

проектирования задания только структуры ТО недостаточно, потому что параметры (характеристики) подсистем, составляющих технологическую оснастку отличаются, то есть, необходимо параметрическое описание наряду со структурным. Это дает возможность генерировать множество альтернатив объекта и доставлять информацию для моделей функционирования.

Представлен структурно – параметрический граф литьевых форм

Таким образом, предложенная модель литьевой формы учитывает связи, налагаемые конкретными техническими ограничениями, которым должна удовлетворять проектируемая конструкция оснастки. Эта

модель дает возможность разработать оптимальное математическое программное обеспечение интегрированной системы автоматизированного проектирования технологической оснастки для изготовления пластмассовых деталей.

ЛИТЕРАТУРА:

1. Гуль В.Е., Акутин М.С, Основы переработки пластмасс - М.; Химия. 1985. - 400 с.
2. Modern plastic Analyzing plastics with FEA: Part 4 By Michelle Maniscalco, Jan 01, 2007
3. Жолткевич Г. Н. Автоматизация проектирования технологической оснастки.— К.: Техніка, 1998. — 263 с.

УДК 681.326

ОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ С РЕКОНФИГУРИРУЕМОЙ СТРУКТУРОЙ

К.т.н. М.А. Бережная, Я.Ю. Королева, Харьковский национальный университет радиотехники

В статье предлагается метод построения отказоустойчивых вычислительных сетей процессорных модулей (ПМ), путем реконфигурации сети и замены неисправного ПМ резервным модулем. Предложен алгоритм оптимального размещения резервного модуля, который обеспечивает минимизацию затрат на реконфигурацию.

In this paper, we study the design of fault-tolerant cellular arrays by adding redundant nodes and edges in target graph model. The technique of designing fault-tolerant arrays with both a small number of spare nodes and small degree is developed.

1. Введение

Развитие и внедрение субмикронных технологий в процесс производства современных вычислительных и управляющих систем, широкое использование программируемых логических интегральных схем (ПЛИС), сигнальных процессоров, микроконтроллеров, возможность размещения на одной подложке до 10^9 транзисторов, повышение тактовой частоты работы электронных элементов до $1\div 5$ ГГц, расширение числа выводов, превышающего 1000 на одну микросхему снижение их стоимости, позволяет перейти к созданию систем на одном кристалле, обладающих принципиально новыми архитектурными и функциональными свойствами, такими, как самоорганизация, реконфигурация, искусственный интеллект.

Среди современных многопроцессорных вычислительных, информационно-измерительных и управляющих систем существенно возросло количество реактивных систем, то есть, систем постоянно взаимодействующих с окружающей средой. Это системы реального времени в результате функционирования, которых вырабатывается реакция (отсюда название «реактивные») на входную информацию, в том числе и на изменяющееся воздействие окружающей среды. Эффективность таких систем определяется высоким быстродействием и производительностью, которая достигается конвейеризацией и параллелизмом

обработки информации, как на уровне процессоров, так и на уровне аппаратной реализации алгоритмов.

С другой стороны ошибки в функционировании таких систем, как многопроцессорные информационно-управляющие системы химическим производством и технологическим оборудованием, ядерными реакторами, транспортными средствами, средствами космической техники могут привести к катастрофическим последствиям с большими материальными потерями и человеческими жертвами. Поэтому к надежности и отказоустойчивости функционирования такого рода систем предъявляются чрезвычайно высокие требования.

В широко известных работах по созданию наиболее перспективных вычислительных систем (ВС) В.М. Глушкова, Э.В. Евреинова, В.И. Варшавского, А.В. Палагина, А.В. Каляева, И.В. Прангиншвили, Д.А. Поспелова и др. определены основные характеристики модели коллектива вычислителей: 1) параллельность выполнения операции; 2) конструктивная однородность; 3) переменность и реконфигурируемость структуры. Современная элементная база позволяет создавать высокопроизводительные, экономичные, однородные вычислительные системы (ОВС), удовлетворяющие этим требованиям. В последние годы возрастание стоимости производства заказных СБИС специального применения (ASIC – application specific IC's) определяет интерес разработчиков ВС к использованию функциональных модулей ПЛИС типа FPGA и CPLD, микроконтроллеров, умножителей, ОЗУ и ПЗУ для создания высокопроизводительных однородных сетей (ОС) с реконфигурируемыми архитектурами [1,2,3,4,5,6].

