

Относно недетерминизма при обработката на събитията в една SMT/DLP машина

Милен Луканчевски, Николай Костадинов, Хованес Авакян

About non-determinism and event-handling in a SMT/DLP machine: XS1 family is a modern implementation of CSP. It is based on the event-driven SMT/DLP XCore and XC parallel programming language. Thus XCore/XC platform represents a CSP machine. The SMT/DLP XCore of the CSP machine supports parallelism at architectural level in terms of processes and communications. XC parallel language uses effectively machine support in terms of parallel constructs.

The select statement is one of useful XC constructs and is an implementation of CSP alternative command. It is used for handling events in DLP manner. Effective event-handling is based on non-determinism in CSP alternative command. The main question here is about non-determinism incorporated in the XC select statement. It is presented a test bed to determine non-determinism support as well as some of the test results are discussed. Finally a technique for incorporating of non-determinism when handling events is proposed. That technique uses CRC32 primitive and the parameterized form of select statement.

Key words: non-determinism, events, CSP, alternative commands, guarded commands, simultaneous multithreading, data level parallelism, XC language, select statement.

ВЪВЕДЕНИЕ

Недетерминизмът е фундаментално понятие, срещано в теорията на автоматите, алгоритмите и паралелната обработка [2]. Той е вътрешно присъщ за паралелните системи и в тази област се разглежда за пръв път от *Dijkstra*. Въвеждането на защитените команди от *Dijkstra* позволява този недетерминизъм не само да се овладее, но и да се използва за намирането на ефективни паралелни решения [5]. Защитените команди на *Dijkstra* са развити от Ноаге в теорията на взаимодействиващите последователни процеси CSP [3, 6].

Смяната на изчислителния модел от последователен към паралелен не се проявява единствено с навлизането на структурния паралелизъм (под формата на многоядрени и многопроцесорни машини). Постепенно се налага архитектурната поддръжка на паралелизма – паралелното изпълнение на множество процеси и взаимодействието им се осигурява не от операционната система, а на машинно ниво.

Фамилията XS1 на фирмата XMOS е съвременна реализация на CSP машина [7]. Тя е базирана на SMT/DLP ядрото XCore и на езика за паралелно програмиране XC. Ядрото XCore осигурява архитектурната поддръжка на паралелизма, а езикът XC предоставя паралелни конструкции за използване на тази поддръжка [9, 10].

В доклада се разглежда оператора select на езика XC, като реализация на алтернативната команда на CSP. Представя се тестова постановка за проверка съответствието на тази реализация на алтернативната команда. Предлага се техника за вграждане на недетерминизма с оглед ефективната обработка на множеството асинхронни събития.

НЕДЕТЕРМИНИРАН ИЗБОР

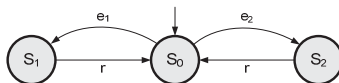
В теорията на автоматите се разграничават детерминирани и недетерминирани автомати. С всеки детерминиран автомат е свързана функцията на преходите $F: (A \times S) \rightarrow S$, където A е азбуката, а S - множеството състояния на автомата. Вход за функцията на преходите се явява даден елемент от азбуката A и конкретно състояние от множеството S . Изход се явява състояние от S . При детерминирани автомати, за всеки вход има точно определен изход. Ако F не е функция, а непълно множество правила, автоматът се определя като недетерминиран [1, 2].

Паралелната обработка предполага друга интерпретация на недетерминизма – той се проявява в неопределеността, в случайния характер на избора на прехода, а не в отсъствието на предварително дефинирани преходи.

Нека е дадена следната защитена команда с две алтернативи

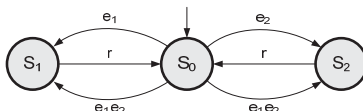
$$\{G_1 \rightarrow P_1 \square G_2 \rightarrow P_2\},$$

където G_1 и G_2 са съответно защитите на процесите P_1 и P_2 . Ако се предположи, че защитите не сработват едновременно, т.е. събитията e_1 и e_2 настъпват поотделно, се получава диаграмата от фиг. 1.



Фиг. 1 – Диаграма на състоянията при детерминиран избор

Ако събитията e_1 и e_2 настъпят едновременно, изборът на алтернатива е случаен – ще се избере или процеса P_1 и ще се премине в S_1 , или ще се избере процеса P_2 и ще се премине в S_2 . След връщането в изходното състояние ще се отработи и другото събитие, тъй като неговата защита все още е в истина.



Фиг. 2 – Диаграма на състоянията при недетерминиран избор

Подобна на ситуацията от фиг. 2 се получава, ако защитите са команди за взаимодействие, например

$$\{P_1?x \rightarrow SKIP \square P_2?x \rightarrow SKIP\}.$$

В този случай, събитието e_1 отговаря на получаването на съобщение от процеса P_1 , а събитието e_2 - на получаването на съобщение от процеса P_2 . Вследствие на паралелизма на ниво данни (DLP), двете събития могат да възникнат едновременно.

