

багатоагентних систем для вирішення цих проблем та запропоновано низку відповідних підходів на інфраструктурному та прикладному рівнях організації функціонування ОКМ.

1. Dejan S. Milojicic, et al. *Peer-to-Peer Computing*, HP Laboratories Palo Alto, HPL-2002-57, 2002. – 51 p.
2. *Peer to Peer: Harnessing the Power of Disruptive Technologies*, Andy Oram (editor), O'Reilly & Associates Inc., 2001. – 448 p. (<http://www.oreilly.com/catalog/peertopeer>)
3. Jianhua Ma, Makoto Shizuka, Jeneung Lee, Runhe Huang, *A P2P Groupware System with Decentralized Topology for Supporting Synchronous Collaborations*, *Second International Conference on Cyberworlds (CW'03)*, 2003.
4. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, by Gerhard Weiss (Editor), MIT Press, 2000.
5. Michael Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.
6. Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
7. Jos'e M. Vidal, *Learning in Multiagent Systems: An Introduction from a Game-Theoretic Perspective*, University of South Carolina, Computer Science and Engineering, Columbia, SC 29208, 2003.
8. Бутрименко А.В. *Игры автоматов и их применение для управления сетью связи: Автореф. ... канд. дисс.* – М., 1967.
9. Бочкар'єв О.Ю. *Колективна поведінка мобільних інтелектуальних агентів в задачах автономних розподілених досліджень* // *Вісн. Нац. ун-ту "Львівська політехніка"*. – 2005. – № 546. – С. 12–17.
10. Ratnasamy S., Francis P., Handley M., Karp R., Shenker S. *A Scalable Content-Addressable Network* // *Proceedings of the SIGCOMM*. – 2001. – P. 161–172.
11. Бесєдинський Д.І., Бочкар'єв О.Ю. *Організація функціонування однорангових комп'ютерних мереж на основі технології багатоагентних систем* // *Матеріали 1-ї Міжнар. наук. конф. студентів, аспірантів та молодих науковців "Комп'ютерні науки та інженерія – 2006" (CSE – 2006)* – С. 47–50.
12. Бочкар'єв О.Ю. *Вирішення задачі механічного зрівноваження колективом мобільних агентів* // *Вісн. Нац. ун-ту "Львівська політехніка"*. – 2002. – № 463. – С. 14–18.
13. Бесєдинський Д.І. *Стратегії колективної поведінки у конфліктних ситуаціях: Магістерська кваліфікаційна робота.* – Львів, 2005.
14. Botchkaryov A., Kovala S. *CyberCromlech: A new framework for collective behaviour game experiments*, In *Proceedings of 20th European Conference on Modelling and Simulation (ECMS-2006)*, Bonn, Sankt Augustin, Germany, 28–31 May, 2006. – P. 540–545.
15. Льюс Р.Д., Райфа Х. *Игры и решения.* – М.: Изд-во иностр. лит., 1961.
16. Poundstone W. *Prisoner's Dilemma*, Oxford University Press, Oxford, 1992.

УДК 621.372

Є.Я. Ваврук

Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин

ВПЛИВ ОРГАНІЗАЦІЇ ДІАГНОСТИЧНИХ ЯДЕР НА ХАРАКТЕРИСТИКИ СИСТЕМИ ОПРАЦЮВАННЯ СИГНАЛІВ

© Ваврук Є.Я., 2006

Проаналізовано структури діагностичних ядер та систем опрацювання сигналів. Визначено вплив діагностичного ядра на апаратні та часові затрати, достовірність і надійність роботи системи опрацювання сигналів.

Structures of diagnostic cores and systems of signal processing have been reviewed in this article. Influence of diagnostic's cores on hardware, time expenses and reliability of digital signal processing systems is determined.

