

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
РАДИОЭЛЕКТРОНИКИ

ISSN 1563-0064

РАДИОЭЛЕКТРОНИКА И ИНФОРМАТИКА

Научно-технический журнал

Основан в 1997 г.

№ 3(78), июнь – сентябрь 2017

Выходит 4 раза в год

© Харьковский национальный
университет радиоэлектроники, 2017

Свидетельство о государственной регистрации КВ № 12097-968 ПР 14.12.2006

СОДЕРЖАНИЕ

РАДИОТЕХНИКА

- БОНДАРЕНКО И.Н., ВАСИЛЬЕВ Ю.С.** ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ АПЧ ИЗМЕРИТЕЛЬНОГО ГЕНЕРАТОРА ПРИ ФОРМИРОВАНИИ ПЕРВИЧНЫХ ИНФОРМАЦИОННЫХ СИГНАЛОВ РЕЗОНАТОРНОГО ДАТЧИКА 4

СИСТЕМЫ И ПРОЦЕССЫ УПРАВЛЕНИЯ

- ДУБА Т.В., КОЛОСОВА С.В.** ПІДХІД ДО РОЗВ'ЯЗАННЯ НЕСТАЦІОНАРНИХ ЗАДАЧ МАТЕМАТИЧНОЇ ФІЗИКИ З НЕЛОКАЛЬНИМИ УМОВАМИ 9
- ЧУБ І.А.** ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ ЕКОЛОГІЧНО НЕБЕЗПЕЧНИХ ОБ'ЄКТІВ 12

КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ

- САПОЖНИКОВ В.В., САПОЖНИКОВ Вл.В., ЕФАНОВ Д.В., ПИВОВАРОВ Д.В.** СПОСОБ ПОСТРОЕНИЯ СИСТЕМЫ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ НА ОСНОВЕ ЛОГИЧЕСКОГО ДОПОЛНЕНИЯ ПО РАВНОВЕСНОМУ КОДУ «1 ИЗ 5» 15
- ФИЛИППЕНКО И.В., КОНДРЮКОВ С.Э., КУЛАК Г.К.** ВСТРОЕННАЯ СИСТЕМА РЕГУЛИРОВАНИЯ ТЕМПЕРАТУРЫ В “УМНОМ ДОМЕ” 23
- БОЯДЖЯН А.Г., КОТЛЯРОВ А.С.** СИНТЕЗ КУБИТНЫХ ПОКРЫТИЙ ДЛЯ ЦИФРОВЫХ СИСТЕМ 28

КОМПЬЮТЕРНЫЕ НАУКИ

- ЗАЯЦЬ В.М., ЗАЯЦЬ М.М.** СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ – ОСНОВА СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЇХ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ 33

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- МАГЕРРАМОВ З.Т., АБДУЛЛАЕВ В.Г., МАГЕРРАМОВА А.З.** BIG DATA: ПРОБЛЕМЫ, МЕТОДЫ АНАЛИЗА, АЛГОРИТМЫ 42
- ГВОЗДИНСКИЙ А.Н., СЕРИК Е.Э.** ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ МЕТОДОВ УПРАВЛЕНИЯ СОВРЕМЕННЫМ ПРЕДПРИЯТИЕМ 53
- ЛИТВИНОВА Є.І., ЄМЕЛЬЯНОВ І.В., ХАХАНОВ І.В.** ХМАРНИЙ СЕРВІС ДЛЯ ТЕСТУВАННЯ І ВЕРИФІКАЦІЇ СИСТЕМ НА КРИСТАЛАХ 60
- РЕФЕРАТИ** 69
- ПРАВИЛА ОФОРМЛЕНИЯ РУКОПИСЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО ЖУРНАЛА** 72

ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ АПЧ ИЗМЕРИТЕЛЬНОГО ГЕНЕРАТОРА ПРИ ФОРМИРОВАНИИ ПЕРВИЧНЫХ ИНФОРМАЦИОННЫХ СИГНАЛОВ РЕЗОНАТОРНОГО ДАТЧИКА

БОНДАРЕНКО И.Н., ВАСИЛЬЕВ Ю.С.

Рассматриваются особенности функционирования системы регистрации информационных сигналов измерительного резонаторного датчика на основе АПЧ СВЧ генератора с меняющимися параметрами эталона. Приводятся уравнения, описывающие работу системы, а также расчетные соотношения. Проводится анализ режимов работы.

1. Введение

Резонансные измерительные преобразователи (РИП) – это основные элементы измерительных систем, в которых используются резонансные методы измерения [1]. Первичными информационными сигналами при этом являются значения резонансной частоты и добротности РИП, меняющиеся в известной зависимости от определяемых физических величин.

Наиболее интенсивно используются РИП в сканирующей микроволновой микроскопии (СММ), которая обеспечивает возможность исследования электрофизических характеристик поверхностей различных материалов, в том числе диэлектриков и полупроводников [1, 2].

Во многих работах по СММ в качестве системы формирования информационных сигналов используется система АПЧ измерительного СВЧ генератора по резонансному измерительному преобразователю с частотной модуляцией СВЧ генератора [3-7].

Однако в этих работах практически не анализируется ситуация, связанная с тем, что при изменении параметров РИП, который одновременно является частотным эталоном для системы АПЧ, изменяются также характеристики и режимы работы самой системы АПЧ и, соответственно, условия определения первичных информационных сигналов.

Целью исследования является определение и анализ возможных режимов работы системы формирования сигналов измерительной информации на основе АПЧ СВЧ генератора с меняющимися параметрами частотного эталона – измерительного резонаторного преобразователя.

2. Функциональная схема измерительной системы

Система формирования информационных сигналов с использованием АПЧ измерительного генератора по РИП с ЧМ СВЧ генератора может быть реализована в двух вариантах, отличающихся способом включения РИП (на проход или на отражение) [6, 7]. Функциональная схема системы с РИП, включенным на проход, приведена на рис. 1.

В качестве задающего СВЧ генератора обычно используется твердотельный генератор с возможностью электрической перестройки и модуляцией частоты, который охвачен системой автоматической ее настройки на резонансную частоту РИП (на схеме выделена пунктирной линией) [6, 7]. Работает рассматриваемая система АПЧ следующим образом. С генератора модулирующей частоты Ω 8 напряжение подается на элемент, управляющий частотой стабилизируемого генератора СВЧ 1 для осуществления неглубокой частотной модуляции. Частотно-модулированный сигнал через развязывающее устройство – ферритовый вентиль 3 поступает в резонатор 4, на выходе которого через ферритовый вентиль включен детектор 5. После детектора сигнал усиливается в усилителе модулирующей частоты 6 и поступает на фазовый детектор 7, куда в качестве опорного напряжения подается напряжение с генератора модулирующей частоты.

При средней частоте генератора, равной резонансной частоте резонатора, напряжение ошибки на выходе детектора отсутствует. Чем больше разность частот генератора и резонатора (вблизи f_0), тем больше амплитуда напряжения модулирующей частоты. При увеличении Δf переменное напряжение достигает максимума, а затем уменьшается. Фаза напряжения промежуточной частоты зависит от знака Δf . После сравнения сигналов с выхода усилителя частоты модуляции и от опорного генератора модулирующей частоты в фазовом детекторе на его выходе видно, что присутствует сигнал ошибки на постоянном токе, величина которого определяется величиной отклонения частоты генератора от частоты резонатора, а полярность – знаком отклонения. После усиления в УПТ 9 фильтрации в ФНЧ 10 сигнал поступает на элемент, управляющий частотой стабилизируемого СВЧ генератора, и перестраивает его до тех пор, пока частота генератора не станет равной частоте резонатора. Ответитель 2 предназначен ответвлять часть мощности генератора для измерения начального значения частоты с помощью частотомера 12. Генератор пилообразного напряжения 11 используется для предварительной настройки схемы.

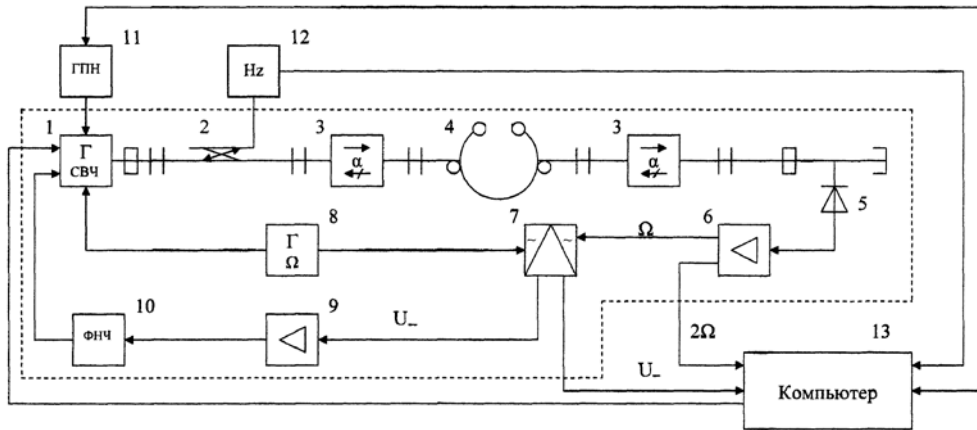


Рис. 1. Функциональная схема СВЧ системы на основе генератора с АПЧ по РИП с проходным включением резонатора

Сигнал, связанный с изменением частоты РИП, выделяется в виде сигнала ошибки после фазового детектора 7. Сигнал, связанный с изменением добротности РИП, выделяется в виде напряжения на СВЧ детекторе на частоте 2Ω , усиливается узкополосным усилителем и далее поступает на систему обработки.

Особенностью рассматриваемой схемы формирования сигналов является то, что автоподстройка частоты измерительного СВЧ генератора и работа в режиме совмещения его частоты с резонансной частотой РИП позволяют снизить влияние на информационный сигнал шумов, обусловленных частотными флуктуациями генератора. Использование информационного сигнала на частоте 2Ω позволяет уменьшить составляющую шума детектора, пропорциональную $1/f$, а селективное усиление – полосу усиливаемых шумов.

3. Основные соотношения

Уравнение, описывающее рассматриваемую систему АПЧ, выделенную пунктирной линией на рис. 1, можно представить в следующем виде [8]:

$$\Delta f_{\text{ст}}(t) = \Delta f_{\text{ст.с}}(t) + K_y \cdot K_T \cdot K_\phi \cdot K_d \cdot K_p \cdot [\Delta f_{\text{эт}}(t) - \Delta f_{\text{ст}}(t) - \Delta f_d(t)] \quad (1)$$

где K_y , K_T , K_ϕ , K_d , K_p – операторы управляющего элемента, УПТ, ФНЧ, дискриминатора, УПЧ; $\Delta f_{\text{ст.с}}(t)$ – изменение собственной частоты стабилизируемого генератора; $\Delta f_d(t)$ – средняя частота дискриминатора; $\Delta f_{\text{эт}}(t)$ – изменение частоты эталонного генератора; $\Delta f_{\text{ст.с}}(t)$ – изменение частоты стабилизируемого генератора в процессе подстройки.

Анализ уравнения (1) при допущении неустойчивости (вариации) характеристик эталонного сигнала приводит к выводу, что основным элементом, влияющим на изменения параметров системы авторегулирования, является оператор или функция передачи дискриминатора K_d , поскольку

операторы остальных элементов при этом функционально не меняются.

Таким образом, информацию о характере изменения параметров системы стабилизации частоты (в нашем случае – системы формирования информационных сигналов) при вариации характеристик эталонного сигнала (в нашем случае – изменениях резонансной частоты и добротности резонаторного измерительного преобразователя) можно получить при анализе соответствующих изменений функции $K_d(\Delta f_{\text{эт}}, Q_{\text{эт}})$ или $U_d(\Delta f_{\text{эт}}, Q_{\text{эт}})$.

Предполагается, что средняя частота генератора f_T изменяется по закону:

$$f_T = f_{0T} + \Delta f_0 \sin \Omega t, \quad (2)$$

где f_{0T} – средняя частота генератора; Δf_0 – амплитуда девиации частоты; Ω – угловая частота модуляции.

Выходное напряжение дискриминатора можно записать в следующем виде [9]:

$$u_d = \frac{\alpha P_0}{1 + (a_1 - a_0 \sin \Omega t)^2}, \quad (3)$$

где P_0 – мощность, подводимая к детектору при резонансе; α – коэффициент, зависящий от чувствительности детекторов; $a_0 = 2Q_0 \frac{\Delta f_0}{f_{0T}}$ – обобщенная амплитуда девиации частоты; Q_0 – добротность резонатора; $a_1 = 2Q_0 \frac{\Delta f}{f_{0p}}$ – обобщенная расстройка; Δf – отклонение частоты генератора от частоты резонатора; f_{0p} – частота настройки резонатора.

При равенстве средней частоты генератора и резонатора первая гармоника частоты модуляции равна нулю. При отклонении средней частоты генератора относительно частоты f_0 резонатора появляется первая гармоника частоты модуляции, фаза которой определяется знаком рас-

стройки. С помощью фильтра, включенного на выходе детектора, можно выделить только первую гармонику и использовать это напряжение расстройки для управления частотой стабилизируемого генератора.