ПРОВЕРКА ЗА ПОДДРЪЖКАТА НА НЕДЕТЕРМИНИЗЪМ

Командата на CSP за алтернативен избор има няколко известни реализации – операторите *ALT* в езика *OCAM*, *select* в езика *Ada* [4] и в езика *XC* [9, 10]. Всяка една реализация може да се провери за поддръжка на съдържащия се в алтернативната команда на CSP недетерминизъм.

Опитната постановка представлява паралелна система от три процеса – два източника P_1 и P_2 и един консуматор Q на съобщенията (фиг. 3). Описва се от CSP уравненията

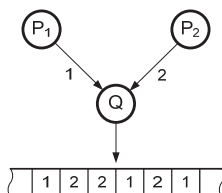
$$\begin{aligned} & \{P_1 || P_2 || Q\}, \\ P_1 = P_2 = P = & * \{Q!msg \rightarrow \Delta \rightarrow SKIP\}_L, \\ Q = & * \{P_1?x \rightarrow write(x) \rightarrow SKIP \square P_2?x \rightarrow write(x) \rightarrow SKIP\}_{2 \times L}. \end{aligned}$$

Източниците P_1 и P_2 изпълняват по L на брой цикъла. В рамките на всеки цикъл изпращат към Q съобщението *msg*, което съдържа номера на процеса – 1 или 2. Консуматорът Q е недетерминиран по определение – съдържа две алтернативи, чиито защиты са команди за взаимодействие. Изпълнява се $(2 \times L)$ пъти, което отговаря на общия брой съобщения, изпращани от P_1 и P_2 .

В тялото на процесите P е предвидена задръжката Δ . Чрез нея се управлява интензитета на съобщенията, като най-високият интензитет е при задръжка $\Delta = 0$. Реализацията на командата за избор трябва обаче да бъде независима от

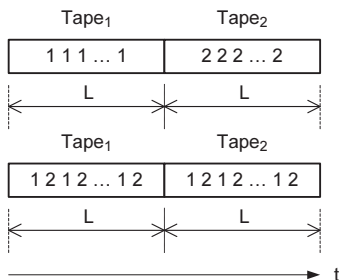
интензитета на съобщенията.

На графа на системата от фиг. 3 е посочена и примерна регистрация на получените от Q съобщения. Регистрираната последователност трябва да има случаен характер, поради случайния характер на избора.



Фиг. 3 – Паралелна система с недетерминизъм – два източника

Извършената с описаната постановка проверка на оператора *select* на ХС за поддръжката на недетерминизъм дава резултата от фиг. 4. При най-тежкия случай ($\Delta = 0$) се проявява детерминиран характер на обработката на събитията, заложен в този оператор (горната половина на фигурата). В резултат, докато не завършат всички взаимодействия с *P1* не се преминава към взаимодействие с *P2*. Предимството, давано на *P1* се обяснява с разположението на алтернативите в оператора *select*. Както даването на подобно „предимство“, така и посоченият начин на отработване на събитията противоречат на изискванията.



Фиг. 4 – Изходна последователност при система с два източника

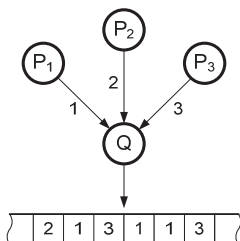
Достатъчно е закъснението Δ да бъде ненулево (в конкретния експеримент е зададена стойност $\Delta = 1$, отговаряща на 10 ns) и обработката на събитията се нормализира – получаването на съобщенията започва да се редува (долната половина на фиг. 4). Детерминираният характер на оператора *select* обаче се потвърждава.

Необходимо е също да се проследят измененията, които настъпват при увеличаване на броя на източниците на съобщения, а оттам и при увеличаване вероятността няколко събития да настъпят едновременно. Така се стига до опитната постановка с три източника, описвана от CSP уравненията

$$\begin{aligned}
 & \{P_1 \parallel P_2 \parallel P_3 \parallel Q\}, \\
 & P_1 = P_2 = P_3 = P = * \{Q!msg \rightarrow \Delta \rightarrow SKIP\}_L, \\
 & Q = * \{P_1?x \rightarrow write(x) \rightarrow SKIP \parallel P_2?x \rightarrow write(x) \rightarrow SKIP \parallel P_3?x \rightarrow write(x) \rightarrow SKIP\}_{3 \times L}.
 \end{aligned}$$

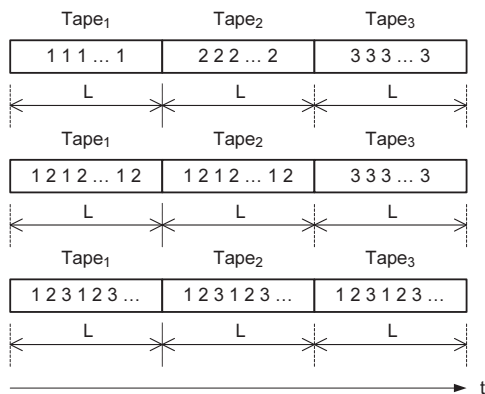
Графът на тази паралелна тестова система от четири процеса съдържа и примерна регистрация на получените от Q съобщения (фиг. 5). Регистрираната последователност отново би трябвало да има случаен характер, поради случайния

характер на избора.