Вступ

Вимоги до сучасних систем опрацювання сигналів (СОС), насамперед щодо забезпечення роботи в режимі реального часу (РРЧ) та роботи в складних умовах експлуатації і використання в складі високочастотних НВІС окреслили їхні нові особливості як об'єктів діагностування (ОД) та

примушують переглянути стратегію контролю та діагностування. Головний акцент зміщується в бік суттєвого зменшення обсягів додаткового обладнання засобів контролю і діагностування (КіД), що досягається шляхом широкого використання основних (функціональних) апаратних і програмних засобів СОС в процесі проведення діагностування. Результати аналізу шляхів реалізації контролю і діагностування свідчить, що одним з оптимальних варіантів у цьому випадку є використання діагностичних ядер (ДЯ).

Аналіз стану діагностування СОС засобами ДЯ

Загальні підходи до процедури проведення діагностування обчислювальних систем засобами ДЯ і типи самих ядер розроблено давно [1]. Проте є певні особливості їхнього застосування саме для СОС. На основі аналізу засобів КіД для таких систем в [2, 3] доведено доцільність використання ДЯ для діагностування СОС, наведено особливості застосування та сформовано вимоги до них, головними з яких є:

- висока надійність і завадозахищеність;
- можливість визначення несправностей суміжних з СОС систем та вузлів технічного комплексу шляхом аналізу параметрів вхідних і вихідних каналів СОС;
- адаптованість до різних методів контролю працездатності СОС;
- можливість оперативного переналаштування між режимами перевірки та пошуку несправностей;
- програмованість;
- нарощування (зміна) функціональних можливостей і часових параметрів роботи;
- відсутність впливу на параметри СОС під час її роботи;
- незмінність конфігурації зв'язків між основними вузлами системи.

Хоча ці вимоги є загальними і вимагають певних уточнень і доповнень, на їхній основі можна розробляти різні типи ДЯ, які легко адаптуються до основних типів систем СОС.

Проте сьогодні практично відсутні роботи з дослідження впливу організації ДЯ та процедури проведення діагностування на параметри самих СОС. До таких параметрів, виходячи з особливостей побудови і експлуатації СОС, насамперед, необхідно віднести: додаткові часові й апаратні затрати у випадку використання ДЯ та їх вплив на достовірність і надійність роботи системи.

Тому дослідження в даному напрямку є актуальними і доцільними і є цікавими як в теоретичному плані, так і в плані практичної реалізації.

Постановка задачі

На основі аналізу базових структур систем опрацювання сигналів і основних шляхів проведення їхнього діагностування дослідити вплив організації різних типів діагностичних ядер на характеристики систем опрацювання сигналів.

Опис базових структур СОС

Систему чи процесор опрацювання сигналів можна побудувати на основі однієї з структур: система з буферною пам'яттю, надалі “система першого типу” (I); конвеєрна – “система другого типу” (II); векторна – “система третього типу” (III); конвеєрно-векторна – “система четвертого типу” (IV). Вважатимемо, що кількість процесорів у перших трьох структурах дорівнює N , в останній структурі – $N \cdot M$. Структури конвеєрного, векторного і векторно-конвеєрного типу є класичними. Структуру СОС з буферною пам'яттю наведено на рис. 1, де Пр – процесор, БП – блок пам'яті, БФ – буфер, причому кожен з процесорів може звернутися до блоку пам'яті іншого процесора. Процесори, що входять до складу СОС, можуть мати різну складність і бути побудованими на різній елементній базі. Серед них можуть бути програмовані процесори опрацювання сигналів – ППОС (DSP – процесори) та орієнтовані на апаратну реалізацію – АПОС (апаратні процесори опрацювання сигналів, реалізовані на НВІС). Стосовно типу процесорів, що входять до складу системи, поділимо СОС на три класи: 1 – до складу системи входять тільки ППОС (А), 2 – до складу входять ППОС і АПОС (Б), 3 – до складу входять тільки АПОС (В).

