

# ПОДХОД ЗА ХАРДУЕРНО РЕШЕНИЕ НА ПРОБЛЕМА С КОНФЛИКТИТЕ В КОМУТАЦИОННИ ВЪЗЛИ

## AN APPROACH FOR HARDWARE SOLUTION OF THE CONFLICTS PROBLEM IN SWITCHING NODES

К.Колчаков, В. Монов

Институт по информационни и комуникационни технологии Българска Академия на Науките,  
1113, София, България, ул.Акад. Г. Бончев бл.2, тел.: (+359) 2 979 -32 -40 e-mail: [kkolchakov@abv.bg](mailto:kkolchakov@abv.bg)  
[vmonov@iit.bas.bg](mailto:vmonov@iit.bas.bg)

**Abstract:** An attempt for a complete hardware solution of non conflict-requests management in a node of type Crossbar is done. For this purpose algorithms are selected for non-conflict schedule using diagonally activation of the connections matrix. A schematic diagram of a system of finite automata is designed controlling the switching matrix of the node without conflicts.

**Key words:** network nodes, node traffic, crossbar switch, conflict elimination, packet messages.

### УВОД

В изследването е направен опит да се намери изцяло хардуерно решение на проблемът с конфликтите в комутационни възли от тип Кросбар. Основа в този изцяло хардуерен подход е знанието, че диагонално разположените заявки за обслужване в матрицата на връзките са безконфликтни. Естествено е, да се използват крайни автомати, които да управляват заявките безконфликтно, като ги активират диагонално по определен от съответния алгоритъм ред.

Постановката на проблема с конфликтите, съпътстващи работата на комутационните възли е следната: комутаторите в комутационните възли са с размери  $N \times N$ , като  $N$  на брой източници на пакетни съобщения се свързват през комутатора на комутационния възел с  $N$  на брой приемници на тези съобщения. Конфликти се получават при два случая:

- когато един източник на съобщение дава заявка за свързване към два или повече приемника на съобщения.
- когато към един приемник на съобщения има заявка за свързване от два или повече източника на съобщения.
- 

Състоянието на комутатора на един комутационен възел се представя с т.н. матрица на връзките. За комутатор с размери  $N \times N$  матрицата на връзките  $T$  също е с размери  $N \times N$ , като всеки член  $T_{ij} = 1$ , ако има заявка за връзка между източник на съобщение  $i$  и приемник  $j$ . В противен случай  $T_{ij} = 0$ .

Конфликтна ситуация се създава, когато в който и да е ред от матрицата на връзките броя на единиците е по-голям от една, това съответства на случая, когато един източник заявява връзка с повече от един приемник. Наличието на повече от една единица, в която и да е колона от матрицата  $T$  е също указание за конфликтна ситуация и означава, че повече от един източник е заявил връзка към един и същ приемник [1].

### ОПИСАНИЕ НА АЛГОРИТМИТЕ ИЗПОЛЗВАЩИ ДИАГОНАЛНА АКТИВАЦИЯ НА МАТРИЦАТА НА ВРЪЗКИТЕ

Общото между изследваните алгоритми за безконфликтно разписание, е знанието, че диагонално разположените заявки в матрицата на връзките са безконфликтни.

#### 1. Алгоритъм ADA (Algorithm with Diagonal Active)[2].

Същността на алгоритъмът се състои в последователното активиране на диагонално разположените заявки за комуникации в диагонали успоредни на главния диагонал на матрицата на връзките. Под активиране се разбира разрешение за реализиране на заявките. В даден момент разрешени са заявките само на един от успоредните на главния диагонал диагонали и само те се изпълняват. Последни се активират и изпълняват заявките в главния диагонал. Използва се знанието, че заявки разположени диагонално не създават конфликти при едновременното им изпълнение.

ADA (Algorithm with Diagonal Active) е изследван с програмният модел SMADA, описващ алгоритъма ADA на програмния език на MATLAB 6.5. [2].