Для анализа характера изменения амплитуды выходного сигнала в работе [9] предлагается использовать следующую формулу:

$$U_{\approx} = \alpha P_0 \left\{ \frac{\sqrt{2}}{3} \left[\frac{1}{1 + \left(a_1 + a_0 \frac{\sqrt{2}}{2} \right)^2} - \frac{1}{1 + \left(a_1 - a_0 \frac{\sqrt{2}}{2} \right)^2} \right] + \frac{1}{6} \left[\frac{1}{1 + (a_1 + a_0)^2} - \frac{1}{1 + (a_1 - a_0)^2} \right] \right\} \quad (4)$$

Крутизна характеристики дискриминатора при этом

$$K_d = \left(\frac{dU_{\approx}}{da_1} \right)_{a_1=0} = \frac{2}{3} \alpha P_0 a_0 \left[\frac{8}{(2+a_0)^2} + \frac{1}{(1+a_0)^2} \right] \quad (5)$$

Амплитуда выходного напряжения дискриминатора зависит не только от расстройки Δf , но также от индекса частотной модуляции.

4. Графические оценки

Нормированная на величину αP_0 характеристика изменения крутизны преобразования дискриминатора при изменении величины a_0 , построенная с использованием формулы (5), приведена на рис. 2.

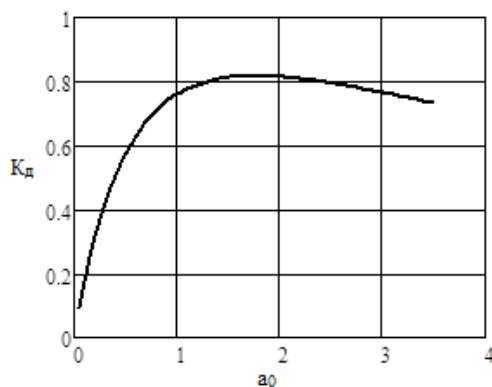


Рис. 2. Нормированная характеристика крутизны преобразования дискриминатора в зависимости от обобщенной амплитуды девиации частоты a_0

Можно видеть, что крутизна преобразования при росте девиации частоты достигает максимума, а потом падает. При этом значение девиации частоты значительно превышает полуширину полосы пропускания резонатора. Однако для нормальной работы системы автоподстройки частоты

(в нашем случае – системы регистрации информационных сигналов) необходимо, чтобы частота генератора не выходила за пределы полосы пропускания резонатора и была по возможности в зоне линейного участка его фазочастотной характеристики, т.е. величина a_0 не должна превышать 1.

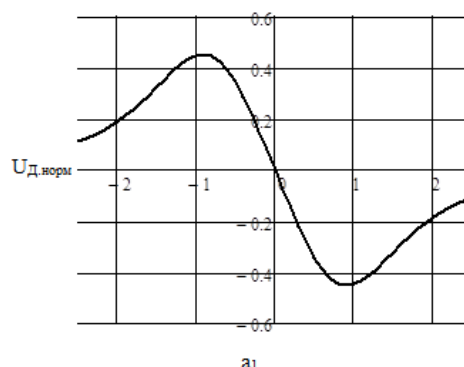


Рис. 3. Нормированная характеристика дискриминатора ($a_0 = 1$)

На рис. 3 приведена нормированная на величину αP_0 характеристика дискриминатора, вычисленная по формуле (4) при $a_0 = 1$. Переход характеристики через ноль соответствует изменению фазы на 180° . Как видно из рис.3, ширина характеристики равна 2. Это соответствует полосе пропускания резонатора.

Рассмотрим, как будет меняться величина нормированного выходного сигнала дискриминатора $U_{д,норм}$ при добротности резонатора 10^3 , рабочей частоте 10 ГГц и $a_0 = 0,5; 0,1; 0,02$. На рис. 4 приведены графики, полученные с помощью формулы (4).

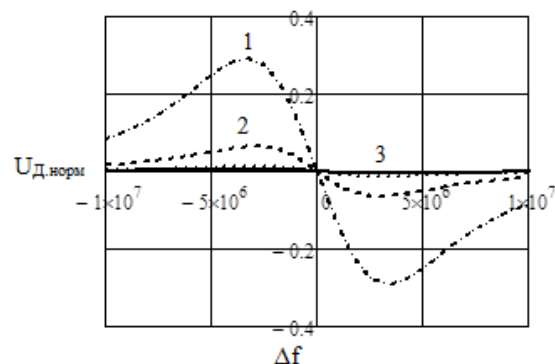


Рис. 4. Зависимости нормированных значений выходного сигнала дискриминатора от величины частотной расстройки при $Q_p = 10^3$, $f = 10$ ГГц, $a_0 = 0,5$ (1); $0,1$ (2); $0,02$ (3)

Графики, приведенные на рис. 4, показывают сильную зависимость выходного сигнала дискриминатора от величины девиации частоты. Следовательно, для получения максимального значения требуемого информационного сигнала

необходимо, чтобы девиация частоты измерительного генератора была сравнима с полушириной полосы пропускания резонатора.

При использовании резонатора дискриминатора в качестве РИП при взаимодействии с исследуемым объектом в зависимости от его свойств может происходить изменение не только резонансной частоты резонатора, но и его добротности. На рис. 5 приведены характеристики изменения величины выходного сигнала дискриминатора при различных значениях его добротности.

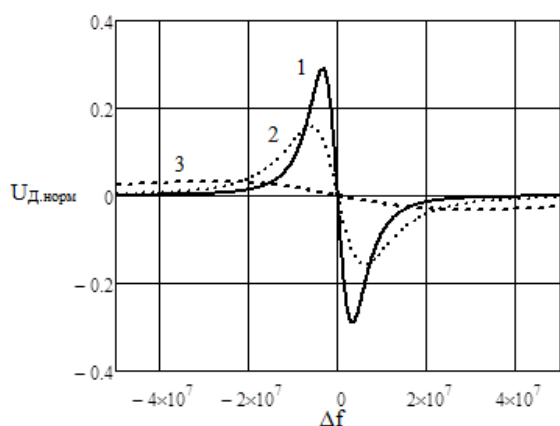


Рис. 5. Зависимости нормированных значений выходного сигнала дискриминатора от величины частотной расстройки при $f = 10$ ГГц, $a_0 = 0,5$, $Q_p = 10^3$ (1); $5 \cdot 10^2$ (2); 10^2 (3)

Из графиков (см. рис. 5) видно, что при изменении добротности резонатора сильно меняется не только амплитуда выходного сигнала дискриминатора, но и крутизна фазочастотной характеристики.

Графики изменения крутизны характеристики дискриминатора при изменении добротности резонатора приведены на рис. 6.

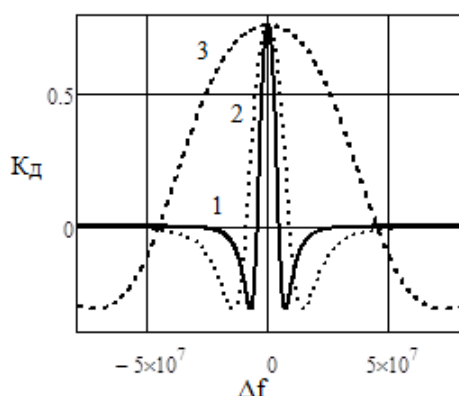


Рис. 6. Зависимости изменения крутизны характеристики дискриминатора от величины частотной расстройки при $Q_p = 10^3$ (1); $5 \cdot 10^2$ (2); 10^2 (3)

Зависимости изменения крутизны характеристик дискриминатора в положительной полуплоскости (см. рис. 6) практически повторяют форму

резонансных кривых при соответствующих значениях добротности.

Обычно для обеспечения эффективной работы системы АПЧ на основе дискриминатора с ЧМ СВЧ генератора величину обобщенной амплитуды девиации частоты a_0 выбирают значительно меньшей 1, т.е. абсолютное значение девиации частоты значительно меньше полуширины полосы пропускания резонаторного эталона. Это позволяет использовать только линейный участок фазочастотной характеристики и обеспечить оптимальный режим работы фазового детектора 7 (см. рис. 1). Малое значение сигнала на выходе детектора компенсируется дополнительным его усилением. При этом выбор величины девиации частоты привязан к максимально возможному значению добротности и не предполагает изменения полосы пропускания резонаторного эталона вследствие ее изменения, как это имеет место в нашем случае.

При использовании рассматриваемой системы АПЧ в качестве системы регистрации первичных информационных сигналов РИП, включаемого вместо резонаторного эталона, изменение (уменьшение) его добротности приводит, во-первых, к существенному уменьшению амплитуды выходного сигнала дискриминатора (см. рис. 5 и 6) и, соответственно, величин сигналов на частотах Ω и 2Ω , во-вторых - к изменению в худшую сторону возможностей системы АПЧ по отслеживанию изменений частоты РИП в процессе измерений. При этом будет уменьшаться отношение сигнал/шум информационных сигналов и увеличиваться погрешность измерения изменений частоты и добротности РИП. Некоторого улучшения ситуации можно достичь путем увеличения значения девиации частоты (см. рис. 2, 3 и 4), однако отмеченные факторы все равно будут влиять на работу системы регистрации и должны учитываться при ее эксплуатации.

5. Выводы

Таким образом, проведенное качественное рассмотрение возможных режимов работы системы АПЧ на основе дискриминатора с ЧМ измерительного СВЧ генератора как системы регистрации первичных информационных сигналов РИП показывает, что при значительных изменениях добротности РИП в процессе измерений происходит изменение характеристик системы АПЧ. Это, в свою очередь, ведет к дополнительным погрешностям при измерении изменений частоты и добротности РИП.

Для устранения выявленных факторов необходимо количественно оценить их влияние и выработать методы их учета и компенсации при измерениях и обработке сигналов.

Литература: 1. *Chen L.F., Ong C.K., Neo C.P., Varadan V.V., Varadan V.K.* Microwave Electronics. Measurements and Materials Characterization. John Wiley & Sons, Ltd., 2004. 537p. 2. *Rosner B., Van der Weide D.W.* High-frequency near-field microscopy // Review of Scientific Instruments, 2002. V. 73, N7. P. 2505-2525. 3. *Steinhauer D.E., Vlahacos C.P., Dutta S.K., Feenstra B.J., Wellstood F.S., Anlage S.M.* Quantitative imaging of sheet resistance with a scanning near-field microwave microscope // Appl. Phys. Lett., 1998. V. 72, N7. P. 861-863. 4. *Anlage S.M., Steinhauer D.E., Vlahacos C.P., Feenstra B.J., Thanawalla A.S., Wensheng Hu, Dutta S.K., Wellstood F.S.* Superconducting material diagnostics using a scanning near-field microwave microscope // IEEE Trans. on Appl. Superconductivity, 1999. Vol. 9, N2. P. 4127-4132. 5. *Imtiaz Atif, Anlage S.M.* Effect of tip geometry on contrast and spatial resolution of near-field microwave microscope // Journ. of Appl. Phys., 2006, 100, 044304. 6. *Бондаренко И.Н., Гордиенко Ю.Е., Ларкин С.Ю.* Формирование сигналов сканирования в микроволновой микроскопии с резонаторным микрозондом // Радиотехника. 2009. Вып. 158. С. 59-67. 7. *Бондаренко И.Н., Гордиенко Ю.Е., Ларкин С.Ю.* Применение системы АПЧ измерительного генератора в микроволновой сканирующей микроскопии // Радиоэлектроника и информатика. 2009. № 3. С. 3-6. 8. *Каганов В.И.* Системы автоматического регулирования в радиопередатчиках. М.: Связь, 1969. 232с. 9. *Бычков С.И., Буренин Н.И., Сафаров Р.Т.* Стабилизация частоты генераторов СВЧ. М.: Советское радио, 1962. 376с.

Transliterated bibliography:

1. *Chen L.F., Ong C.K., Neo C.P., Varadan V.V., Varadan V.K.* Microwave Electronics. Measurements and Materials Characterization. John Wiley & Sons, Ltd., 2004. 537p.
 2. *Rosner B., Van der Weide D.W.* High-frequency near-field microscopy // Review of Scientific Instruments, 2002, v. 73, N7, p. 2505-2525.
 3. *Steinhauer D.E., Vlahacos C.P., Dutta S.K., Feenstra B.J., Wellstood F.S., Anlage S.M.* Quantitative imaging of sheet resistance with a scanning near-field microwave microscope // Appl. Phys. Lett., 1998, v. 72, N7, pp. 861-863.
 4. *Anlage S.M., Steinhauer D.E., Vlahacos C.P., Feenstra B.J., Thanawalla A.S., Wensheng Hu, Dutta S.K., Wellstood F.S.* Superconducting material diagnostics using a

scanning near-field microwave microscope // IEEE Trans. on Appl. Superconductivity, 1999, vol. 9, N2, pp. 4127-4132.