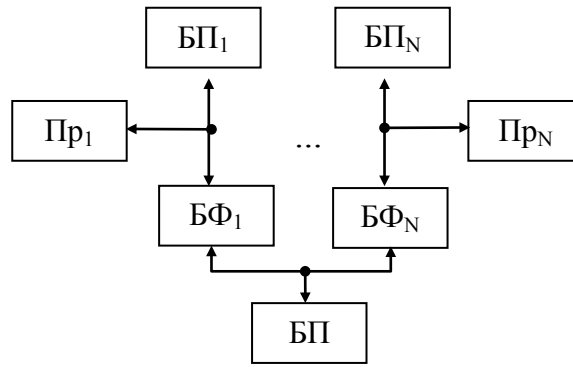


Рис. 1. Блок-схема СОС з буферною пам'яттю

Аналіз шляхів побудови і організації діагностичних ядер

Діагностичні графи, які ілюструють різні принципи діагностування, наведено на рис.2 [1], де а – централізований з послідовною процедурою діагностування (1); б – централізований з послідовно-паралельною процедурою діагностування (2); в – принцип розкручення (3); г – принцип розподіленого ядра (4). На рис. 2 позначено: t_i – час проведення діагностування, $(i+j)$ – порядок проведення діагностування, ЛДЯ – локальне ДЯ, ЦДЯ – центральне ДЯ.