#### 2. Алгоритъм AJDA (Algorithm with Joint Diagonals Activations) [1].

Идеята на алгоритъма AJDA е основана на знанието, че диагонално разположените заявки за обслужване в матрицата на връзките  $T$  са безконфликтни в диагонала, където са разположени. Съществуват диагонали със заявки, които са безконфликтни един спрямо друг. Съществува възможност безконфликтните диагонали да се активират едновременно и заявките в тях да се изпълняват едновременно и безконфликтно[1].

AJDA (Algorithm with Joint Diagonals Activations) е изследван с програмният модел SMAJDA, описващ алгоритъма AJDA на програмния език на MATLAB 6.5.

#### 3. ADAJS (Algorithm with diagonal activations of joint sub-switching matrices)

Матрицата на връзките  $T$ , която е с размери  $N \times N$ , като  $N$  е степен на двойката се разделя на субматрици ( $S$ ) с размерност  $n \times n$ , ( $n$  също е степен на двойката).

Множествата от субматрици, разположени успоредно на главния диагонал се обработват едновременно във всеки

един от диагоналите. За субматриците в диагонали успоредни на главния се използва принципът на съвместяването[1].

Същността на алгоритъма **ADAJS** (Algorithm with diagonal activations of joint sub-switching matrices) е основана на знанието, че диагонално разположените субматрици със заявки за обслужване в матрицата на връзките  $T$  са безконфликтни в диагонала, където са разположени. Съществуват диагонали със субматрици от заявки, които са безконфликтни един спрямо друг. На Фигура 1 са показани съвместени двойки безконфликтни диагонали със субматрици от заявки означени като стъпки за обслужване и главния диагонал от субматрици, който не може да се съвмести с никой друг.

Целият процес по реализация на безконфликтно разписание чрез алгоритъма **ADAJS** е разделен на стъпки.

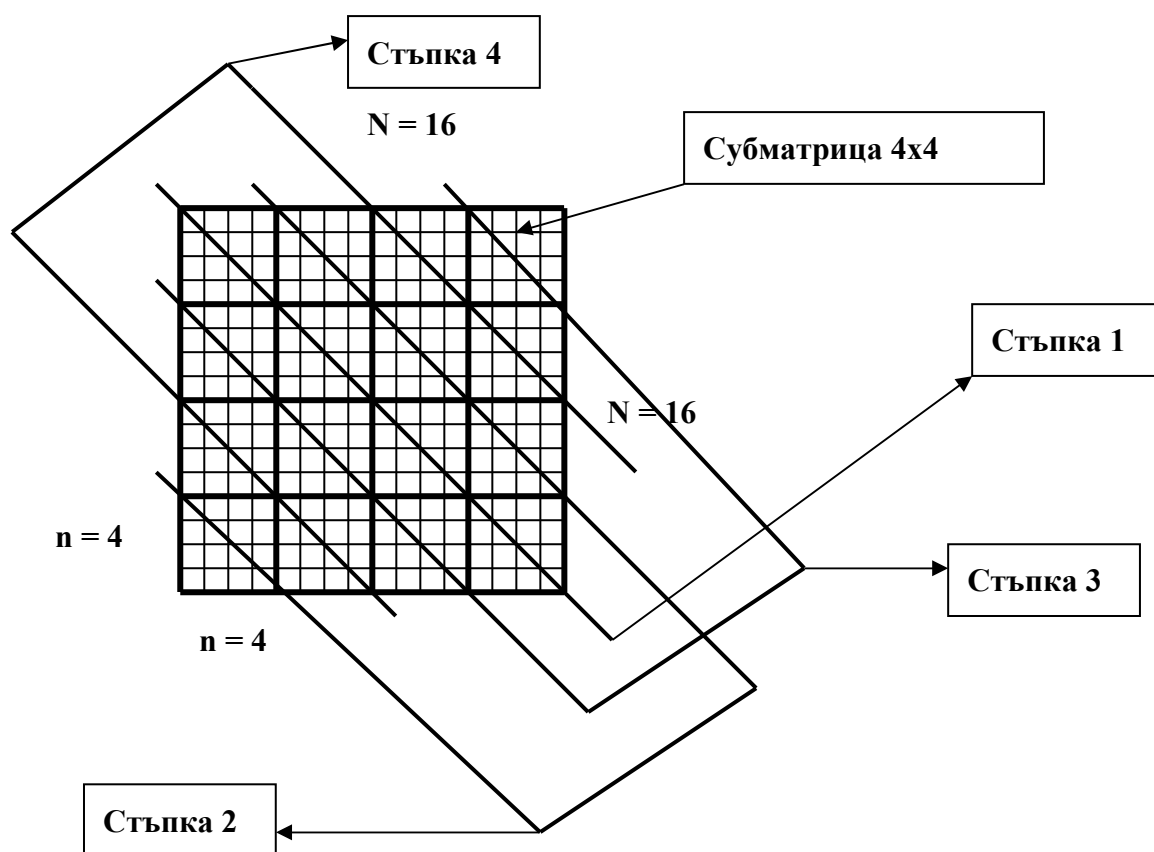
Първата стъпка се отнася до субматриците в главния диагонал, които се обработват едновременно и безконфликтно. Следващите стъпки са свързани със съвместяването на диагоналите, успоредни на главния диагонал по двойки (Фигура 1)[1].