Фиг. 5 – Паралелна система с недетерминизъм – три източника

На практика експериментът потвърждава предходните резултати – докато не спре изпращането на съобщенията от P₁, Q не може да премине към получаване на съобщенията от P₂. А докато не спре изпращането на съобщенията от P₂, Q не може да премине към получаване на съобщенията от P₃ (горната част на фиг. 6). Този резултат се запазва и при внасянето на минимална задръжка – до около 20 ns ($\Delta = 2$). При следваща стъпка на задръжката от 30 ns ($\Delta = 3$) започва да се редува получаването на съобщенията от P₁ и P₂ (средната част на фиг. 6). И едва при задръжка от 50 ns ($\Delta = 3$) и нагоре започва редуването на получаването на съобщенията и от трите източника.



Фиг. 6 – Изходна последователност при система с три източника

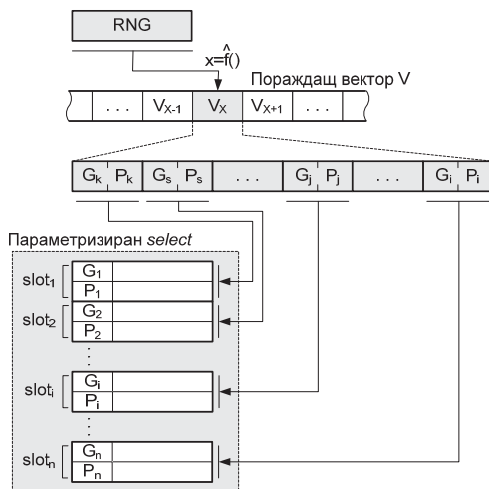
Разгледаният експеримент с оператора *select* е извършен с процесора XS1-L1 и развойната среда XDE версия 11.11.0 на фирмата XMOS. Представява интерес следващата проверка на реактивността на оператора при максимален брой източници на съобщения, който за XS1-L1 е 7, а за XS1-G4 е 31.

ТЕХНИКА ЗА ВКЛЮЧВАНЕ НА НЕДЕТЕРМИНИЗЪМ

Констатираните в т.2 резултати са изненадващи. Те са най-вероятното обяснение защо в документацията на езика XC не се прави пряка съпоставка между оператора *select* и *алтернативната команда*, въпреки връзката между тях. Разбира се, резултатите в голяма степен се дължат на екстремната ситуация, целенасочено създадена и при двете тестови системи – максимално интензивен поток от входящи съобщения. Подобна ситуация е практически рядко срещана. Въпреки това, би

трябвало да се предвидят начини за правилното ѝ отработване, което може да се постигне единствено чрез включването на недетерминизъм при обработката на събитията.

Включването на недетерминизъм би трябвало да използва в максимална степен съществуващата машинна поддръжка и конструкциите на езика XC.



Фиг. 7 – Включване на недетерминизъм при обработката на събитията

Предлаганата техника се основава на параметризирания вариант на оператора *select* на XC [10]. В този вариант операторът *select* се изпълнява като функция, чиито аргументи служат за параметризация на оператора. Идеята е преди обръщението към тази модификация на оператора *select* по случаен начин да се задава реда на алтернативите му.

Предварително се дефинира пораждащия масив V , който съдържа всички възможни уникални подредби на n -те алтернативи. Броят на тези n подредби на n елемента е частен случай на възможните вариации на m елемента от n при $m = n$

$$A_n^m = n(n-1)(n-2) \dots [n-(m-1)] = n(n-1)(n-2) \dots 1 = n!$$

При две алтернативи ($n = 2$), възможните вариации са две, а при три алтернативи ($n = 3$) – шест. Съответно всеки елемент V_x на пораждащия масив представлява вектор, съдържащ елементите на конкретната подредба. А всеки елемент на подредбата представлява структура от параметрите на алтернативата. Най-общо тази структура може да се отбележи като двойката $\langle G_k, P_k \rangle$.

Преди обръщението към параметризирания оператор *select* по случаен закон се определя индекса x от пораждащия масив V за достъп до елемента му V_x . Съдържанието на елемента V_x формира параметрите за обръщение към оператора *select* и определя реда на алтернативите в него (фиг. 7).