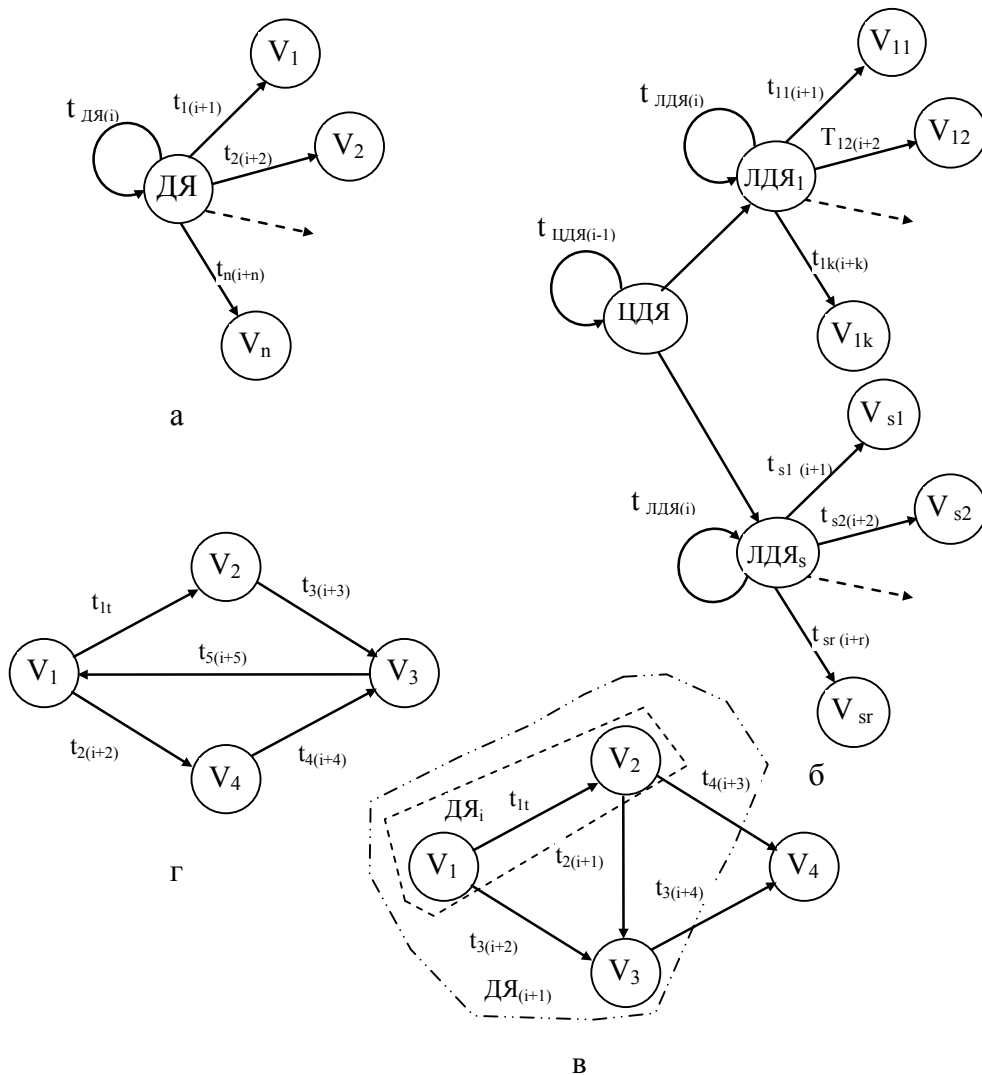


Рис. 2. Діагностичні графи

Принцип централізованого діагностування можна використовувати в системах першого і другого класу, де на одному або кількох програмних процесорах формуються тестові дії, задаються режими діагностування і збирається вся інформація про стан системи. Причому процедура діагностування може бути паралельною, послідовною або паралельно-послідовною.

При використанні *принципу розкручування* на кожному наступному кроці діагностичної процедури діагностичне ядро і апаратура, перевірена за допомогою діагностичного ядра раніше, утворюють засоби діагностування для чергової частки апаратури. При цьому з'являється можливість створення складніших тестів, необхідних для перевірки ще не перевірених вузлів.

Розіб'ємо систему на декілька підсистем V_1, V_2, \dots, V_s так, щоб довільний з вузлів V_q ($i \neq q$) міг діагностуватися хоча б одним з вузлів V_i ($i \neq q$). Тоді вузол V_i діагностує вузол V_q за умови, що з його виходів можна подати тест t_i , що перевіряє вузол V_q , зняти з вузла V_q результат тестування, порівняти з еталоном і видати інформацію про справність (несправність) вузла V_q . Вузол V_i стосовно вузла V_q виконує функції діагностичного ядра. Принцип доцільно використовувати в системах першого і другого класу.

Принцип розподіленого ядра полягає в розбитті апаратури на такі вузли, які були б здатні за певними правилами здійснювати взаємне тестування. При цьому кожний вузол повинен реалізувати не тільки свої робочі алгоритми, але і алгоритми тестового діагностування інших вузлів. Для отримання правильних результатів діагностування необхідно забезпечити певне співвідношення між кількістю одночасно несправних вузлів і кількістю n вузлів у системі. Під час послідовного пошуку несправних вузлів несправний вузол (вузли) має бути відновлений, а процес діагностування продовжений доти, доки не буде отримано інформацію про справність всієї системи.

Принцип розподіленого ядра застосовують для діагностування систем першого класу, кожний з вузлів яких є активним.

Отже, для діагностування систем першого класу можна використовувати всі принципи, а для діагностування систем третього класу жоден принцип використовувати не можна без введення в систему додаткових затрат. Тому для діагностування систем третього класу до відповідної структури необхідно ввести додаткове (початкове) ДЯ.

Для другого і третього класів систем необхідний спеціальний діагностичний інтерфейс, для систем третього класу діагностування може бути забезпечено існуючим міжмодульним інтерфейсом, що реалізує робочий режим функціонування апаратури.

Отже, в загальному випадку ДЯ виконує такі функції: завантаження діагностичної інформації; подачу тестових масивів на вхід вузла, що перевіряється; опитування відповідей з виходу вузла, що перевіряється; порівняння отриманих відповідей з етальонними; аналіз результатів. Очевидно, що результати діагностування правильні тільки в тому випадку, якщо початкове ДЯ є справним вузлом і має вищу надійність порівняно з усіма іншими апаратними засобами. Тому необхідно проектувати його максимально простим і, по можливості, таким, що само перевіряється. Для систем третього класу вимоги до початкового ДЯ виконати простіше, оскільки розробляється процесор, який виконує тільки функції діагностування. Підвищення надійності діагностичного ядра може бути досягнуто як за рахунок застосування відмовостійких методів побудови цифрових вузлів, так і за рахунок мінімізації апаратури діагностичного ядра, у якому необхідно використовувати раціональні схемотехнічні рішення, реалізовані на елементній базі з високим ступенем інтеграції. Тому можливі варіанти побудови діагностичного ядра на базі ППОС і програмованої логіки.