Большое число работ в этой области посвящено разработке теоретических и практических аспектов построения ОС, отражающих их максимальные конструктивные и вычислительные характеристики, возможности моделирования сложных динамических систем, реконфигурации и оптимального размещения вычислительных и управляющих алгоритмов, автоматных моделей в ОС.

С возрастанием степени интеграции и плотности компоновки СБИС возрастает вероятность появления различных дефектов в процессе их производства, что в целом существенно влияет на показатели надежности сетей из современных СБИС. Поэтому разработка методов реконфигурации ОС, обеспечивающих требуемые показатели надежности и отказоустойчивости всей системы, является актуальной проблемой.

Анализ существующих методов построения отказоустойчивых ОС показывает, что существует два подхода к решению этой проблемы. В одном из них предполагается, что неисправный ПМ исключается из ОС, а его функции распределяются между исправными ПМ, обеспечивающие работоспособность сети. В этом случае в ОС не используются избыточные резервные ПМ, что обеспечивает экономичность сети, но снижает ее производительность. Такие ОС называют сетями с деградирующей производительностью (Degradable Arrays) [7,8,9,10].

В другом подходе предполагается использовать резервные ПМ, позволяющие реконфигурировать ОС при появлении неисправных модулей без снижения производительности сети [11,12,13,14]. В статье представлен метод проектирования отказоустойчивых сетей с резервными ПМ, основанный на решении задачи минимаксного размещения резервных ПМ, обеспечивающих реконфигурацию структуры ОС с минимальными временными затратами на перекоммутацию и пересылку данных между ПМ и изоморфизм реконфигурированной ОС первоначальной архитектуре и вычислительным характеристикам ОС [14,15,16].

2. Микропроцессорные однородные сети с реконфигурируемой структурой.

Предложен метод реконфигурации однородных сетей (ОС) процессорных модулей ПМ при наличии в сети одного неисправного модуля. Этот подход может применяться к модульным процессорным сетям различного назначения, в которых неисправный ПМ обнаруживается либо внешними, либо встроенными средствами диагностирования.

Если между модулями сети нет взаимосвязи, то простейшим решением является исходное подключение резервных модулей к общей шине. В случае отказа работающего модуля включается один из резервных, выполняется пересылка данных и неисправный модуль отключается.

Наличие взаимосвязей между модулями сети усложняет задачу реконфигурации ОС. В этом случае явно недостаточно подключить резервный модуль и отключить неисправный. Должна быть создана новая схема интерфейсных связей между модулями, которая зависит от области применения ОС. На рис 1 представлена наиболее распространенная структура ОС, которая будет в дальнейшем исследоваться более детально. В этой ОС модули соединены в одномерную

ОС с однонаправленными связями между модулями. Очевидно, что при отказе одного из модулей такая система становится неработоспособной.

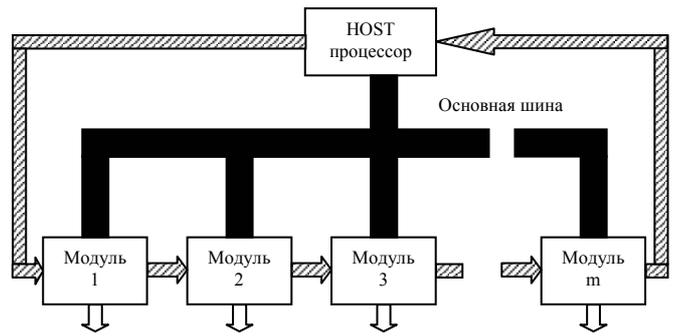


Рис. 1. Однонаправленная ОС процессорных модулей

При отказе модуля необходимо выполнить следующую последовательность операций: 1) локализовать неисправные модули; 2) определить новую рабочую конфигурацию структуры; 3) выполнить соответствующую пересылку данных; 4) реконфигурировать систему путем перекоммутации модулей.

3. Централизованное управление реконфигурацией.

В этом методе построение отказоустойчивых ОС управление осуществляет HOST компьютер с учетом: а) объема данных, которые необходимо переслать перед выполнением реконфигурации; б) сложности алгоритма реконфигурации; в) числа s резервных модулей, необходимых для обеспечения отказоустойчивости системы; г) сложности коммутационной схемы, необходимой для выполнения реконфигурации.