Размерът ( $n$ ) на субматрицата определя броя на стъпките ( $I$ ) по формула 1.

$$I = N/n. \quad (1)$$

При  $N = \text{const.}$ ,  $I = f(n)$ , където  $1 < n \leq N/2$ .

Програмният модел **SMADAJS**, описващ алгоритъма **ADAJS** (Algorithm with diagonal activations of joint sub-switching matrices) е написан на програмния език на MATLAB. Изследването на програмния модел **SMADAJS** е извършено на работна станция Dell Precision 420 [1].



Фигура 1. Съвместени диагонали от субматрици.

СРАВНЕНИЕ НА АЛГОРИТМИТЕ ЧРЕЗ ПРОГРАМНИТЕ ИМ МОДЕЛИ ПО ОТНОШЕНИЕ НА БЪРЗОДЕЙСТВИЕ И НЕОБХОДИМА ПАМЕТ

Сравнението е направено на база резултатите в Таблица 1 и Таблица 2[1]. Резултатите са получени чрез програмните модели **SMADA**, **SMAJDA** и **SMADAJS** на работна станция Dell Precision 420.

**ADAJS** (Algorithm with diagonal activations of joint sub-switching matrices ) е многократно по-бърз от **ADA** (Algorithm with Diagonal Active) и **AJDA** (Algorithm with Joint Diagonals Activations), като нуждите му от памет са пренебрежимо малки в сравнение с **ADA** и **AJDA**. Тези резултати (Таблица 1 и Таблица 2) се отнасят само и единствено когато алгоритмите се интерпретират програмно чрез съответните програмни модели [1].

В настоящето изследване алгоритмите се реализират чрез крайни автомати, като сравнението се прави само по отношение на бързодействието, представено като време за работа на крайния автомат (Таблица 3). Необходимата памет е съсредоточена в крайните автомати и понеже е много малка по обем, сравнението се обезсмисля.

**Таблица 1. Резултати по отношение на бързодействието.**

| N    | Бързодействие [Sec.] SMADA | Бързод. [Sec.] SMAJDA | Бързод. [Sec.] SMADAJS |
|------|----------------------------|-----------------------|------------------------|
| 500  | 26,80                      | 34,39                 | 2,98                   |
| 1000 | 99,21                      | 116,39                | 5,84                   |
| 1500 | 223,61                     | 252,61                | 9,03                   |
| 2000 | 522,96                     | 427,20                | 11,39                  |
| 2500 | 617,37                     | 588,35                | 13,42                  |
| 3000 | 863,60                     | 852,12                | 16,35                  |
| 3500 | 1006,32                    | 994,37                | 18,33                  |
| 4000 | 1258,74                    | 1360,68               | 21,61                  |
| 4500 | 1835,4                     | 2170,36               | 24,29                  |