Визначення впливу організації ДЯ на характеристики СОС

Розглянемо, як засоби діагностування впливають на такі параметри системи: обсяг додаткових апаратних затрат (W), обсяг часових затрат (T), достовірність (D) та надійність роботи системи.

Визначення обсягу додаткових апаратних затрат на проведення діагностування залежно від вибраних методів діагностування. Введемо такі позначення: S – кількість діагностичних ядер (ЦДЯ+ЛДЯ), $W_{\text{мнос}}$ – апаратні затрати одного ППОС, $W_{\text{анос}}$ – апаратні затрати одного АПОС, $W_{\text{им}}$ – апаратні затрати на реалізацію діагностичного інтерфейсу системи, L – кількість ППОС в системі.

Вважатимемо, що $W_{анос} = k \cdot W_{тмос}$, де $0 < k \leq 1$; $W_{оя(A,B)} = p \cdot W_{тмос}$, де $W_{оя(A,B)}$ – апаратні затрати ДЯ для першого і другого класів систем, $1 < p \leq 2$; $W_{оя(B)} = r \cdot W_{тмос}$, де $W_{оя(B)}$ – апаратні затрати ДЯ для третього класу систем, $0 < r \leq 1$; $W_{им} = m \cdot W_{тмос}$, де $0 < m \leq N$; в АПОС не передбачено засобів діагностування інших процесорів. Тобто, апаратні затрати на організацію інтерфейсу в найгіршому випадку співмірні з затратами на процесори. За цим підходом не враховують обсяг пам'яті для зберігання додаткових діагностичних даних у штатному ППОС, що використовується як ДЯ. Із використанням другого принципу діагностування $3 \leq L \leq N/2$. Відносний обсяг додаткових апаратних затрат наведено в таблиці, де відповідні типи структури системи позначені (I–IV), а класи системи – (A–B).

Відносний обсяг додаткових апаратних затрат

Тип, клас системи	Принцип діагностування			
	1	2	3	4**
I	2	3	4	5
I А	$p/(p+N-1)$	$pS/[N+S(p-1)]$	$p/(p+N-1)$	$\frac{\sum_{i=1}^{N-1} p_i}{N}$
I Б	$p/[p+kN+L(1-k)-1]$	$pS/[kN+S(p-1)+L(1-k)]$	$p/[p+kN+L(1-k)-1]$	X
I В	$r/(r+kN)$	$rS/(rS+kN)$	$r/(r+kN)$	X
II А*	X	X	$p/(p+N-1)$	$\frac{\sum_{i=1}^{N-1} p_i}{N}$
II Б*	X	X	X	X
II В*	X	X	X	X
II А, III А	$(m+p)/(m+p+N-1)$	$\frac{\sum_{i=1}^S m_i + pS}{\sum_{i=1}^S m_i + pS + N - S}$	$p/(p+N-1)$	$\frac{\sum_{i=1}^{N-1} p_i}{N}$
II Б, III Б	$(m+p)/[m+p+kN+L(1-k)-1]$	$\frac{\sum_{i=1}^S m_i + pS}{\sum_{i=1}^S m_i + pS + L(1-k) + N - S}$	$(m+p)/[m+p+kN+L(1-k)-1]$	X
II В, III В	$(r+m)/(r+m+kN)$	$\frac{\sum_{i=1}^S m_i + rS}{\sum_{i=1}^S m_i + rS + kN}$	$(r+m)/(r+m+kN)$	X
IV А	$(m+p)/(m+p+MN-1)$	$\frac{\sum_{i=1}^S m_i + pS}{\sum_{i=1}^S m_i + S(p-1) + MN}$	$p/(p+MN-1)$	$\frac{\sum_{i=1}^{MN-1} p_i}{MN}$
IV Б	$(m+p)/[m+p+kMN+L(1-k)-1]$	$\frac{\sum_{i=1}^S m_i + pS}{\sum_{i=1}^S m_i + S(p-1) + kMN + L(1-k)}$	$(m+p)/[m+p+kMN+L(1-k)-1]$	X

1	2	3	4	5
IV B	$(r+m)/(r+m+kMN)$	$\frac{\sum_{i=1}^S m_i + rS}{\sum_{i=1}^S m_i + rS + kMN}$	$(r+m)/(r+m+kMN)$	X

* У випадку відсутності діагностичного інтерфейсу.