5. *Imtiaz Atif, Anlage S.M.* Effect of tip geometry on contrast and spatial resolution of near-field microwave microscope // Journ. of Appl. Phys., 2006, 100, 044304.

6. *Bondarenko I.N., Gordienko Yu.E., Larkin S.Yu.* Формирование сигналов сканирования в микро-волновой микроскопии с резонаторным микрозондом // Радиотехника. Vseukr. mezhved. nauch.-tekhn. sb., 2009, vyp. 158, s. 59-67.

7. *Bondarenko I.N., Gordienko Yu.E., Larkin S.Yu.* Применение системы APCh измерительного генератора в микроволновой сканирующей микроскопии // Радиоэлектроника и информатика. 2009. № 3. с. 3-6.

8. *Kaganov V.I.* Системы автоматического регулирования в радиопередатчиках. М.: Связь, 1969. 232 с.

9. *Bychkov S.I., Burenin N.I., Safarov R.T.* Стабилизация частоты генераторов СВЧ. М.: Sovetskoe radio, 1962. 376 с.

Поступила в редколлегию 11.06.2017

Рецензент: д-р физ.-мат. наук, проф. Панченко А.Ю.

Бондаренко Игорь Николаевич, д-р физ.-мат. наук, профессор, зав. кафедрой микроэлектроники, электронных приборов и устройств ХНУРЭ. Научные интересы: техника СВЧ, бесконтактные методы диагностики, криогенная радиофизика. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. (057)702-13-62.

Васильев Юрий Сергеевич, ассистент кафедры микроэлектроники, электронных приборов и устройств ХНУРЭ. Научные интересы: техника СВЧ, бесконтактные методы диагностики. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. (057)702-13-62.

Bondarenko Igor Nikolaevich, Doctor of Physical and Mathematical Sciences, Professor, Head of Department of Microelectronics, Electronic Devices and Devices of KNURE. Scientific interests: microwave technology, contactless diagnostic methods, cryogenic radiophysics. Address: Ukraine, 61166, Kharkov, Science, 14, tel. (057) 702-13-62.

Vasiliev Yuri Sergeevich, assistant of the department of microelectronics, electronic devices and devices of KNURE. Scientific interests: microwave technology, unconstrained diagnostic methods. Address: Ukraine, 61166, Kharkov, Science, 14, tel. (057) 702-13-62.

СИСТЕМЫ И ПРОЦЕССЫ УПРАВЛЕНИЯ

УДК517.95:519.633

ПІДХІД ДО РОЗВ'ЯЗАННЯ НЕСТАЦІОНАРНИХ ЗАДАЧ МАТЕМАТИЧНОЇ ФІЗИКИ З НЕЛОКАЛЬНИМИ УМОВАМИ ДУБА Т.В., КОЛОСОВА С.В.

Розглядається застосування проєкційного методу Бубнова-Гальборкіна до початково-крайової задачі для нестационарного рівняння теплопровідності з нелокальною умовою. Для таких задач важливим є питання про вибір координатних функцій, на яке ми намагаємося дати відповідь.

Вступ

Актуальність дослідження. Велика кількість явищ у різних областях науки і техніки достатньо повно може бути описана за допомогою диференціальних рівнянь у частинних похідних, точні розв'язки яких вдається отримати для досить вузького класу задач. У даній роботі розглядається застосування проєкційного методу Бубнова-Гальборкіна до початково-крайової задачі теорії теплопровідності з нелокальною (некласичною) умовою (у нелокальних граничних умовах задається зв'язок значень розв'язку та його похідних у різних точках граничних та внутрішніх многовидів). Такі нелокальні умови зустрічаються досить часто, наприклад, у задачах, що описують процес дифузії частинок у турбулентній плазмі, у процесах розповсюдження тепла у тонкому нагрітому стержні, якщо задано закон зміни загальної кількості тепла стержня, у економічних задачах з рівнянням грошових накопичень тощо.

Метою даної роботи є розробка методу чисельного аналізу для початково-крайової задачі теорії теплопровідності з нелокальною умовою. Застосування проєкційного методу Бубнова-Гальборкіна дозволяє отримати розв'язок у аналітичному вигляді, що при проведенні обчислювальних експериментів з різними даними може допомогти зацікавленим структурам роботи прогнозу та розробляти стратегію своєї поведінки.

1. Постановка задачі

Розглянемо початково-крайову задачу:

$$a^2 \frac{\partial^2 u}{\partial x^2} - h(u - u_0) + f(x, t) = \frac{\partial u}{\partial t}, \quad (1)$$

$$\forall x \in (0, 1), \quad t > 0,$$

$$u|_{t=0} = \varphi(x), \quad u|_{x=0} = 0, \quad \int_0^1 u(x, t) dx = K(t), \quad (2)$$

тут $u(x, t)$ – температура точки x у момент часу t ; $\varphi(x)$ – розподіл температур в точках стержня в початковий момент часу $t = 0$; $a^2 = \text{const}$ – коефіцієнт теплопровідності; $h = \frac{\alpha}{c\rho} = \text{const}$; ρ – густина маси; c – питома теплоємність; α – коефіцієнт теплообміну між поверхнею стержня та навколишнім середовищем з температурою u_0 ; $K(t)$ – загальна кількість тепла стержня у момент часу t [1, 2]. Шукаємо розв'язок задачі (1), (2) $u(x, t) \in C(\bar{B})$, $\bar{B} = \{(x, t) | 0 \leq x \leq 1, t \geq 0\}$. При цьому повинні виконуватися умови узгодження

$$\varphi(0) = 0, \quad \int_0^1 \varphi(x) dx = K(0). \quad (3)$$

У задачі (1), (2) зробимо заміну:

$$u(x, t) = v(x, t) + W(x, t), \quad (4)$$

$$W(x, t) = \frac{2x}{l^2} K(t), \quad (5)$$

що приведе до наступної задачі з однорідними крайовими умовами:

$$a^2 \frac{\partial^2 v}{\partial x^2} - h(v - u_0) + F(x, t) = \frac{\partial v}{\partial t}, \quad (6)$$

$$\forall x \in (0, 1), \quad t > 0,$$

$$v|_{t=0} = \bar{\varphi}(x), \quad v|_{x=0} = 0, \quad \int_0^1 v(x, t) dx = 0, \quad (7)$$

$$\bar{\varphi}(x) = \varphi(x) - \frac{2x}{l^2} K(0),$$

де

$$F(x, t) = f(x, t) - \frac{2hx}{l^2} K(t) - \frac{2x}{l^2} K'(t).$$

2. Побудова алгоритму розв'язання задачі за допомогою методу Бубнова-Гальборкіна

Згідно з методом Бубнова-Гальборкіна [3, 4] наближений розв'язок задачі (6), (7) шукаємо у вигляді

$$v_n(x, t) = \sum_{k=1}^n a_k(t) \varphi_k(x), \quad (8)$$

де $\varphi_1, \varphi_2, \dots, \varphi_n$ – координатні функції (кожна $\varphi_k(x) \in D(A)$ – області визначення оператора A , відповідного до задачі (6), (7) операторного рівняння, при будь-якому n $\varphi_1, \dots, \varphi_n$ є лінійно незалежними, послідовність $\varphi_1, \dots, \varphi_n$ є повною у гільбертовому просторі $L_2(0, 1)$; тут

$$Av = a^2 \frac{\partial^2 v}{\partial x^2} - h(v - u_0), \quad D(A) = \{v | v \in C^2(B),$$

$$v \in C(\bar{B}), \quad v \in L^2(0, 1) \forall t \geq 0, \quad v(0, t) = 0,$$

$$\int_0^1 v(x, t) dx = 0 \quad \forall t \geq 0\}.$$

Прирівнюючи нулеві скалярний добуток відхилу рівняння задачі (6), (7) до кожної з координатних функцій $\varphi_1, \dots, \varphi_n$, маємо:

$$\begin{aligned} & (a^2 \frac{\partial^2 v_n}{\partial x^2} - h(v_n - u_0) + F(x, t) - \frac{\partial v_n}{\partial t}, \varphi_j) = \\ & = a^2 (\frac{\partial^2}{\partial x^2} \sum_{k=1}^n a_k(t) \varphi_k, \varphi_j) - h(\sum_{k=1}^n a_k(t) \varphi_k, \varphi_j) - \\ & - (\frac{\partial}{\partial t} \sum_{k=1}^n a_k(t) \varphi_k, \varphi_j) + (F + hu_0, \varphi_j) = 0, \quad j=1, \dots, n. \end{aligned}$$

Звідси отримуємо наступну систему лінійних звичайних диференціальних рівнянь зі сталими коефіцієнтами для відшукування функцій $a_k(t)$, $k=1, \dots, n$:

$$\begin{aligned} & \sum_{k=1}^n a'_k(t) (\varphi_k, \varphi_j) + \sum_{k=1}^n a_k(t) [h(\varphi_k, \varphi_j) - \\ & - a^2 (\varphi'_k, \varphi_j)] = (F, \varphi_j) + \\ & + h(u_0, \varphi_j), \quad j=1, \dots, n. \end{aligned} \quad (9)$$

Відповідні початкові умови для системи (9) отримуємо таким чином. Початкову функцію $\bar{\varphi}(x)$ зображуємо у вигляді n -ї часткової суми ряду за координатними функціями $\varphi_k(x)$, а саме

$$\bar{\varphi}(x) = \sum_{k=1}^n \beta_k \varphi_k(x), \text{ та добираємо сталі } \beta_k, k=1, \dots, n,$$

$$\text{з умови } \|\bar{\varphi}(x) - \sum_{k=1}^n \beta_k \varphi_k(x)\|_{L_2(0,1)}^2 \rightarrow \min.$$

Ця умова приводить до наступної системи рівнянь відносно β_k :

$$\sum_{k=1}^n \beta_k (\varphi_k, \varphi_j) = (\bar{\varphi}, \varphi_j), \quad j=1, \dots, n. \quad (10)$$

Отже, початкову умову $v(x, 0) = \bar{\varphi}(x)$ записуємо у вигляді $\sum_{k=1}^n a_k(0) \varphi_k(x) = \sum_{k=1}^n \beta_k \varphi_k(x)$, що приводить до співвідношень

$$a_k(0) = \beta_k, \quad k=1, \dots, n. \quad (11)$$

Таким чином, отримали для відшукування функцій $a_k(t)$, $k=1, \dots, n$, задачу Коші (9), (11).

3. Про вибір координатних функцій

У [5] за координатні функції для розв'язання рівняння $Au = f$, де A – додатно визначений оператор, пропонується взяти систему власних елементів (якщо вона повна) оператора \bar{A} , схожого та спорідненого з оператором A . Введемо у розгляд оператор \bar{A} :

$$\bar{A}v = a^2 \frac{\partial^2 v}{\partial x^2}, \quad D(\bar{A}) = \{v \mid v(x, t) \in L_2(0, 1)\}$$

$$\forall t \geq 0, v(x, t) \in C^2(B), v(x, t) \in C(\bar{B}),$$

$$v(0, t) = 0, \int_0^1 v(x, t) dx = 0 \quad \forall t \geq 0\}.$$

Відшукаємо власні числа та власні функції оператора \bar{A} . Маємо задачу

$$\phi''(x) + \lambda \phi(x) = 0 \quad \forall x \in (0, 1), \quad \phi(0) = 0, \int_0^1 \phi(x) dx = 0,$$

$$\text{звідки } \lambda_k = \frac{4\pi^2 k^2}{1^2}, \quad \phi_k(x) = \sin \frac{2\pi kx}{1}, \quad k=1, 2, 3, \dots$$

Тому що оператори A та \bar{A} є схожими та спорідненими, за координатні функції пропонуємо взяти $\varphi_k(x) = \sin \frac{2\pi kx}{1}$, $k=1, 2, \dots$

4. Обчислювальний експеримент

Добираючи для проведення обчислювальних експериментів дані задачі (1), (2), треба мати на увазі умови узгодження (3). Надалі вважаємо $l=1$. Покладемо $\varphi(x) = -3x^2 + 6x$, при цьому виконується перша умова узгодженості $\varphi(0) = 0$.

Крім того, у початковий момент часу $t=0$ маємо загальну кількість тепла стержня

$$K(0) = \int_0^1 (-3x^2 + 6x) dx = 2. \text{ Наведемо деякі можливі}$$

вигляди функції $K(t)$, яка повинна задовольняти

$$\text{другу умову узгодження } \int_0^1 \varphi(x) dx = K(0):$$

$$K(t) = \{2, 2(t+1), 2(t^2+1), 2(t+1)^2, \dots\}.$$

Обчислювальний експеримент проведено для $h=0.0155$, $a^2=0.16$ (ці значення відповідають матеріалу стержня – чавун), $K(t)=2(t+1)$, $u_0=0$, $f(x, t)=1$. На рис. 1 – 3 наведені графіки функції $u(x, t)$ у моменти часу $t=0, 1, 3$.

