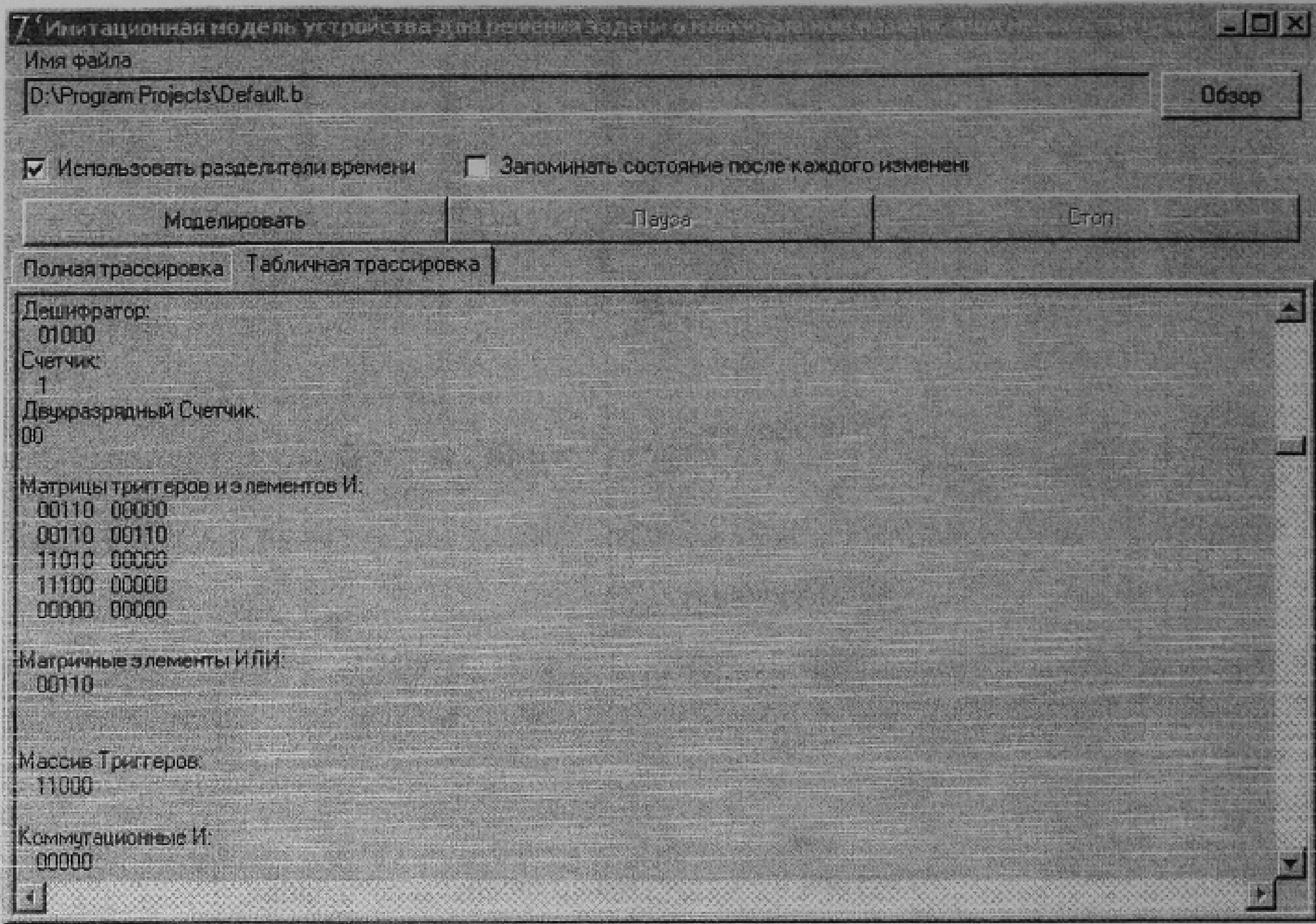

Рисунок 6 – Окно приложения, моделирующего работу процессора

Список использованной литературы

1. Deo N. Graph theory with applications to engineering and computer science. Prentice-Hall, 1974.
2. Avondo-Bodeno G. Economic applications of the theory of graphs. Gordon and Breach Science Publishers, 1962.
3. Berge C. The theory of graphs and its applications. Methuen, 1962.
4. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2000. – 304 с.
5. Bomze I. M., Budinich M., Pardalos P.M., Pelillo M. The maximum clique problem. In D.-Z. Du and Pardalos P.M., editors, Handbook of Combinatorial Optimization, Kluwer Acad. Publishers, 1999. – pp. 1-74.
6. Blum C., Roli A., Metaheuristics in Combinatorial optimization: overview and conceptual comparison. – ACM computing surveys, vol. 35, No 3, September 2003. – pp. 268-308.
7. Abello J., Butenko S., Pardalos P.M. Finding independent set in a graph using continuous multivariable polynomial formulation. AT&T Labs technical report, 2000.
8. Ладыженский Ю.В., Куркчи В.А. Моделирование параллельных алгоритмов для поиска независимых множеств//Моделирование и компьютерная графика: Материалы 1-й международной конференции, г.Донецк, 04-07 октября 2005г. – Донецк, ДонНТУ, МОН Украины, 2005. – сс. 127-133.
9. Авт. св. СССР №1684795, МПК G06G15/20, опубл. 15.10.91г., Бюлл. №38.
10. Декларацiйний патент України на корисну модель №17119, МПК (2006) G06F 17/00, опубл. 15.09.2006, Бюл. №9.