** Якщо не враховувати додаткового обсягу пам'яті в ППОС для зберігання додаткових діагностичних даних, додаткових затрат обладнання не буде.

X – цей принцип діагностування не використовують для такого класу систем.

Результати аналізу таблиці свідчать:

1. Під час використання будь-якого з принципів діагностування найменші додаткові апаратні затрати для систем з буферною пам'яттю.

2. Для проведення діагностування в класичних структурах СОС (конвеєрна, векторна, конвеєрно-векторна, матрична) необхідне введення додаткового діагностичного інтерфейсу.

3. Стосовно п. 2 при реалізації СОС на базі ППОС і АПОС перевагу порівняно з іншими має принцип розкручення.

4. Використання принципу розкручення і розподіленого діагностичного ядра найефективніше у разі реалізації СОС на базі ППОС.

5. Наведені вирази відображають загальну тенденцію впливу діагностичного ядра на додаткові апаратні затрати.

Визначення обсягу часових затрат на проведення діагностування залежно від вибраних методів діагностування. Введемо такі позначення: $t_{Д(1-ІІІА)}$ – загальний час діагностування (в дужках вказано: „використаний принцип діагностування – тип та клас системи”), t_{nnoc} – час діагностування ППОС, $t_{анос}$ – час діагностування АПОС. Вважатимемо, що $t_{анос} = l * t_{nnoc}$, де $0 < l \leq 1$. Час визначення працездатності ДЯ, що побудований на базі ППОС, становить: $t_{ДЯ} = q * t_{nnoc}$, де $1 < q \leq 2$. Під час аналізу часових затрат у випадку використання принципу розкручування необхідно ввести коефіцієнти u та w додаткових часових затрат, які характеризують ефективність використання діагностичних масивів i -го процесора для діагностування $(i+1)$ -го. Тоді $t_{nnoc(i+1)} = u * t_{nnoc(i+1)}$, де $0 < u \leq 2$, $t_{анос(i+1)} = w * t_{анос(i+1)}$, де $0 < w \leq 2$. Крім того, при використанні принципу розподіленого ядра початково кожен другий з процесорів необхідно вважати діагностичним ядром.

З урахуванням цього визначено такі аналітичні вирази часових затрат для проведення діагностування:

$$t_{Д(1-ІА)} = t_{Д(1-ІІА)} = t_{Д(1-ІІІА)} = t_{nnoc} * (q + N - 1);$$

$$t_{Д(1-ІБ)} = t_{Д(1-ІІБ)} = t_{Д(1-ІІІБ)} = t_{nnoc} * [q + N * l + L * (1 - l)];$$

$$t_{Д(1-ІВ)} = t_{Д(1-ІІВ)} = t_{Д(1-ІІІВ)} = l * t_{nnoc} * (q + N - 1);$$

$$t_{Д(1-ІVА)} = t_{nnoc} * (q + M * N - 1);$$

$$t_{Д(1-ІVБ)} = t_{nnoc} * [q + M * N * l + L * (1 - l)];$$

$$t_{Д(1-ІVВ)} = l * t_{nnoc} * (q + M * N - 1);$$

$$t_{Д(2-ІА)} = t_{Д(2-ІІА)} = t_{Д(2-ІІІА)} = t_{nnoc} * [2q + S - 1 + (N - S) / (S - 1)];$$

$$t_{Д(2-ІБ)} = t_{Д(2-ІІБ)} = t_{Д(2-ІІІБ)} = t_{nnoc} * [2q + S - 1 + L * (N - S) / (S - 1) + l * (N - S - L) * (N - S) / (S - 1)];$$

$$t_{Д(2-ІВ)} = t_{Д(2-ІІВ)} = t_{Д(2-ІІІВ)} = l * t_{nnoc} * [2q + S - 1 + (N - S) / (S - 1)];$$