**Таблица 2. Резултати по отношение на необходимата памет.**

| N    | M [KB] SMADA | M [KB] SMAJDA | M [KB] SMADAJS |
|------|--------------|---------------|----------------|
| 500  | 10 008, 008  | 10 008, 024   | 0,760          |
| 1000 | 40 016, 008  | 40 016, 024   | 0,768          |
| 1500 | 90 024, 008  | 90 024, 024   | 0,768          |
| 2000 | 160 032, 008 | 160 032, 024  | 0,752          |
| 2500 | 250 040, 008 | 250 040, 024  | 0,768          |
| 3000 | 360 048, 008 | 360 048, 024  | 0,768          |
| 3500 | 490 056, 008 | 490 056, 024  | 0,752          |
| 4000 | 640 064, 008 | 640 064, 024  | 0,760          |
| 4500 | 810 072, 008 | 810 072, 024  | 0,768          |

**СИНТЕЗ НА КРАЕН АВТОМАТ ЗА ДИАГОНАЛНА АКТИВАЦИЯ НА МАТРИЦАТА НА ВРЪЗКИТЕ**

Подходящ е краен автомат на Мур, защото на всяко състояние  $A_i$  съответства един изходен сигнал  $St_i$ , който не се влияе от входната дума, както е при автомата на Мили[7]. Необходимо е и едно състояние  $A_0$ , което е начално при рестарт на автомата с изходен сигнал  $R_0$ . В състояние  $A_0$  са забранени всички заявки.

Под действието на  $Z_1$  автоматът преминава последователно през състояния от  $A_0$  до  $A_K$ , като на всяко състояние от  $A_1$  до  $A_K$ , съответства изходен сигнал  $St_1$ (Step1) до  $St_K$ (StepK), управляващ един диагонален от матрицата със заявки или съвместена двойка диагонали

от заявки или съвместени двойки от диагонали със субматрици според съответния алгоритъм.

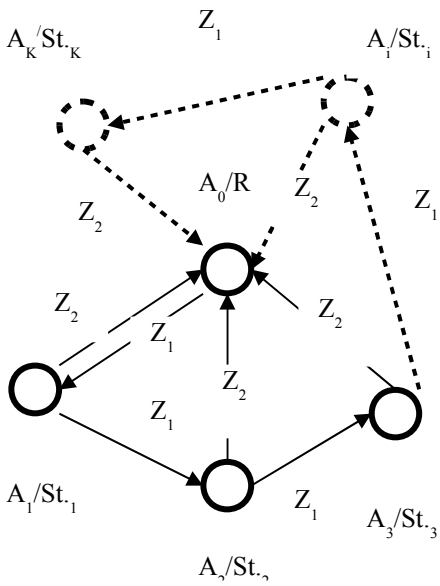
Под действие на  $Z_2$ , автоматът преминава в състояние  $A_0$  с изходен сигнал  $R$ , който е индикация за процесора на комутационния възел, че автомата е свободен и може да бъде стартиран по всяко време. На Фигура 2 е показан обобщен граф на преходите на синтезирания автомат на Мур за управление на диагоналите, съвместените диагонали и диагонали от субматрици.