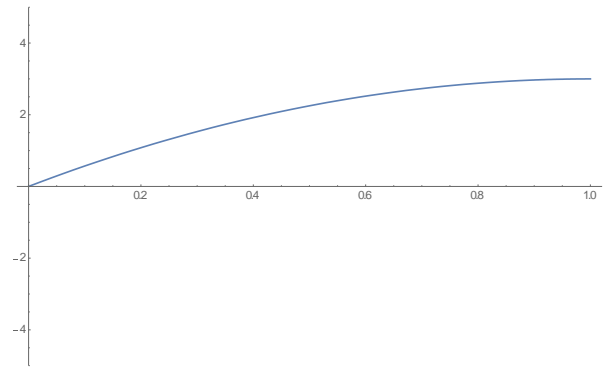


Рис. 1. Графік функції $u(x, t)$ у момент часу $t=0$

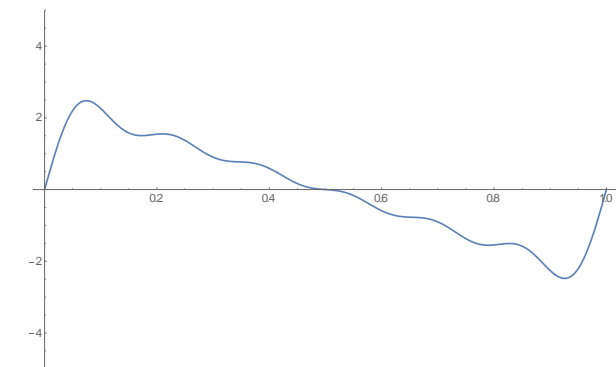


Рис. 2. Графік функції $u(x, t)$ у момент часу $t=1$

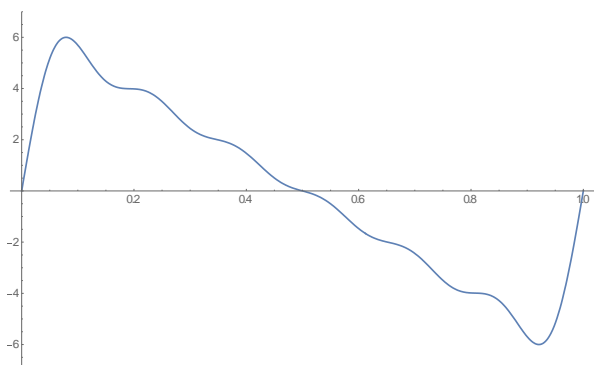


Рис. 3. Графік функції $u(x, t)$ у момент часу $t = 3$

Висновки

Вперше запропоновано для розв'язання початково-крайових задач математичної фізики з нелокальними крайовими умовами застосування проєкційного методу Бубнова-Гальоркіна, що надає можливість отримати результати у аналітичному вигляді. Надано пропозиції про вибір координатних функцій для задач з нелокальними умовами. Розроблений метод дозволяє проводити математичне моделювання багатьох технологічних процесів. Цим і визначається наукова новизна та практична значущість роботи.

Література: 1. *Ионкин Н.И.* Решение одной краевой задачи теории теплопроводности с неклассическими краевыми условиями // Дифференциальные уравнения. 1977. Т. 13. № 2. С. 294-304. 2. *Будак Б.М., Самарский А.А., Тихонов А.Н.* Сборник задач по математической физике. М.: Физматлит, 2004. 688 с. 3. *Михлин С.Г.* Вариационные методы в математической физике. М.: Наука, 1970. 512 с. 4. *Савула Я.Г.* Числовой анализ задач математической физики вариационными методами. Львів: видавничий центр ЛНУ ім. І. Франка, 2004. 221 с. 5. *Михлин С.Г.* Численная реализация вариационных методов. М.: Наука, 1966. 432 с.

Transliterated bibliography:

1. *Ionkin N.I.* Reshenie odnojkraevoy zadachi teorii teploprovodnosti s neklassicheskimi kraevymi usloviyami // Differentsial' nyeuravneniya. 1977. T. 13. № 2. S. 294-304.
2. *Budak B.M., Samarskiy A.A., Tihonov A.N.* Sbornik zadach po matematicheskoy fizike. M.: Fizmatlit, 2004. 688 s.
3. *Mihlin S.G.* Variatsionnye metody v matematicheskoy fizike. M.: Nauka, 1970. 512 s.
4. *Savula Ya. H.* Chyslovyy analiz zadach matematychnoy fizyky variatsiynymy metodamy. L'viv: vydavnychytsentr LNU im. I. Franka, 2004. 221 s.
5. *Mihlin S.G.* Chislennaya realizatsiya variatsionnykh metodov. M.: Nauka, 1966. 432 s.

Надійшла до редколегії 18.04.2017

Рецензент: д-р физ.-мат. наук, проф. Колосов А.И.

Дуба Тетяна Вікторівна, ст. гр. ПМ-13-1 ф-ту інформаційно-аналітичних технологій і менеджменту ХНУРЕ. Наукові інтереси: математична фізика, математичне моделювання, чисельні методи. Захоплення та хобі: мистецтво та література. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. (096) 2425488.

Колосова Світлана Василівна, канд. фіз.-мат. наук, проф. кафедри прикладної математики ХНУРЕ. Наукові інтереси: математичне моделювання, чисельні методи математичної фізики. Захоплення на хобі: театр, мистецтво та література. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. (057) 7021423.

Duba Tetyana Victorivna, Student, gr. PM-13-1 Faculty of information and analytical technologies and management, Kharkiv National University of Radioelectronics. Scientific interests: mathematical physics, mathematical modeling, numerical methods. Hobbies and Hobbies: Arts and Literature. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14, tel. (096) 2425488.

Kolossova Svetlana Vasyliyva, PhD, prof., Department of Applied Mathematics, KNURE. Scientific interests: mathematical modeling, numerical methods of mathematical physics. Hobbies: theater, art and literature. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14, tel. (057) 7021423.

ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ ЕКОЛОГІЧНО НЕБЕЗПЕЧНИХ ОБ'ЄКТІВ ЧУБ І.А.

Розглядається імплементація одного з основних методів зменшення антропогенного впливу екологічно небезпечних об'єктів на навколишнє середовище – раціонального розміщення джерел забруднення. Описується оптимізаційна математична модель розміщення пожежонебезпечних об'єктів з мінімізацією величини концентрації забруднюючих домішок у множині точок контролю та ідеологія чисельного методу розв'язання задачі. Проводяться чисельні експерименти.

1. Вступ

Захист навколишнього середовища є складовою частиною концепції сталого розвитку суспільства, що забезпечує потреби сучасників без шкоди для майбутніх поколінь. Серед активних методів захисту навколишнього середовища на основі технологічних рішень по створенню і застосуванню ресурсозберігаючих і маловідходних технологій, а також різних природоохоронних технологій, технічних систем і пристроїв [1] велике значення має раціональне розміщення джерел забруднення. Останній підхід відносять до методів боротьби із забрудненням, яке утворилося, що дозволяє звести до мінімуму або виключити утворення шкідливих речовин в наступних екологічних ситуаціях. Такі заходи реалізуються на етапі створення планувальних рішень, тому носять прогнозний характер та вимагають наявності арсеналу оптимізаційних моделей, що дозволяють визначити оптимальне розміщення екологічно небезпечних об'єктів з урахуванням кліматичних факторів та суміжних рекреаційних зон.

Значний внесок в екологічну безпеку території вносять пожежі, що внаслідок великої кількості викидів забруднюючих аерозольних та газоподібних продуктів горіння в атмосферу порушують екологічний баланс. Таким чином, як специфічний клас екологічно небезпечних об'єктів розглядаються пожежонебезпечні об'єкти, пожежі на яких є джерелами забруднюючих аерозольних викидів в атмосферу.

Отже, метою дослідження є побудова оптимізаційної математичної моделі розміщення пожежонебезпечних об'єктів з мінімізацією величини концентрації забруднюючих домішок у множині точок контролю.

2. Виклад основного матеріалу

Побудуємо математичну модель оптимізаційної задачі розміщення пожежонебезпечних об'єктів. Нехай є замкнена обмежена багатокутна область $S_0 \subset \mathbb{R}^2$, задана координатами її вершин v_ψ , $\psi = \overline{1, \Psi}$ та скінченний набір S пожежонебезпечних об'єктів $S = \{S_i\}$, $i = \overline{1, N}$.

Якщо виключити з розгляду ландшафтні (лісові, степові тощо) пожежі, то їх площа набагато менша площі, на яку поширюються забруднюючі домішки. Тому в даній статті пожежонебезпечні об'єкти S_i вважаються точковими джерелами забруднюючої домішки (рис. 1).

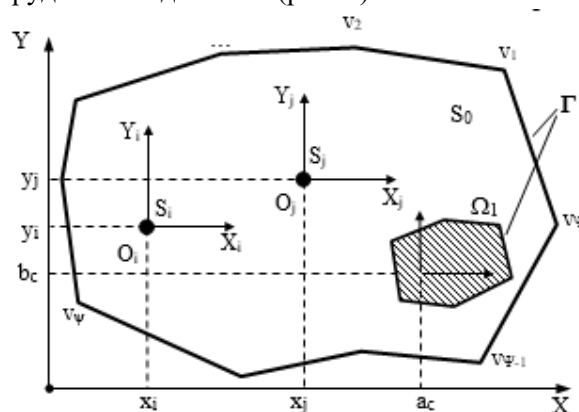


Рис. 1. Область S_0 з розміщеними об'єктами S_1, S_2 , зоною заборони Ω_1

В області S_0 виділяється багатозв'язна область розміщення Ω з границею Γ вигляду

$$\Omega = S_0 \setminus \bigcup_{c=1}^C \Omega_c,$$

де $\Omega_c(a_c, b_c)$ – область заборони; (a_c, b_c) – фіксовані параметри розміщення, $c = \overline{1, C}$.

Нехай також множина пожежонебезпечних об'єктів S_i є такою, що аерозольні викиди можливих пожеж на цих об'єктах породжують в області Ω зони забруднення T_i з границею Γ_i , $i = \overline{1, N}$. При знаходженні оптимального розміщення джерел викидів (пожежі) в області не можуть бути точно вказані значення кліматичних параметрів (напрямі і швидкість вітру, умови стратифікації тощо) в момент пожежі через невизначеність часу її виникнення. У цьому випадку найбільш ймовірний напрямок вітру і його швидкість моделюються розгою вітрів, характерною для даної місцевості, а форма області забруднення – це восьмикутник.

Кожна з вершин зони забруднення відповідає одному з напрямків частин світу. У цьому випадку об'єкт T_i задається набором $r = \{r_i^k\}$ відстаней від полюса об'єкта до його вершин v_i^k (x_i^k, y_i^k),

$k = \overline{1,8}$. Вектори \vec{r}_i^k орієнтовані по розі вітрів. Довжини векторів \vec{r}_i^k дорівнюють відстані від центру об'єкта S_i до точки на осі викиду, в якій досягається максимальна концентрація забруднюючої домішки в даному напрямку λ :

$$C(x,y)|_{(x,y) \in \Gamma_i} = \max C(x,y)|_{\lambda} = C_M|\lambda.$$

Така побудова області забруднення T_i гарантує, що за її межами концентрація домішки буде менша, ніж на границі Γ_i .

Тоді оптимізаційна задача розміщення N пожежонебезпечних об'єктів в області Ω формулюється так:

визначити таке розміщення пожежонебезпечних об'єктів S_i , $i = \overline{1,N}$, в області Ω , щоб максимальна сумарна концентрація аерозольних викидів можливих пожеж у скінченній множині точок контролю $P = \{p_\zeta\} = \{x_\zeta^p, y_\zeta^p\}$, $\zeta = \overline{1,\lambda}$, була мінімальною, тобто знайти

$$\min_{(x_i, y_i) \in D} \max_{(x_\zeta^p, y_\zeta^p) \in P} \sum_{i=1}^N C(x_\zeta^p, y_\zeta^p, x_i, y_i, G_i, Q). \quad (1)$$

Результатом розв'язання задачі (1) є вектор $u = (x_1, y_1, x_2, y_2, \dots, x_N, y_N)$ параметрів розміщення пожежонебезпечних об'єктів.

Дана задача допускає різні трактування щодо виду функції мети, однак загальним для цього класу задач є те, що область допустимих розв'язків D формується системою обмежень, яка включає геометричні обмеження на параметри розміщення об'єктів T_i і фізичні обмеження на характеристики результуючого поля приземних концентрацій аерозольних викидів пожежі в області Ω .

Крім того, система обмежень може містити додаткові (технологічні) умови.

Геометричні обмеження на розміщення пожежонебезпечних об'єктів в області Ω включають в себе умови належності множини об'єктів T_i , $i = \overline{1,N}$, області Ω , а також умови попарного взаємного неперетину об'єктів T_i :

$$\Phi_{i0}(x_i, y_i, m_0) \geq 0, \quad (2)$$

$$\Phi_{ij}(x_i, y_i, x_j, y_j) \geq 0, \quad i, j = \overline{1,N}, i \neq j, \quad (3)$$

де $\Phi_{i0}(x_i, y_i, m_0)$ – Φ -функція [2] об'єкта T_i і області $\overline{\Omega} = \text{cl}(R^2 \setminus \Omega)$; m_0 – метричні характеристики області Ω ; $\Phi_{ij}(x_i, y_i, x_j, y_j)$ – Φ -функція об'єктів T_i і T_j , $i, j = \overline{1,N}, i \neq j$.