$$t_{Д(2-ІVА)} = t_{nnoc} * [2q + S - 1 + (M * N - S) / (S - 1)];$$

$$t_{Д(2-ІVБ)} = t_{nnoc} * [2q + S - 1 + L * (M * N - S) / (S - 1) + l * (M * N - S - L) * (M * N - S) / (S - 1)];$$

$$t_{Д(2-ІVВ)} = l * t_{nnoc} * [2q + S - 1 + (M * N - S) / (S - 1)];$$

$$t_{Д(3-ІА)} = t_{Д(3-ІІА)} = t_{Д(3-ІІА*)} = t_{Д(3-ІІІА)} = t_{nnoc} * (q + \sum_{i=1}^{N-1} u_i);$$

$$t_{Д(3-ІБ)} = t_{Д(3-ІІБ)} = t_{Д(3-ІІІБ)} = t_{nnoc} * (q + \sum_{i=1}^{L-1} u_i + l \sum_{i=1}^{N-L} w_i);$$

$$t_{Д(3-IB)} = t_{Д(3-IIB)} = t_{Д(3-IIIB)} = l * t_{nnoc} * (q + \sum_{i=1}^{N-1} w_i);$$

$$t_{Д(3-IVA)} = t_{nnoc} * (q + \sum_{i=1}^{M*N-1} u_i);$$

$$t_{Д(3-IVB)} = t_{nnoc} * (q + \sum_{i=1}^{L-1} u_i + l \sum_{i=1}^{M*N-L} w_i)$$

$$t_{Д(3-IVB)} = l * t_{nnoc} * (q + \sum_{i=1}^{M*N-1} w_i);$$

$$t_{Д(4-IA)} = t_{Д(4-IIA)} = t_{Д(4-IIIA)} = t_{nnoc} * (2q + \log_2 N);$$

$$t_{Д(4-IVA)} = t_{nnoc} * [2q + \log_2 (M*N)];$$

Аналіз вищенаведених формул свідчить, що за використанням принципу розподіленого ядра процес діагностування виконується як найшвидше.

Надійність системи опрацювання сигналів залежить від надійності апаратних засобів (АЗ) та надійності програмного забезпечення (ПЗ). Показники надійності АЗ зменшуються із збільшенням обсягу додаткових апаратних затрат на проведення діагностування. Надійність ПЗ можна підвищити, наприклад, використовуючи принципу багатOVERсійного програмування.

Визначення достовірності роботи системи залежно від вибраних методів діагностування

Для визначення достовірності функціонування необхідно враховувати такі параметри: $D_{\phi}(t)$ – достовірність функціонування, $D_{n,\phi}(t)$ – достовірність правильного функціонування, $D_{n,\phi}(t)$ – достовірність помилкового функціонування.

$$D_{\phi}(t) = 1 - P_{n,\phi}(t) - P_{n,n}(t)$$

$$D_{n,\phi}(t) = P_{np,n}(t) / [P_{np,n}(t) + P_{n,\phi}(t)]$$

$$D_{n,\phi}(t) = P_{n,n}(t) / [P_{n,n}(t) + P_{n,n}(t)],$$

де $P_{n,\phi}(t)$ – ймовірність пропуску помилки засобами діагностування; $P_{n,n}(t)$ – ймовірність того, що СОС працює правильно, але сигнал помилки є наявним, $P_{np,n}(t)$ – ймовірність безпомилкової роботи системи, $P_{n,n}(t)$ – ймовірність неправильної роботи СОС, про що свідчить сигнал помилки.

Показники достовірності оцінюють працездатність як об'єкта діагностування, так і засобів діагностування й істотно залежать від показників останніх. Тому їх використовують як критерії, за якими обирають засоби діагностування на етапі проектування СОС і оцінювання її ефективності. Для СОС, що працюють у критичних умовах, головним показником при діагностуванні повинна бути достовірність помилкового функціонування.