При всички крайни автомати индексът  $i$  е в границите  $0 \leq i \leq K$ . За алгоритъм **ADA** (Algorithm with Diagonal Active), индексът  $i$  е в границите  $0 \leq i \leq 2N-1$  т.е.  $K = 2N - 1$  в графа на преходите от Фигура 2. При алгоритъм **AJDA** (Algorithm with Joint Diagonals Activations) индексът  $i$  е в границите  $0 \leq i \leq N$  т.е.  $K = N$  в графа на преходите от Фигура 2. По отношение на алгоритъм **ADAJS** (Algorithm with diagonal activations of joint sub-switching matrices) за крайните автомати управляващи диагоналите в субматриците индексът  $i$  е в границите  $0 \leq i \leq n$  т.е.  $K = n$  в графа от Фигура 2, докато за крайния автомат управляващ диагоналите от субматрици индексът  $i$  е в границите  $0 \leq i \leq N/n$  т.е.  $K = N/n$ . Крайният автомат на второ ниво, който управлява съвместените диагонали от субматрици и е с  $N/n$  на брой състояния пребивава  $n$  пъти по-дълго време във всяко едно от състоянията си, отколкото автомата на първо ниво, който управлява диагоналите от заявки в субматрицата.

$\tau$  е означено времето за пребиваване на крайния автомат във всяко от състоянията му. За автомата от второ ниво за алгоритъма **ADAJS** това време е  $n \cdot \tau$ .

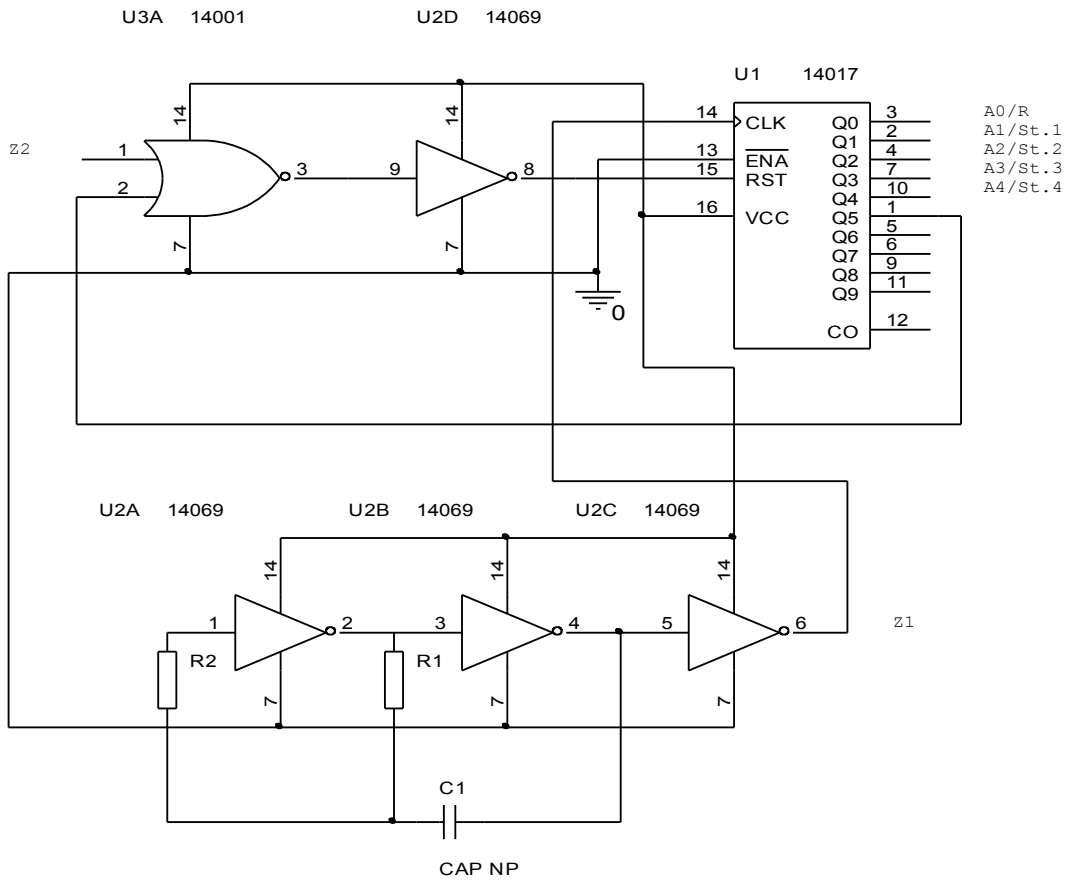
**Таблица 3. Време за работа на крайните автомати за безконфликтна реализация на заявките при алгоритмите с диагонална активация**