За случайния избор е подходяща функцията *crc32()*, капсулация на машинната инструкция *CRC32* [8]. Нейният прототип е

void crc32(unsigned &checksum, unsigned data, unsigned poly)

Параметърът *checksum* отговаря на променливата, съдържаща глобалното състояние на така формирания генератор на псевдослучайни последователности. Параметърът *data* трябва да бъде $0xFFFFFFFF$. Шестнайсетичният код на пораждащия полином се задава чрез параметъра *poly*. Авторите са експериментирали с полинома на *Коортан* с шестнайсетичен код $0xEB31D82E$ [8].

ЗАКЛЮЧЕНИЕ

Недетерминизмът не само е вътрешно присъщ за паралелните системи, той е и средство за ефективно обслужването на множеството асинхронни събития в този клас системи. Представеният експеримент нагледно показва как отсъствието на недетерминизъм води до намаляване реактивността на системата.

За преодоляване на този недостатък, в доклада е предложена техника за вграждане на недетерминизъм при обслужването на събитията. Техниката се основава на случаен избор и на параметризирания вариант на оператора *select*. За максимална ефективност на случайния избор се прилага машинната примитива *CRC32* на ядрото. В резултат се постига необходимата адекватност на оператора *select*, като реализация на алтернативната команда на *CSP*.

Техниката е подходяща за малък брой алтернативи (какви са повечето практически случаи) поради пространствената ѝ сложност и ограничения обем на вградената памет на фамилията *XS1*. При това, времевата ефективност на предложеното решение е константа.

ЛИТЕРАТУРА

- [1] Андерсон, Дж. Дискретна математика и комбинаторика. – М.: Издателският дом „Вилъямс“, 2004.
- [2] Armoni, M., B. Mordechai. The Concept of Nondeterminism: Its Development and Implementations for Teaching. // ACM SIGCSE Bulletin, Vol. 41, Num. 2, pp. 141- 160.
- [3] Bernstein, A. Output Guards and Nondeterminism in “Communicating Sequential Processes”. // ACM Transactions on Programming Languages and Systems, Vol. 2, Num. 2, pp. 234-238.
- [4] Burns, A., A. Wellings. Concurrent and Real-Time Programming in Ada. Cambridge: Cambridge University Press, 2008.
- [5] Dijkstra, E. Guarded Commands, nondeterminacy and formal derivation of programs. // Communications of the ACM, Vol. 18, Num. 8, pp. 453-457.
- [6] Hoare, C.A.R. Communicating Sequential Processes. // Communications of the ACM, Vol. 21, Num. 8, pp. 666-677.
- [7] May, D. The XMOS XS1 Architecture. – XMOS Ltd, 2009.
- [8] Random Numbers on the XS1-L1. – XMOS Ltd, 2010.
- [9] Watt, D. Programming XC on XMOS Devices. – XMOS Ltd, 2009.
- [10] XC Programming Guide. – XMOS Ltd, 2011.

За контакти:

гл. ас. д-р инж. Милен Луканчевски, Катедра “Компютърни системи и технологии”, Русенски университет “Ангел Кънчев”, тел.: 0887 303 850, e-mail: mil@ieee.org

гл. ас. инж. Николай Костадинов, Катедра “Компютърни системи и технологии”, Русенски университет “Ангел Кънчев”, тел.: 082 888 674, e-mail: nkostadinov@ecs.uni-ruse.bg

гл. ас. инж. Хованес Авакян, Катедра “Компютърни системи и технологии”, Русенски университет “Ангел Кънчев”, e-mail: havakian@ecs.uni-ruse.bg

Докладът е рецензиран.



**РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ”
UNIVERSITY OF RUSE „ANGEL KANCHEV“**

ДИПЛОМА

**Програмният комитет на
Научната конференция РУ&СУ'12
награждава с КРИСТАЛЕН ПРИЗ
“THE BEST PAPER”**

**гл. ас. д-р МИЛЕН ЛУКАНЧЕВСКИ,
гл. ас. НИКОЛАЙ КОСТАДИНОВ
и гл. ас. ХОВАНЕС АВАКЯН**

автори на доклада

**“Относно недетерминизма при обработката
на събитията в една SMT/DLP машина”**

DIPLOMA

**The Programme Committee of
the Scientific Conference RU&SU'12
Awards the Crystal Prize "THE BEST PAPER"
to MILEN LUKANCHEVSKY, PhD,
princ. assist. NIKOLAY KOSTADINOV
and princ. assist. HOVANES AVAKYAN
authors of the paper**

**“About non-determinism and event-handling
in a SMT/DLP machine”**

**РЕКТОР
RECTOR**

**проф. д.т.н. Христо Белоев
Prof. DSc Hristo Beloev**

30.10.2012