В задачі (1) фізичним обмеженням на характер поля приземних концентрацій аерозольних викидів пожеж є умова:

$$C(x, y, u, G, Q)|_{(x,y) \in P} + C_\phi \leq \text{ГДК}, \quad (4)$$

де C – функція концентрації забруднюючих викидів, що досягається викидами з N джерел; C_ϕ – фонові концентрації, $u = (x_1, y_1, \dots, x_N, y_N)$, $G = (g_1, g_2, \dots, g_k)$ – вектор фізичних параметрів пожежі; $Q = (q_1, q_2, \dots, q_N)$ – вектор параметрів, що характеризують природно-кліматичні умови області S_0 .

Технологічні обмеження, що накладаються на розміщення пожежонебезпечних об'єктів в області Ω , представляються, як правило, набором максимальних (мінімальних) допустимих відстаней $L_{i\mu}^*$ між об'єктами розміщення S_i і елементами Σ_μ транспортних мереж, інженерних комунікацій тощо:

$$L_{i\mu} \geq (\leq) L_{i\mu}^*, \quad i = \overline{1,N}, \quad \mu = \overline{1,M}. \quad (5)$$

Відзначимо основні властивості оптимізаційної задачі (1), що впливають з її математичної постановки:

Властивість 1. Простір параметрів, в якому визначається екстремум функції мети задачі (1), має розмірність $2N$, де N – число пожежонебезпечних об'єктів розміщення: $D \subset R^{2N}$.

Властивість 2. Для формалізації обмежень (4) на розподіл поля концентрації забруднюючої домішки розв'язок крайової задачі математичної фізики представляється як функція параметрів u, G і Q .

Наведемо основні етапи методу розв'язання оптимізаційної задачі розміщення пожежонебезпечних об'єктів.

Через складність оптимізаційної задачі (1) – (5) не доводиться розраховувати на можливість отримання точного розв'язку. Тому пропонується метод пошуку раціональних рішень і їх перебір, в результаті якого визначається наближене значення локального екстремуму функції мети. Метод розв'язання оптимізаційної задачі (1) – (5) складається з таких основних етапів:

- визначення початкового варіанту розміщення пожежонебезпечних об'єктів S в області Ω ;
- пошук вектора u^* , що відповідає локальному мінімуму функції мети

$$F(u) = \max_{(x_\zeta^p, y_\zeta^p) \in P} \sum_{i=1}^N C(x_\zeta^p, y_\zeta^p, x_i, y_i, G_i, Q);$$

- перебір локальних мінімумів функції мети $F(u)$. При цьому виділяється рекордне значення функції мети і визначається відповідний йому вектор u^{**} параметрів розміщення об'єктів. Розглянемо розв'язання тестової задачі.

Нехай дана область Ω розміщення з кусково-лінійною границею Γ (модель території нафтобази, де Γ_s є границею санітарно-захисної зони) і п'ять пожежонебезпечних об'єктів S_i , $i = \overline{1,5}$. Як об'єкти S_i обрані резервуари з різними нафтопродуктами. Параметри об'єктів зведені в табл. 1. Характеристики метеорологічних умов в області Ω дані в табл. 2.

Таблиця 1. Характеристики пожежонебезпечних об'єктів

№ об'єкта, i	Пальна речовина	Діаметр резервуару, м	Висота резервуару, м	Площа горіння, m^2
1	Бензин	5	6	19,63
2	Бензин	5	6	19,63
3	Гас	4	5	12,56
4	Дизельне паливо	5	6	19,64
5	Мазут	6	8	28,27

Таблиця 2. Кліматичні умови в області Ω

Напрямок вітру	Пн	Пн-С	С	Пд-С	Пд	Пд-З	З	Пн-З
Середня швидкість вітру, м/с	3,9	4,2	4,4	4,2	3,3	3,2	3,3	3,9

Задача оптимізації розміщення пожежонебезпечних об'єктів в області Ω є такою: необхідно в області Ω знайти таке розміщення пожежонебезпечних об'єктів S_i , $i = \overline{1,5}$, щоб максимальне значення концентрації аерозольних викидів пожеж (в разі їх виникнення) на границі Γ_s санітарно-захисної зони було мінімальним:

$$\min_{(x_i, y_i) \in D} \max_{(x, y) \in \Gamma_s} \sum_{i=1}^N C(x, y, x_i, y_i, G_i, Q) \quad (6)$$

Розв'язання задачі (6) здійснюється наближеним ітераційним методом пошуку раціональних рішень. Графічне представлення результатів розрахунків дано на рис. 2.

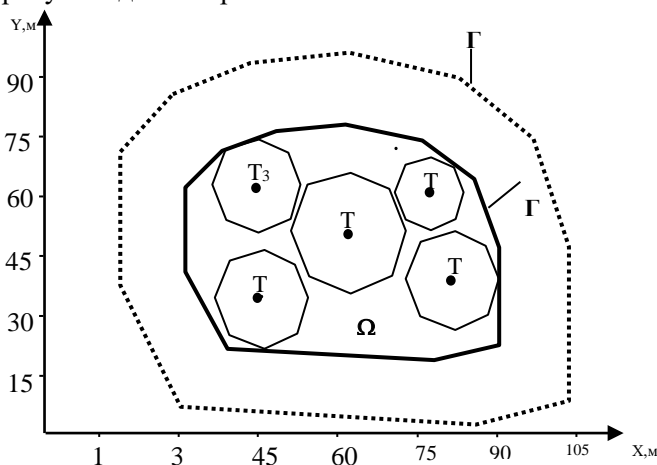


Рис. 2. Раціональне розміщення джерел забруднюючої домішки в області Ω

3. Висновки.

Проведено моделювання можливої форми зони забруднення земної поверхні викидами пожежі. Показано, що можлива зона забруднення моделюється восьмикутником, кожна з вершин якого відповідає одному з напрямків частин світу. Розрахунок ліній рівних концентрацій проводиться на основі середньорічного значення швидкості вітру регіону по кожному з напрямків.

Запропоновано метод розв'язання оптимізаційної задачі розміщення пожежонебезпечних об'єктів у заданій області з урахуванням впливу небезпечних чинників можливої пожежі на навколишнє середовище та кліматичних умов. Виконано чисельне дослідження задачі.

Література: 1. *Ветошкин А.Г.* Основы инженерной защиты окружающей среды. М.: Инфра-Инженерия, 2016. 456 с. 2. *Стойан Ю. Г., Яковлев С.В.* Математические модели и оптимизационные методы геометрического проектирования. К.: Наук. думка, 1986. 267 с. 3. *Новожилова М.В., Чуб І.А.* Условия размещения экологически опасного объекта с учетом изменения геометрических характеристик области возможного загрязнения // Научный вестник строительства. 2009. № 54. С. 240-244. 4. *Чуб І.А., Новожилова М.В., Андронов В.А.* Моделирование размещения пожароопасных объектов: минимизация загрязнения атмосферы. Харків: ІД ІНЖЕК. 2012. 133 с.

Transliterated bibliography:

1. *Vetoshkin A.G.* Osnovy inzhenernoy zaschityi okruzhayuschey sredy. M.: Infra-Inzheneriya, 2016. 456 s.
2. *Stoyan Yu. G., Yakovlev S.V.* Matematicheskie modeli i optimizatsionnyie metodyi geometricheskogo proektirovaniya. K.: Nauk. dumka, 1986. 267 s.
3. *Novozhilova M.V., Chub I.A.* Usloviya razmescheniya ekologicheskii opasnogo ob'ekta s uchotom izmeneniya geometricheskii harakteristik oblasti vozmozhnogo zagryazneniya // Naukoviy vIsnik budIvnitstva. 2009. # 54. S. 240-244.
4. *Chub I.A., Novozhilova M.V., Andronov V.A.* Modelirovanie razmescheniya pozharoopasnyih ob'ektov: minimizatsiya zagryazneniya atmosferyi. HarkIv: ID INZhEK. 2012. 133 s.

Надійшла до редколегії 20.01.2017

Рецензент: д-р фіз.-мат. наук, проф. Сізова Н.Д.

Чуб Ігор Андрійович, д-р техн. наук, професор, начальник кафедри пожежної профілактики в населених пунктах Національного університету цивільного захисту України, м. Харків. Наукові інтереси: математичне моделювання, оптимізація структури та функціонування складних організаційно-технічних систем. Адреса: Україна, 61000, Харків, вул. Чернишевська, 94, тел. (057) 707-34-13.

Chub Igor Andreevich, Dr. of Tech. Sciences, professor, head of department, fire prevention in settlements department, National University of Civil Defense, Ukraine, Kharkiv. Scientific interests: mathematical modeling and optimization of the structure and functioning of complex organizational and technical systems. Address: Ukraine, 61000, Kharkov, street Chernyshevskaya, 94, tel. (057) 707-34-13.

КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ

УДК 681.518.5:004.052.32

СПОСОБ ПОСТРОЕНИЯ СИСТЕМЫ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ НА ОСНОВЕ ЛОГИЧЕСКОГО ДОПОЛНЕНИЯ ПО РАВНОВЕСНОМУ КОДУ «1 ИЗ 5»

САПОЖНИКОВ В.В., САПОЖНИКОВ Вл.В.,
ЕФАНОВ Д.В., ПИВОВАРОВ Д.В.

Предлагается разработанный авторами статьи способ построения систем функционального контроля логических схем автоматики и вычислительной техники на основе метода логического дополнения по равновесному коду «1 из 5». Приводится общий подход к построению систем функционального контроля методом логического дополнения. Описываются базовые структуры систем функционального контроля на основе логического дополнения по равновесному коду «1 из 5». Даются формулы вычисления контрольных функций дополнения, позволяющие решить задачу формирования полного множества тестовых комбинаций для тестера кода «1 из 5». Отмечаются особенности реализации систем функционального контроля по разработанному способу с условием обеспечения свойства полной самопроверяемости структуры. Даются результаты экспериментов по применению разработанного способа при организации систем диагностирования контрольных комбинационных схем из набора MCNC Benchmarks.

1. Введение

Системы функционального контроля предназначены для технического диагностирования устройств в режиме реального времени в процессе выполнения ими своих функций [1, 2]. При построении системы функционального контроля исходное логическое устройство $F(x)$ (объект диагностирования), вычисляющее значения ряда рабочих функций f_1, f_2, \dots, f_m , дополняется специализированной схемой контроля, осуществляющей оценку правильности вычислений. В процессе эксплуатации в устройстве $F(x)$ возникают неисправности, которые могут проявляться на рабочих выходах в виде искажений вычисляемых значений. Задачей схемы контроля является идентификация этих искажений. Таким образом, реализуется стратегия косвенной оценки технического состояния объекта диагностирования по результатам вычислений [3, 4].

При организации систем функционального контроля используются помехоустойчивые коды с небольшой избыточностью, ориентированные на обнаружение ошибок [5, 6]. К ним относятся коды с суммированием (коды Бергера [7]) и их модификации [8 – 12], а также равновесные коды [13] и их модификации [14, 15]. Небольшая избыточность кода обуславливает приемлемые для разработчика системы функционального контроля характеристики структурной избыточности, обнаруживающей способности, энергопо-

требления и пр. Так, структурная избыточность системы диагностирования не должна превышать избыточности системы дублирования.

Применение помехоустойчивых кодов для построения систем функционального контроля описано в большом количестве работ, например, в [16 – 20]. Известно два основных подхода к организации систем функционального контроля: на основе метода вычисления контрольных разрядов разделимого кода и на основе метода логического дополнения [21].

При реализации системы функционального контроля на основе метода вычисления контрольных разрядов выходы устройства $F(x)$ отождествляются с информационным вектором $\langle f_m f_{m-1} \dots f_2 f_1 \rangle$, а в схеме контроля выделяется блок контрольной логики $G(x)$, вычисляющий контрольные функции g_1, g_2, \dots, g_k , объединяемые в контрольный вектор $\langle g_k g_{k-1} \dots g_2 g_1 \rangle$. Для сравнения значений информационного и контрольного векторов используется самопроверяемый тестер кода TSC [22]. При таком подходе характеристики системы диагностирования полностью определяются выбранным на этапе ее проектирования разделимым кодом [20]. Следует отметить, что реализация блока контрольной логики является единственной при использовании метода вычисления контрольных разрядов.

Применение метода логического дополнения подразумевает преобразование значений вектора $\langle f_m f_{m-1} \dots f_2 f_1 \rangle$ с помощью блока логического дополнения в вектор выбранного кода $\langle h_m h_{m-1} \dots h_2 h_1 \rangle$ [23]. Это позволяет минимизировать структурную избыточность за счет упрощения блока контрольной логики $G(x)$. Кроме этого, при использовании метода логического дополнения гораздо проще обеспечить свойство полной самопроверяемости структуры тестера благодаря возможности подбора значений преобразуемых функций [24].