Пример выполнения реконфигурации ОС на основе коммутационной сети приведен на рис. 2 [2,3,6,8]. В этой структуре допускаются произвольные соединения между модулями. На рис. 2 а представлена однородная ОС с $(m + s)$ ПМ коммутационной сетью. Для каждой интерфейсной связи между ПМ, коммутационная сеть должна иметь $(m + s)$ входных и столько же выходных контактов. Однако в каждый момент времени используется только m соединений между этими контактами, в связи с этим необходимо использовать коммутационную сеть с неполной перекоммутацией, которая может быть реализована более экономично.

На рис. 2 (б, в) иллюстрируется процесс реконфигурации при наличии неисправного ПМ под №2 в ОС с $m = 4$ – число ПМ в структуре ОС и $s = 2$ – число резервных ПМ. Основным недостатком данной схемы является значительная сложность коммутационной сети, которая возрастает пропорционально с ростом числа m ПМ и линий связи между ними. Кроме того, такая ОС является ядром вычислительной системы, которая должна быть защищена от устойчивых и перемежающихся неисправностей, что создает дополнительные проблемы при проектировании ОС.

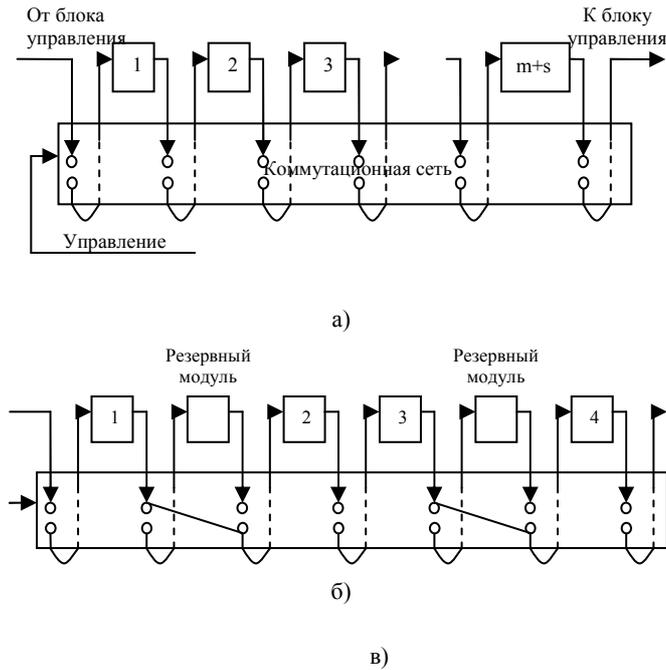


Рис. 2. Реконфигурация на основе коммутационной сети:
 а) $m + s$ модулей, соединенных через коммутационную сеть;
 б) функционирование с двумя резервными модулями;
 в) структура сети с неисправным модулем 2

Идея использования шунтирующих коммутационных сетей была предложена в [9]. Такие сети реконфигурируются из простых ячеек с двумя типами соединений и имеющие по два входа и выхода. Примеры такой ячейки и двух возможных соединений приведены на рис. 3. На рис. 4 показано использование таких ячеек в механизме реконфигурации.

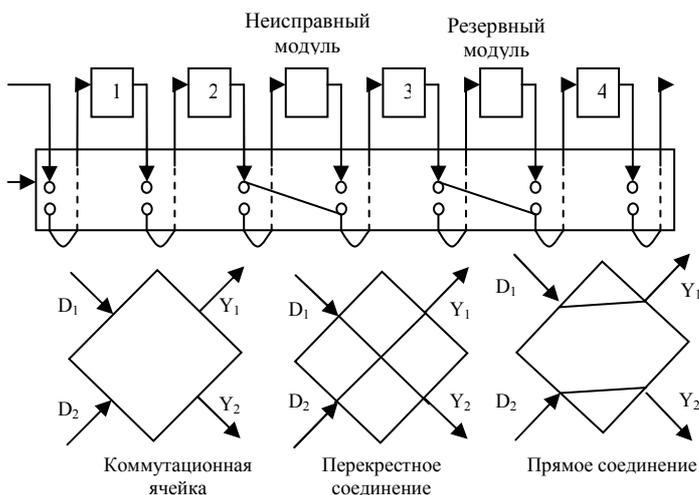


Рис. 3. Коммутационная ячейка с двумя типами соединений

Недостатком данной схемы, как видно из рис. 4, является необходимость пересылки большого объема данных в случае возникновения неисправности. Число таких пересылок можно сократить путем оптимального размещения резервных модулей в структуре ОС.