| Алгоритми с диагонална активация на заявките | Време за работа на крайните автомати за безконфликтна реализация на заявките |
|--|--|
| <b>ADA</b>                                   | $(2N-1) \cdot \tau$  |
| <b>AJDA</b>                                  | $N \cdot \tau$   |
| <b>ADAJS</b>                                 | $(N/n)n \cdot \tau = N \cdot \tau$   |



Фигура 2. Граф на преходите.

На Фигура 3 е представена принципна електрическа схема на краен автомат на Мур, съответстващ на графа на преходите от Фигура 2, при  $K = 4$ .



Фигура 3. Принципна електрическа схема на краен автомат при  $K = 4$ .

## ЗАКЛЮЧЕНИЕ

Направените изследвания показват, че изцяло хардуерната реализация на безконфликтно разписание, чрез крайни автомати е удачна при алгоритмите с диагонална активация на матрицата на връзките. **AJDA** и **ADAJS** са равностойни и два пъти по-бързи от **ADA**. Времената  $(2N-1)\tau$  и  $N\tau$  са за безконфликтна реализация на заявките, като  $\tau$  се определя от броя на пакетите в заявките. Паралелът между сравненията по отношение бързодействието на програмните модели от една страна и тези при хардуерната реализация с крайни автомати от друга показва, че алгоритъма **ADAJS** е оптимален.

## ЛИТЕРАТУРА

1. Kolchakov K., Monov V. "An Algorithm for Non Conflict Schedule with Diagonal activation of Joint Sub Matrices." Proceedings of the "Distributed Computer and communication networks DCCN 2013", Moscow: R&D Company "Information and Networking technologies" (под печат).
2. Kolchakov K., "An Algorithm Synthesis of Non-Conflict Schedule by Diagonal Connectivity Matrix Activation" Proceedings of the International Conference AUTOMATICS AND INFORMATICS'11, John Atanasoff Society of Automatics and Informatics, Bulgaria, Sofia 03.10-07.10.2011., pp. B-247 – B251, **Proceedings ISSN 1313-1850, CD ISSN 1313-1869.**
3. K. Kolchakov, "An Approach for Performance Improvement of Class of Algorithms for Synthesis of Non-conflict Schedule in the Switch Nodes" , Proceedings of the 11th International Conference on Computer Systems and Technologies, CompSysTech'10, 17-18 June 2010, Sofia, Bulgaria, pp.235-239, **ISBN 978-1-4503-0243-2.**
4. P. Wanjari, A. Chonhari, "Implementation of 4x4 crossbar switch for Network Processor", International Journal of Emerging Technology and Advanced Engineering, Website: [www.ijetae.com](http://www.ijetae.com) ( ISSN 2250 – 2459, Volume 1, Issue 2, December 2011).
5. D. Kim, K. Lee and H. Yoo, " A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip", IEEE International Symposium on Circuits and Systems, 2005.
6. J. Chang, S. Ravi and A. Raghunathan, "FLEXBAR: A crossbar switching fabric with improved performance and utilization", IEEE Custom Integrated Circuits Conference, May 2002.
7. M.Schaller, K. Svozil, "Automaton Logic", International Journal of Theoretical Physics, Vol.35, No. 5, pp. 911-940, 1996, Plenum Publishing Corporation-Springer.
8. D. Serpanos, P. Antoniadis: "FIRM: A Class of Distributed Scheduling Algorithms for High-Speed ATM Switches with Multiple Input Queues", *IEEE Infocom 2000 Conference*, Tel Aviv, Israel, March 2000.
9. P. Gupta, N. McKeown, "Designing and Implementing a Fast Crossbar Scheduler", *IEEE Micro*, Jan.-Feb. 1999, pp.20-28.