Рассмотрим особенности реализации системы функционального контроля на основе метода логического дополнения.

2. Структурная схема системы функционального контроля на основе логического дополнения

В схеме контроля системы диагностирования на основе метода логического дополнения выделяются три функциональных блока (рис. 1): блок контрольной логики $G(x)$, блок логического дополнения и тестер TSC. Блок контрольной логики вычисляет значения функций дополнения рабочих функций f_1, f_2, \dots, f_m до вектора $\langle h_m h_{m-1} \dots h_2 h_1 \rangle$ выбранного кода по правилу:

$$h_i = f_i \oplus g_i, \quad i = 1, m. \quad (1)$$

Таким образом, блок логического дополнения представляет собой каскад сумматоров по модулю два, на выходах каждого из которых формируется один из разрядов кодового слова, выбранного на этапе проектирования кода. Тестер представляет собой устройство проверки принадлежности поступающего на его входы вектора мно-

жеству кодовых слов выбранного кода. Структура тестера должна обладать свойством полной самопроверяемости, что подразумевает при наличии на его входах вектора, не принадлежащего коду, и при наличии внутренних дефектов тестера формирование контрольного сигнала $\langle z^0 z^1 \rangle = \langle 00 \rangle$ или $\langle 11 \rangle$ [22].

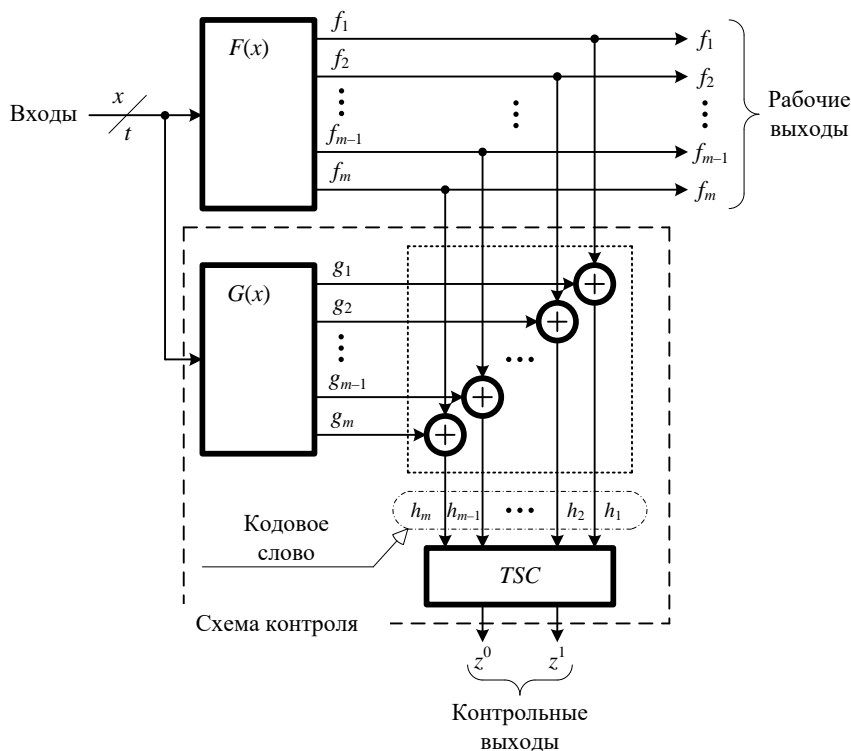


Рис. 1. Структурная схема системы функционального контроля на основе логического дополнения

При реализации системы функционального контроля на основе метода логического дополнения удобно использовать коды с небольшой длиной кодовых слов (тестеры таких кодов требуют для полной проверки небольшое количество комбинаций). К таким кодам, к примеру, относятся равновесные коды « r из n » или r/n -коды (n – длина кодового слова, а r – его вес). При организации системы функционального контроля на основе r/n -кодов используется следующий подход:

- 1) выходы объекта диагностирования $F(x)$ разбиваются на подмножества по n выходов в каждом (при этом подмножества могут пересекаться);
- 2) контроль каждого подмножества осуществляется на основе выбранного r/n -кода с помощью отдельной схемы контроля;
- 3) выходы отдельных схем контроля объединяются на входах самопроверяемой схемы сжатия парафазных сигналов [25, 26].

В [23, 27 – 31] изучены возможности применения 1/3-кода при организации систем функционального контроля, в [24, 32 – 34] – 1/4-кода и в [35 – 37] – 2/4-кода. Тестеры 1/3-кода и 1/4-кода про-

веряются подачей на их входы полного множества рабочих комбинаций, а тестер 2/4-кода в наиболее простой реализации – только четырех комбинаций их множества {0011; 1100; 1001; 0110} [35].

Перспективным для организации систем функционального контроля на основе метода логического дополнения может оказаться применение 1/5-кода. Множество тестовых комбинаций для полной проверки тестера данного кода содержит пять комбинаций {00001; 00010; 00100; 01000; 1000}.

Рассмотрим особенности применения 1/5-кода при организации систем функционального контроля.

3. Базовая структура системы функционального контроля на основе логического дополнения по 1/5-коду

Такая структура изображена на рис. 2. В схеме контроля блок контрольной логики имеет пять выходов, а блок логического дополнения содержит пять элементов сложения по модулю два, осуществляющих преобразования разрядов век-

тора $\langle f_5 f_4 f_3 f_2 f_1 \rangle$ в разряды кодовых слов $\langle h_5 h_4 h_3 h_2 h_1 \rangle$ равновесного 1/5-кода.

Значения контрольных функций g_1, \dots, g_5 получаются путем подбора с учетом следующих ограничений:

1) каждый вектор $\langle f_5 f_4 f_3 f_2 f_1 \rangle$ должен быть преобразован в вектор $\langle h_5 h_4 h_3 h_2 h_1 \rangle$, принадлежащий 1/5-коду;

2) необходимо хотя бы по одному разу сформировать каждый из векторов 1/5-кода, что обеспечивает тестирование 1/5-TSC (рис. 3);

3) необходимо обеспечить тестируемость каждого элемента сложения по модулю два в блоке логического дополнения, что требует при их канонической реализации формирования хотя бы по разу значений $\langle f_i g_i \rangle = \{00, 01, 10, 11\}$ [38].

Используя различные способы подбора значений контрольных функций, можно не только решить задачу обеспечения полной самопроверяемости

структуры системы диагностирования, но и минимизировать сложность технической реализации блока $G(x)$.

Необходимо отметить, что снижение структурной избыточности системы функционального контроля может быть достигнуто путем упрощения блока логического дополнения из следующих соображений. Любой вектор $\langle f_5 f_4 f_3 f_2 f_1 \rangle$ может быть преобразован в вектор $\langle h_5 h_4 h_3 h_2 h_1 \rangle$, принадлежащий 1/5-коду, при дополнении только четырех функций из пяти (рис. 4). Таким образом, один из разрядов преобразовывать не потребуется, а блок $G(x)$ при этом будет иметь четыре выхода вместо пяти.

Метод подбора значений контрольных функций имеет один недостаток – требуется подбор большого количества вариантов дополнения для оптимизации показателей системы функционального контроля.

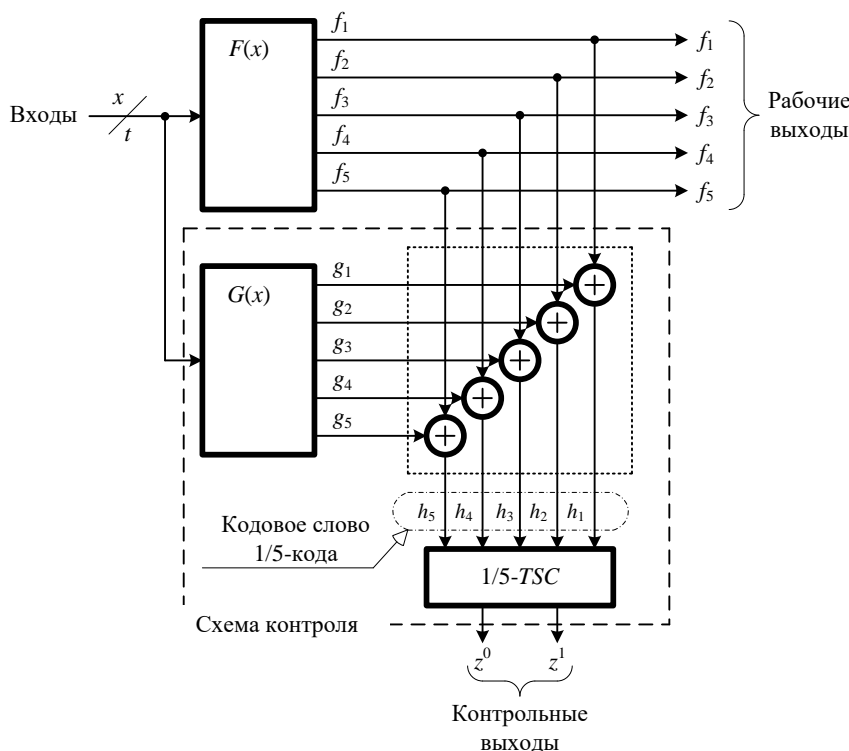


Рис. 2. Базовая структура системы функционального контроля на основе дополнения по 1/5-коду

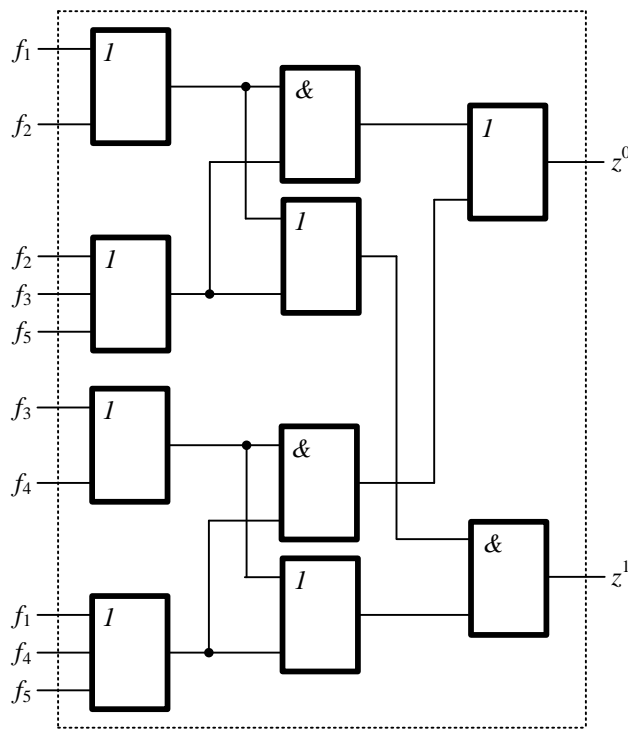


Рис. 3. Схема 1/5-TSC [39]

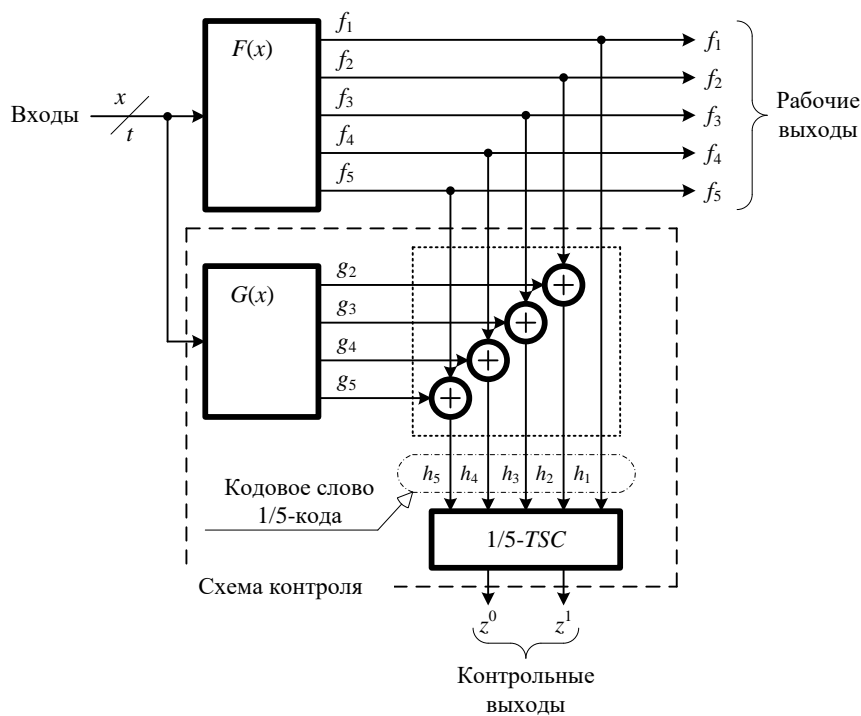


Рис. 4. Упрощенная базовая структура системы функционального контроля на основе дополнения по 1/5-коду

Значения выражений для контрольных функций могут быть получены по методике, примененной в работах [24, 33] для вычисления функций дополнения рабочих выходов до кодовых слов 1/3 и 1/4 кодов:

$$\begin{cases} g_1 = 0; \\ g_2 = f_1 f_2; \\ g_3 = (f_1 \vee f_2) f_3; \\ g_4 = (f_1 \vee f_2 \vee f_3) f_4; \\ g_5 = (f_1 \vee f_2 \vee f_3 \vee f_4) f_5 \vee \overline{f_1 f_2 f_3 f_4 f_5}. \end{cases} \quad (2)$$

4. Экспериментальные результаты

Для оценки эффективности разработанного авторами способа организации системы функционального контроля были проведены исследования со схемами с пятью выходами из набора контрольных комбинационных схем *MCNC Benchmarks* [40].