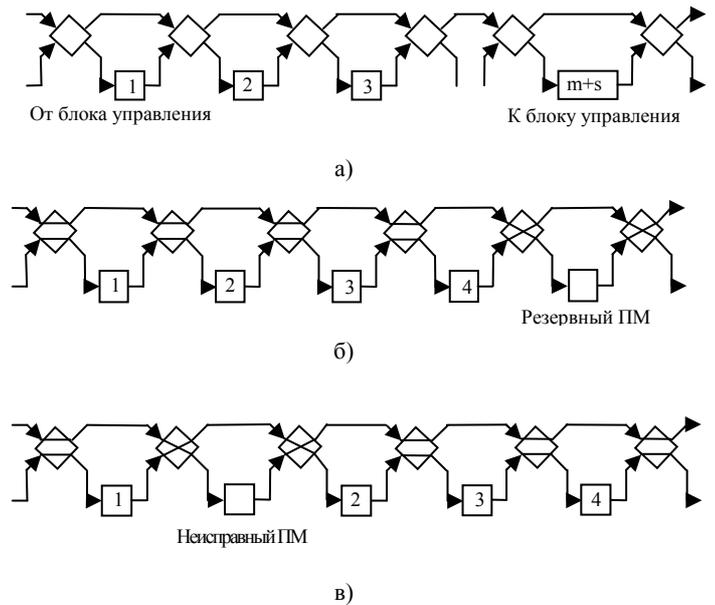


Рис. 4. Реконфигурация на основе шунтирующей коммутационной сети
 а) $m+s$ ПМ, соединенных через шунтирующую сеть;
 б) структура сети с одним резервным ПМ;
 в) структура сети с неисправным ПМ 2

При наличии в сети m работающих и одного резервного модуля среднее число необходимых пересылок данных для модуля, находящегося в k -й позиции ($i \leq k \leq m+1$) составит:

$$\frac{1}{m} \cdot [1 + 2 + \dots + (k-1) + 1 + 2 + \dots + (m+1-k)] = \frac{m+1}{2} - \frac{(k-1) \cdot [(m+1)-k]}{m}$$

Из этого выражения следует, что объем пересылаемой информации, является максимальным для  или $m+1$ и минимальным для $k = m/2 + 1$. Минимальное число пересылок составит $(m+2)/4$ для четных m .

При использовании большего числа резервных модулей ($s > 1$) аналогичное уравнение составить намного сложнее. Предлагается следующий подход к решению этой задачи. Предположим, в ОС, имеющей p неисправных модулей () отказывает $(p+1)$ -й модуль. Пусть резервные модули находятся в позициях $k_1 < k_2 < \dots < k_s$. Их расположение является оптимальным только до первого отказа, интуитивно можно предположить, что оптимальным размещением является расположение резервных модулей в середине группы из $1 + m/s$ модулей.

Аналогичный результат может быть получен следующим образом. Находится среднее число пересылок при первом отказе как функция от (k_1, k_2, \dots, k_s) переменных. Полученная функция

минимизируется. Для неисправного модуля с порядковым номером, меньшим, чем k_l , среднее число пересылок составит:

$$\frac{1}{k_l - 1} \cdot [(k_l - 1) + (k_l - 2) + \dots + 1] = \frac{k_l}{2}$$

Для исправного модуля с порядковым номером, большим, чем k_s , среднее число пересылок равно:

$$\frac{1}{m + s - k_s} \cdot [(m + s - k_s) + (m + s - k_s - 1) + \dots + 1] = \frac{m + s - k_s + 1}{2}$$

Для исправного модуля с порядковым номером, заключенным между k_{p+1} и k_p $p = 1, 2, \dots, (s-1)$, среднее число пересылок представляется как:

для нечетных значений $k_{p+1} - k_p$:

$$\frac{2}{k_{p+1} - k_{p-1}} \cdot [1 + 2 + \dots + \frac{k_{p+1} - k_{p-1}}{2}] = \frac{k_{p+1} - k_{p-1}}{4}$$

для четных значений $k_{p+1} - k_p$:

$$\frac{1}{k_{p+1} - k_{p-1}} \cdot [1 + 2 + \dots + \frac{k_{p+1} - k_{p-1}}{2} + \dots + 2 + 1] = \frac{(k_{p+1} - k_{p-1})^2}{4 \cdot (k_{p+1} - k_{p-1})} = \frac{k_{p+1} - k_{p-1}}{4} + \frac{1}{4 \cdot (k_{p+1} - k_{p-1})}$$

Для упрощения можно пренебречь величиной $1/[4 \cdot (k_{p+1} - k_{p-1})]$ и использовать значение $(k_{p+1} - k_p)^2 / [4 \cdot (k_{p+1} - k_p - 1)]$ для двух рассматриваемых случаев.