Висновки

В роботі оцінено вплив організації діагностичних ядер на характеристики різних структур систем опрацювання сигналів. Визначені показники є якісними характеристиками і допомагають вибрати ефективний принцип діагностування для певної структури системи опрацювання сигналів. Параметри визначали для системи опрацювання сигналів на базі однотипних процесорів; до складу ППОС входить сигнальний процесор (DSP – процесор фірми Analog Devices) та ПЛІС фірми Altera, АПОС реалізовані на базі ПЛІС. Вважалося, що об'єм вузлів пам'яті кожного з процесорів достатній для зберігання всіх діагностичних коефіцієнтів, констант, програмних кодів. Хоча ефективність використання запропонованих підходів за інших вихідних даних для кожної з структур якісно не зміниться, кількісні показники можуть істотно відрізнятись. Конкретні значення параметрів можна визначити тільки на основі аналізу всіх часових і функціональних показників певної системи. Тому цей підхід не можна трактувати як універсальний. Напрямами подальших досліджень автора є:

1. Розроблення нових типів діагностичних ядер
2. Розроблення комплексного критерію оцінки взаємовпливу різних принципів діагностування на ефективність СОС.

3. Використання для побудови ДЯ і організації процесу діагностування основних теоретичних засад експертних систем, теорії моделювання систем, штучного інтелекту.
4. Шляхи підвищення надійності СОС організацією багатoversійного програмування.
5. Аналіз впливу додаткових обсягів пам'яті на проведення діагностування, параметрів та надійності діагностичних каналів на ефективність СОС.

1. Щербаков Н.С., *Достоверность работы цифровых устройств.* – М.: Машиностроение, 1989. – 224 с. 2. Ваврук Є.Я. Вибір діагностичного ядра для системи опрацювання сигналів // Зб. наук. праць ІПМЕ НАНУ. – 2006. – Вип. 34. – С. 23–29. 3. Ваврук Є.Я. Організація контролю і діагностики бортових пристроїв опрацювання сигналів у режимі реального часу// Збірник матеріалів міжвузівської науково-технічної конференції науково-педагогічних працівників. – Львів: Ліга-Прес, 2006. – С. 191–192.

УДК 004.274

С.О. Власенко

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ПОБУДОВА АДАПТИВНИХ ІНТЕРФЕЙСІВ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ КОНФІГУРОВАНИХ АРХІТЕКТУР

© Власенко С.О., 2006

Розглянуто проблеми побудови адаптивних інтерфейсів у комп'ютерних системах та запропоновано реалізацію одного з методів на основі механізму часткової реконфігурації ПЛІС.

The problems of adaptive interfaces construction for the computer systems and it's realization on basis of partly-configured FPGA approach have been presented.

Вступ

Конфігуровані архітектури [3], або такі архітектури, що дають змогу збільшувати свої функціональні можливості під впливом наборів спеціальних керівних сигналів (сигналів конфігурування) з кожним роком все частіше використовують. Значний поштовх цьому процесу надає стрімкий розвиток технології FPGA (Field Programmable Gate Array) – програмованих логічних інтегральних Схем (ПЛІС). Сам факт появи як конфігурованих систем, так і ПЛІС зокрема, спричинили так звану “хвилю” намагань досягти з їх допомогою значного прискорення виконання різного роду обчислень аж до побудови конфігурованих суперкомп'ютерів. Дійсно, багатьом розробникам вдалося реалізувати на конфігурованих архітектурах прискорення деяких видів задач у сотні та навіть тисячі разів порівняно з їх виконанням на універсальних робочих станціях [2]. Зазвичай, найбільшого прискорення вдається досягти тоді, коли задача реалізується на великій кількості однорідних елементів, що працюють паралельно. Але, незважаючи на чудові “лабораторні” результати, на практиці не завжди вдається досягти загального збільшення швидкодії системи. Проектування архітектур паралельного оброблення даних є досить складним та, що дуже важливо, довготривалим процесом. Отже, конфігуровані комп'ютери з часом позиціонувалися як засоби, що дають змогу зменшити час виконання певного виду алгоритмів за досить значних витрат на їхнє проектування.

У цьому контексті одним з найперспективніших напрямків [1] використання конфігурованих архітектур стає їх використання як засобів комунікації та складових частин систем реального часу.