В ходе эксперимента оценивались показатели структурной избыточности систем диагностирования, реализованных на основе упрощенной базовой структуры, изображенной на рис. 4. На первом этапе было разработано специальное программное обеспечение, позволяющее для контрольных схем, заданных в формате *.pla [41], получать в том же формате описания всех блоков системы функционального контроля по исследуемому способу. На втором этапе с использованием интерпретатора SIS [42] и библиотеки функциональных элементов *stdcell2_2.genlib* определялся показатель площади, занимаемой на кристалле каждым логическим блоком системы функционального контроля (в условных единицах библиотеки). После этого определялся показатель площади системы диагностирования для трех структур: 1) система диагностирования на основе 1/5-кода; 2) система дублирования; 3) система контроля по классическому коду Бергера. Данные, полученные в ходе эксперимента, занесены в таблицу.

В двух последних столбцах таблицы записаны показатели, характеризующие систему функционального контроля по 1/5-коду в сравнении с

системами дублирования и контроля по коду Бергера. Коэффициент δ показывает долю площади системы контроля по 1/5-коду от площади системы дублирования (в процентах), а коэффициент β – долю площади системы контроля по 1/5-коду от площади системы контроля по коду Бергера (в процентах). По анализу значений коэффициентов δ и β можно сделать вывод о том, что для рассмотренных логических схем разработанный способ дает меньшую площадь системы диагностирования, чем при использовании дублирования и при организации контроля по коду Бергера.

Следует также отметить, что в описанном эксперименте никак не комбинировались выходы схем – не производилась перестановка выходов в векторе $\langle f_5 f_4 f_3 f_2 f_1 \rangle$. Однако, как следует из формулы (2), перестановка выходов в информационном векторе будет влиять на конечные выражения для функций g_i , а значит, и на площадь блока контрольной логики в системе функционального контроля (см. рис. 4). Это влияние может оказаться существенным и может позволить уменьшить показатели структурной избыточности разработанного способа организации системы функционального контроля. Всего существует $n!=5!=120$ вариантов перестановки для базовой структуры системы функционального контроля, что позволяет выбрать способ логического дополнения, дающий минимальную структурную избыточность при применении 1/5-кода.

Результаты экспериментов со схемами из набора *MCNC Benchmarks*

Название схемы	Площади элементов системы диагностирования				Площадь системы диагностирования	Площадь системы дублирования	Площадь системы контроля по коду Бергера	Показатели структурной избыточности	
	$F(x)$	$G(x)$	Блок дополнения	1/5-TSC				$\delta, \%$	$\beta, \%$
clpl	640	912	160	336	2048	2128	4480	96,241	45,714
dist	9976	4456	160	336	11920	14784	13048	80,628	91,355
root	6152	1552	160	336	5544	7840	6168	70,714	89,883
newcwp	440	464	160	336	1400	1728	1920	81,019	72,917

5. Заключение

Предложенный авторами способ построения системы функционального контроля на основе логического дополнения по 1/5-коду позволяет в ряде случаев организовывать более простые структуры, чем система дублирования и система контроля на основе кода Бергера. При этом достаточно просто обеспечивается свойство полной самопроверяемости тестера в системе функционального контроля.

Применение 1/5-кода при организации систем функционального контроля является перспек-

тивным и расширяет возможности проектировщика надежных дискретных устройств, что особенно эффективно при использовании схем программируемой логики типа *FPGA*.

Литература: 1. Пархоменко П.П., Согомонян Е.С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М.: Энергоатомиздат, 1981. 320 с. 2. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с. 3. Дрозд А.В., Харченко В.С., Антошук С.Г., Дрозд Ю.В., Дрозд М.А., Сулима Ю.Ю. Рабочее диагностирование безопасных информационно-управляющих систем // Под ред. А.В. Дрозда и В.С. Харченко. Харьков: Национальный аэрокосмический университет им. Н.Е. Жу-

- ковского «ХАИ», 2012, 614 с. **4.** *Kharchenko V., Kondratenko Yu., Kacprzyk J.* Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control", Vol. 74, 2017, 305 p. DOI: 10.1007/978-3-319-44162-7. **5.** *Nicolaidis M., Zorian Y.* On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Application, 1998, Vol. 12, issue 1-2. P. 7-20. DOI: 10.1023/A:1008244815697. **6.** *Mitra S., McCluskey E.J.* Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 03-05 October 2000. P. 985-994. **7.** *Berger J.M.* A Note on Error Detecting Codes for Asymmetric Channels // Information and Control, 1961, Vol. 4, issue 1, pp. 68-73. DOI: 10.1016/S0019-9958(61)80037-5. **8.** *Piestrak S.J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p. **9.** *Fujiwara E.* Code Design for Dependable Systems: Theory and Practical Applications // New Jersey: John Wiley & Sons, 2006. 720 p. **10.** *Lala P.K.* Principles of Modern Digital Design. New-Jersey: John Wiley & Sons, 2007. 436 p. **11.** *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Взвешенные коды с суммированием для организации контроля логических устройств // Электронное моделирование. 2014. Том 36, №1. С. 59-80. **12.** *Блюдов А.А., Ефанов Д.В., Сапожников В.В., Сапожников В.В.* О кодах с суммированием единичных разрядов в системах функционального контроля // Автоматика и телемеханика. 2014. №8. С. 131-145. **13.** *Freiman C.V.* Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control, 1962. Vol. 5. P. 64-71. DOI: 10.1016/S0019-9958(62)90223-1. **14.** *Borden J.M.* Optimal Asymmetric Error Detection Codes // Information and Control, 1982. Vol. 53, issue 1-2. P. 66-73. DOI: 10.1016/S0019-9958(82)91125-1. **15.** *Tallini L., Merani L., Bose B.* Balanced Codes for Noise Reduction in VLSI System // Dig. Pap. 24th International FTC Symposium, Austin, TX, June 15-17, 1994. P. 212-218. **16.** *Busaba F.Y., Lala P.K.* Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications, 1994, Vol. 5, issue 5. P. 19-28. **17.** *Das D., Touba N.A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes", Journal of Electronic Testing: Theory and Applications, 1999. Vol. 15, Issue 1-2. P. 145-155. DOI: 10.1023/A:1008344603814. **18.** *Das D., Touba N.A., Seuring M., Gossel M.* Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of the 6th IEEE International On-Line Testing Workshop (IOLTW), Spain, Palma de Mallorca, July 3-5, 2000. P. 171-176. DOI: 10.1109/OLT.2000.856633. **19.** *Ubar R., Raik J., Vierhaus H.-T.* Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source) // Information Science Reference, Hershey – New York, IGI Global, 2011. 578 p. **20.** *Sapozhnikov V., Sapozhnikov V.I., Efanov D.* Modular Sum Code in Building Testable Discrete Systems // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015), Batumi, Georgia, September 26-29, 2015. P. 181-187. DOI: 10.1109/EWDTS.2015.7493133. **21.** *Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V. 2008. 184 p. **22.** *Сапожников В.В., Сапожников В.В.* Самопроверяемые дискретные устройства. СПб: Энергоатомиздат, 1992. 224 с. **23.** *Гессель М., Морозов А.В., Сапожников В.В., Сапожников В.В.* Логическое дополнение – новый метод контроля комбинационных схем // Автоматика и телемеханика. 2003. №1. С. 167-176. **24.** *Гессель М., Морозов А.В., Сапожников В.В., Сапожников В.В.* Контроль комбинационных схем методом логического дополнения // Автоматика и телемеханика. 2005. №8. С. 161-172. **25.** *Carter W.C., Duke K.A., Schneider P.R.* Self-Checking Error Checker for Two-Rail Coded Data // United States Patent Office, filed July 25, 1968, ser. No. 747,533, patented Jan. 26, 1971, N. Y. 10 p. **26.** *Huches J.L.A., McCluskey E.J., Lu D.J.* Design of Totally Self-Checking Comparators with an Arbitrary Number of Inputs // IEEE Transactions on Computers. 1984. Vol. C-33. No. 6. P. 546-550. **27.** *Saposhnikov V.V., Morozov A., Saposhnikov V.I., Goessel M.* Concurrent Checking by Use of Complementary Circuits for «1-out-of-3» Codes // 5th International Workshop IEEE DDECS 2002, Brno, Czech Republic, April 17-19, 2002. **28.** *Сапожников В.В., Сапожников В.В.* О синтезе полностью самопроверяемых комбинационных схем // Известия Петербургского университета путей сообщения. 2006. №1 С. 97-110. **29.** *Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M.* Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proceedings of the 10th International Workshops on Boolean Problems, Freiburg, Germany, September, 2012. P. 33-40. **30.** *Efanov D., Sapozhnikov V., Sapozhnikov V.I.* Methods of Organization of Totally Self-Checking Concurrent Error Detection System on the Basis of Constant-Weight «1-out-of-3»-Code // Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, October 14-17, 2016. P. 117-125. OI: 10.1109/EWDTS.2016.7807622. **31.** *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Построение полностью самопроверяемых структур систем функционального контроля с использованием равновесного кода «1 из 3» // Электронное моделирование. 2016. Том 38. № 6. С. 25-43. **32.** *Sen S.K.* A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares // National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010), Sikkim Manipal Institute of Technology, Sikkim, held during 22-24 December, 2010. **33.** *Saposhnikov V.V., Saposhnikov V.I., Morozov A., Osadchi G., Gossel M.* Design of Totally Self-Checking Combinational Circuits by Use of Complementary Circuits // Proceedings of East-West Design & Test Workshop, Yalta, Ukraine, 2004. P. 83-87. **34.** *Сапожников В.В., Сапожников В.В., Ефанов Д.В., Пивоваров Д.В.* Метод логического дополнения на основе равновесного кода «1 из 4» для построения полностью самопроверяемых структур систем функционального контроля // Электронное моделирование. 2017. Том 39. №2. **35.** *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Метод функционального контроля комбинационных логических устройств на основе кода «2 из 4» // Известия вузов. Приборостроение, 2016, Т. 59, №7, С. 524-533. DOI 10.17586/0021-3454-2016-59-7-524-533. **36.** *Sapozhnikov V., Sapozhnikov V.I., Efanov D.* Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of «2-out-of-4» Code // Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, October 14-17, 2016. P. 126-133. DOI: 10.1109/EWDTS.2016.7807677. **37.** *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Построение самопроверяемых структур систем функционального контроля на основе равновесного кода «2 из 4» // Проблемы управления. 2017. №1. С. 57-64. **38.** *Аксёнова Г.П.* Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю два // Автоматика и телемеханика. 1979. № 9. С. 126-135. **39.** *Сапожников В.В., Рабара В.* Универсальный алгоритм синтеза 1/n-тестеров // Проблемы передачи информации. 1982. Том 18, №3. С. 62-73. **40.** *Collection of Digital Design Benchmarks* [Режимдоступа: <http://ddd.fit.cvut.cz/prj/Benchmarks/>]. **41.** *Yang S.* Logic Synthesis and Optimization Benchmarks: User Guide: Version 3.0. – Microelectronics Center of North Carolina (MCNC), 1991. 88 p. **42.** *SIS: A System for Sequential Circuit Synthesis* / E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, A. Sangio-

vanni-Vincentelli // Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 4 May 1992. 45 p.