Таким образом, среднее число пересылок данных составит:

$$T = \frac{1}{m} \cdot \left[\frac{k_l}{2} (k_l - 1) + \frac{m + s - k_s + 1}{2} \cdot (m + s - k) + \frac{1}{4} \sum_{p=1}^{s-1} (k_{p+1} - k_p)^2 \right]$$

$$\frac{dT}{dk_l} = \frac{3 \cdot k_l - k_2 - 1}{2 \cdot m}$$

$$\frac{dT}{dk_s} = \frac{3 \cdot k_s - k_{s-1} - (2 \cdot m + 2 \cdot s + 1)}{2 \cdot m}$$

для $1 < r < s$

$$\frac{dT}{dk_r} = \frac{2 \cdot k_r - k_{r+1} + k_{r-1}}{2 \cdot m}$$

Положив $\frac{dT}{dk_p}$, получим следующую систему

линейных уравнений:

$$\begin{cases} 3 \cdot k_1 - k_2 = 1 \\ -k_1 + 2 \cdot k_2 - k_3 = 0 \\ -k_2 + 2 \cdot k_3 - k_4 = 0 \\ \dots \dots \dots \\ -k_{s-2} + 2 \cdot k_{s-1} - k_s = 0 \\ -k_{s-1} + 3 \cdot k_s = 2 \cdot m + 2 \cdot s + 1 \end{cases}$$

Найдем значения $k_2 \dots k_s$ в виде:

$$k_2 = 3 \cdot k_1 - 1$$

$$k_3 = 5 \cdot k_1 - 2$$

$$k_4 = 7 \cdot k_1 - 3$$

.....

$$k_s = (2 \cdot s - 1) \cdot k_1 - (s - 1)$$

Выполнив подстановку, получим:

$$-(2 \cdot s - 3) \cdot k_1 + (s - 2) + 3 \cdot (2 \cdot s - 1) \cdot k - 3 \cdot (2 \cdot s - 1) = 2 \cdot m + 2 \cdot s + 1$$

или

$$k_1 = 1 + \frac{m}{2 \cdot s}$$

выведем общее уравнение для вычисления k_r :

$$k_r = r + \frac{m \cdot (2 \cdot r - 1)}{2 \cdot s}, \quad r = 1, 2, \dots, s$$

Пример. Для $m = 16$ и $s = 3$ (16 работающих ПМ и 3 резервных) определим номера позиций резервных ПМ: $k_1 = 4$; $k_2 = 10$; $k_3 = 16$.

Алгоритм реконфигурации.

В алгоритме используется два обобщенных объекта:

1) *состояние* – вектор, характеризующий состояние ПМ (0 – резервный, 1 – работающий, 2 – неисправный);

2) ячейки – вектор, характеризующий состояние коммутационных ячеек (0 – перекрестное соединение, 1 – прямое соединение).

Входным аргументом для данного алгоритма является номер неисправного ПМ. Алгоритм изменяет значения векторов *состояние* и *ячейки*, а также инициирует процесс пересылки данных. Фактически, вектор ячейки можно не использовать, так как его значение однозначно определяется вектором *состояние*. Однако вектор *ячейки* в дальнейшем будет использоваться, так как алгоритм его обработки и модификации очень прост.

Алгоритм реконфигурации коммутационной сети может быть представлен в виде последовательности следующих шагов:

Шаг 1. Проверить наличие резервных ПМ в сети. Если такой ПМ имеется, перейти к шагу 2 алгоритма, в противном случае выполнить функцию *отказ_системы* и завершить выполнение алгоритма,

Шаг 2. Определить номер позиции резервного ПМ, использование которого позволяет добиться минимального числа пересылок данных.

Шаг 3. Модифицировать значение вектора *ячейки*, изменив состояния коммутационных ячеек на входах и выходах как неисправного, так и вновь включаемого ПМ.

Шаг 4. Модифицировать значение вектора *состояние*, указав новое состояние (1) подключаемого резервного ПМ.

Шаг 5. Выполнить функцию КОП для пересылки данных между ПМ, в случае разрушения данных выполнить процедуру их восстановления путем повторного выполнения фрагментов программы.

Шаг 6. Модифицировать значение вектора *состояние*, указав новое состояние (2) неисправного ПМ. Завершить выполнение алгоритма.

Так как шунтирующая коммутационная сеть является частью ядра системы, она должна быть также отказоустойчивой.

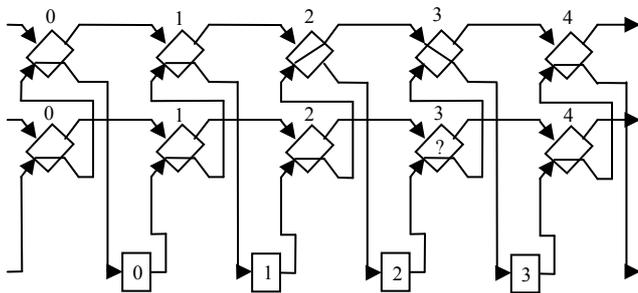


Рис. 5. Избыточная шунтирующая коммутационная сеть

Работоспособность шунтирующей сети может быть восстановлена при возникновении неисправностей, которые могут быть обнаружены с помощью самопроверяемых схем контроля. При использовании избыточной коммутационной сети, приведенной на рис. 5, можно восстановить работоспособность сети при

появлении неисправностей всех трех классов. Как показано на рис. 5, в случае отказа коммутационной ячейки 3, передача данных через сеть не прерывается. Однако в результате такой неисправности ПМ 2 в дальнейшем никогда не используется.

4. Заключение

В статье предложен метод построения одномерных однородных сетей из процессорных модулей с централизованным управлением реконфигурацией. Разработаны и предложены схемы и алгоритм коммутации ОС с централизованной реконфигурацией и шунтирующим коммутационными сетями. Предложен метод расчета оптимальных затрат на реконфигурацию, определяемых местами размещения резервных модулей.

ЛИТЕРАТУРА:

1. Аладьев В.З. Однородные структуры. – К.: Техника, 1990. – 272 с.,
2. Евраинова Э.В. Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.,
3. Прангшвили И.В., Виленкин С.Я., Медведев И.Л. Параллельные вычислительные системы с общим управлением. – М.: Энергоатомиздат, 1983. – 312 с.
4. Глушков В.М., Капитанова Ю.В., Летигевский А.А. Теория структур данных и синхронные параллельные вычисления. // Кибернетика. – 1976. – №6. 2 – 5 с.,
5. Палагин А.В. Опанасенко В.Н. Реконфигурируемые вычислительные системы. – Киев: Просвіта, 2006. – 295 с.,
6. Барский А.Б. Параллельные информационные технологии: учебное пособие. – М.:, 2007. – 503 с.,
7. J. Hastad, T. Leighton, and M. Newman, 'Fast Computation Using Faulty Hypercubes (extended abstract)', Proc. ACM Symp. Theory of Computing, pp.251-263, 1989.,
8. C. Kakkamanis, A.R. Karlin, et al., 'Asymptotically Tight Bounds for Computing with Faulty Arrays of Processors (extended abstract)', Proc. IEEE Symp. Foundations of Computer Science, pp. 285-296, 1990.,
9. F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. San Mateo, Calif.: Morgan-Kaufmann, 1992.,
10. T. Leighton, B. Maggs, and R. Sitaraman, 'On the Fault Tolerance of Some Popular Bounded-Degree Networks', Proc. IEEE Symp. Foundations of Computer Science, pp. 542-552, 1992.,
11. M. Ajtai, N. Alon, et al., 'Fault Tolerant Graphs, Perfect Hash Functions and Disjoint Paths', Proc. IEEE Symp. Foundations of Computer Science, pp. 693-702, 1992.,
12. N. Alon and F. Chung, 'Explicit Construction of Linear Sized Tolerant Networks', Discrete Math., vol. 72, pp. 15-19, 1988.,
13. J. Bruck, R. Cypher, and C. Ho, 'Wildcard Dimensions, Coding Theory and Fault-Tolerant Meshes and Hypercubes', IEEE Trans. Computers, vol. 44, no. 1, pp. 150-155, Jan. 1995.,
14. A.L. Rosenberg, 'The Diogenes Approach to Testable Fault-Tolerant Arrays of Processors', IEEE Trans. Computers, vol. 32, pp. 480-489, 1983.,
15. A.L. Rosenberg, 'Fault-Tolerant Interconnection Networks: A Graph-Theoretic Approach', Proc. Ninth Workshop Graph-Theoretic Concepts in Computer Science, pp. 286-297, 1983.,
16. Кристофидес Н. Теория Графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.