Transliterated bibliography:

1. *Parhomenko P.P., Sogomonyan E.S.* Osnovy tehnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva). M.: Energoatomizdat, 1981, 320 s.
2. *Sogomonyan E.S., Slabakov E.V.* Samoproveryaemye ustroystva i otkazoustoychivyye sistemy. M.: Radio i svyaz, 1989. 208 s.
3. *Drozd A.V., Harchenko V.S., Antoschuk S.G., DrozdYu.V., Drozd M.A., SulimaYu.Yu.* Rabochee diagnostirovanie bezopasnykh informatsionno-upravlyayuschih sistem // Podred. A.V. Drozda i V.S. Harchenko. Harkov: Natsionalnyy aerokosmicheskyy universitetim. N.E. Zhukovskogo «HA», 2012, 614 s.
4. *Kharchenko V., Kondratenko Yu., Kacprzyk J.* Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control", Vol. 74, 2017, 305 p. DOI: 10.1007/978-3-319-44162-7.
5. *Nicolaidis M., Zorian Y.* On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Application, 1998, Vol. 12, issue 1-2, pp. 7-20. DOI: 10.1023/A:1008244815697.
6. *Mitra S., McCluskey E.J.* Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 03-05 October 2000, pp. 985-994.
7. *Berger J.M.* A Note on Error Detecting Codes for Asymmetric Channels // Information and Control, 1961, Vol. 4, issue 1, pp. 68-73. DOI: 10.1016/S0019-9958(61)80037-5.
8. *Piastrak S.J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995, 111 p.
9. *Fujiwara E.* Code Design for Dependable Systems: Theory and Practical Applications // New Jersey: John Wiley & Sons, 2006, 720 p.
10. *Lala P.K.* Principles of Modern Digital Design. New-Jersey: John Wiley & Sons, 2007, 436 p.
11. *Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V.* Vzveshennyie kody s summirovaniem dlya organizatsii kontrolya logicheskikh ustroystv // Elektronnoe modelirovanie. 2014. Tom 36. #1. S. 59-80.
12. *Blyudov A.A., Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I.* O kodakh s summirovaniem edinichnykh razryadov v sistemah funktsionalnogo kontrolya // Avtomatika i telemekhanika. 2014. #8. S. 131-145.
13. *Freiman C.V.* Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control, 1962, vol. 5, pp. 64-71. DOI: 10.1016/S0019-9958(62)90223-1.
14. *Borden J.M.* Optimal Asymmetric Error Detection Codes // Information and Control, 1982, vol. 53, issue 1-2, pp. 66-73. DOI: 10.1016/S0019-9958(82)91125-1.
15. *Tallini L., Merani L., Bose B.* Balanced Codes for Noise Reduction in VLSI System // Dig. Pap. 24th International FTC Symposium, Austin, TX, June 15-17, 1994, pp. 212-218.
16. *Busaba F.Y., Lala P.K.* Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications, 1994, Vol. 5, issue 5, pp. 19-28.
17. *Das D., Toubia N.A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications, 1999, Vol. 15, Issue 1-2, pp. 145-155. DOI: 10.1023/A:1008344603814.
18. *Das D., Toubia N.A., Seuring M., Gossel M.* Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of the 6th IEEE International On-Line Testing Workshop (IOLTW), Spain, Palma de Mallorca, July 3-5, 2000, pp. 171-176. DOI: 10.1109/OLT.2000.856633.
19. *Ubar R., Raik J., Vierhaus H.-T.* Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source) // Information Science Reference, Hershey – New York, IGI Global, 2011, 578 p.
20. *Sapozhnikov V., Sapozhnikov V.I., Efanov D.* Modular Sum Code in Building Testable Discrete Systems // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDT'S 2015), Batumi, Georgia, September 26-29, 2015, pp. 181-187. DOI: 10.1109/EWDT'S.2015.7493133.
21. *Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008, 184 p.
22. *Sapozhnikov V.V., Sapozhnikov V.I.* Samoproveryaemye diskretnyye ustroystva. SPb: Energoatomizdat, 1992, 224 s.
23. *Gessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov V.I.* Logicheskoe dopolnenie – novyy metod kontrolya kombinatsionnykh shem // Avtomatika i telemekhanika. 2003. #1. S. 167-176.
24. *Gessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov V.I.* Kontrol kombinatsionnykh shem metodom logicheskogo dopolneniya // Avtomatika i telemekhanika. 2005. #8. S. 161-172.
25. *Carter W.C., Duke K.A., Schneider P.R.* Self-Checking Error Checker for Two-Rail Coded Data // United States Patent Office, filed July 25, 1968, ser. No. 747,533, patented Jan. 26, 1971, N. Y., 10 p.
26. *Huches J.L.A., McCluskey E.J., Lu D.J.* Design of Totally Self-Checking Comparators with an Arbitrary Number of Inputs // IEEE Transactions on Computers. 1984. Vol. C-33. No. 6. Pp. 546-550.
27. *Saposhnikov V.V., Morozov A., Sapozhnikov V.I., Göessel M.* Concurrent Checking by Use of Complementary Circuits for «1-out-of-3» Codes // 5th International Workshop IEEE DDECS 2002, Brno, Czech Republic, April 17-19, 2002.
28. *Sapozhnikov V.V., Sapozhnikov V.I.* O sintezepolnostyusamoproveryaemykh kombinatsionnykh shem // Izvestiya Peterburgskogo universiteta puty soobshcheniya. 2006. #1 S. 97-110.
29. *Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M.* Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proceedings of the 10th International Workshops on Boolean Problems, Freiberg, Germany, September, 2012, pp. 33-40.
30. *Efanov D., Sapozhnikov V., Sapozhnikov V.I.* Methods of Organization of Totally Self-Checking Concurrent Error Detection System on the Basis of Constant-Weight «1-out-of-3»-Code // Proceedings of 14th IEEE East-West Design & Test Symposium (EWDT'S 2016), Yerevan, Armenia, October 14-17, 2016, pp. 117-125. DOI: 10.1109/EWDT'S.2016.7807622.
31. *Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V.* Postroenie polnostyu samoproveryaemykh struktur sistem funktsionalnogo kontrolya s ispolzovaniem ravnovesnogo koda «1 iz 3» // Elektronnoe modelirovanie. 2016. Tom 38. #6. S. 25-43.
32. *Sen S.K.* A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares // National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010), Sikkim Manipal Institute of Technology, Sikkim, held during 22-24 December, 2010.
33. *Saposhnikov V.V., Saposhnikov V.I., Morozov A., Osadtchi G., Gossel M.* Design of Totally Self-Checking Combinational Circuits by Use of Complementary Circuits // Proceedings of East-West Design & Test Workshop, Yalta, Ukraine, 2004, pp. 83-87.
34. *Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V., Pivovarov D.V.* Metod logicheskogo dopolneniya na osnoven ravnovesnogo koda «1 iz 4» dlya postroeniya polnostyu samoproveryaemykh struktur sistem funktsionalnogo kontrolya // Elektronnoe modelirovanie. 2017. Tom 39. #2.

35. Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V. Metod funktsionalnogo kontrolya kombinatsionnykh logicheskikh ustroystva osnovan na kode «2 iz 4» // Izvestiya vuzov. Priborostroyeniye, 2016, T. 59, #7, S. 524-533. DOI 10.17586/0021-3454-2016-59-7-524-533.
36. Sapozhnikov V., Sapozhnikov V.I., Efanov D. Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of «2-out-of-4» Code // Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, October 14-17, 2016, pp. 126-133. DOI: 10.1109/EWDTS.2016.7807677.
37. Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V. Problemy upravleniya stroeniyami samoproveryaemykh struktursistem funktsionalnogo kontrolya na osnovaniye novosnogo koda «2 iz 4» // Problemy upravleniya. 2017. #1. S. 57-64.
38. Aksyonova G.P. Neobhodimyie i dostatochnyye usloviya postroyeniya polnostyu proveryaemykh shem sverkipomodulyudva // Avtomatika i telemekhanika. 1979. #9. S. 126-135.
39. Sapozhnikov V.V., Rabara V. Universalnyy algoritm sinteza 1/n-testerov // Problemy i peredachi informatsii. 1982. Tom. 18, #3. S. 62-73.
40. *Collection of Digital Design Benchmarks* [Режим доступа: <http://ddd.fit.cvut.cz/prj/Benchmarks/>].
41. Yang S. Logic Synthesis and Optimization Benchmarks: User Guide: Version 3.0. Microelectronics Center of North Carolina (MCNC), 1991, 88 p.
42. SIS: A System for Sequential Circuit Synthesis / E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, A. Sangiovanni-Vincentelli // Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 4 May 1992, 45 p.

Поступила в редколлегию 22.03.2017

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Сапожников Валерий Владимирович, д-р техн. наук, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного универси-

тета путей сообщения Императора Александра I. Окончил Ленинградский институт инженеров железнодорожного транспорта в 1963 г. Научные интересы: надежный синтез дискретных устройств, синтез безопасных систем, синтез самопроверяемых схем, техническая диагностика дискретных систем. Адрес: Россия, Санкт-Петербург, Московский пр., 9, тел.: +7 (812) 310-1097.

Сапожников Владимир Владимирович, д-р техн. наук, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I. Окончил Ленинградский институт инженеров железнодорожного транспорта в 1963 г. Научные интересы: надежный синтез дискретных устройств, синтез безопасных систем, синтез самопроверяемых схем, техническая диагностика дискретных систем. Адрес: Россия, Санкт-Петербург, Московский пр., 9, тел.: +7 (812) 310-0788.

Ефанов Дмитрий Викторович, канд. техн. наук, доцент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I. Окончил Петербургский государственный университет путей сообщения в 2007 г. Научные интересы: дискретная математика, надежность и техническая диагностика дискретных систем. Адрес: Россия, Санкт-Петербург, Московский пр., 9, тел.: +7 (812) 457-8579, +7 (911) 709-2164, email: TrES-4b@yandex.ru

Пивоваров Дмитрий Вячеславович, аспирант кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I. Окончил Петербургский государственный университет путей сообщения в 2016 г. Научные интересы: дискретная математика, техническая диагностика дискретных систем. Адрес: Россия, Санкт-Петербург, Московский пр., 9, тел.: +7 (812) 457-8579, +7 (911) 772-5099, email: pivovarov.d.v.spb@gmail.com

ВСТРОЕННАЯ СИСТЕМА РЕГУЛИРОВАНИЯ ТЕМПЕРАТУРЫ В «УМНОМ ДОМЕ»

ФИЛИППЕНКО И.В., КОНДРЮКОВ С.Э.,
КУЛАК Г.К.

Предлагается метод автоматического регулирования микроклимата и снижения энергопотребления в системах «Умный дом», а также метод регулирования температуры воздуха в доме на основе ПИД регулятора. Выполняется программно-аппаратная реализация системы ПИД регулирования и расчетно-экспериментальный подбор коэффициентов для ПИД регулятора.

1. Введение

Актуальность. Исследования, представленные в этой работе, являются частью проекта студенческой исследовательской группы кафедры АПВТ ХНУРЭ. Проект предусматривает разработку автоматической системы регулирования микроклимата в помещении на основе ПИД регулятора.

Цель – разработать часть системы управления «Умный дом», связанную с регулировкой температуры воздуха. Методы исследования: анализ, синтез, измерение, эксперимент.

Задачи: исследование и обоснование выбора методов регулирования температуры и средств реализации системы регулирования; формулировка общей концепции системы «Умный дом»; разработка регулятора температуры воздуха; программно-аппаратная реализация системы регулирования;

В последние годы вопрос энергосбережения является одним из первоочередных. Это проявляется в использовании энергосберегающих технологий и приспособлений.

Под термином «Умный дом» часто подразумевают современные здания (офисы, дома, квартиры), в которые внедрены различные системы автоматизации, такие как освещение, микроклимат, коммуникационные системы, безопасность. Одной из основных задач данной системы является повышение комфорта жильцов, их безопасность, а также экономия ресурсов.

Система управления «Умным домом» состоит из облачного сервера, устройств управления (мобильных или стационарных), домашнего сервера, контроллера, а также конечных устройств (датчики, управляемые устройства) [1]. Облачный сервер предназначен для хранения данных от

различных устройств системы в базе данных. Также он выступает в качестве посредника между мобильными устройствами управления и домашним сервером. Вначале запросы/данные передаются на облачный сервер, а после – на домашний. Это позволяет мониторить состояние дома в текущий момент времени с помощью мобильных устройств из любой точки мира, при наличии подключения к интернету. Под устройствами управления подразумеваются мобильные и стационарные устройства, с помощью которых пользователь может получить доступ к текущему состоянию всей системы или передать ей определенную команду для выполнения необходимого действия. Домашний сервер - это сервер, отвечающий за мониторинг и работу всех устройств в доме. Он связан с контроллером, облачным сервером, а также с устройствами управления. Контроллер отвечает за управление устройствами и датчиками системы. Он считывает значения с датчиков, передает их данные на обработку серверу, также передает необходимые команды, полученные от сервера. Общий вид описанной выше структуры представлен на рис. 1.

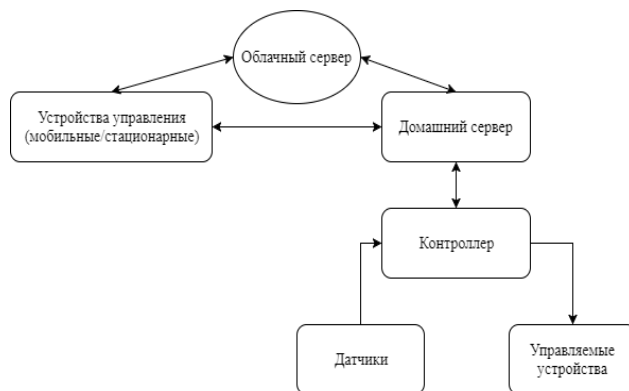


Рис. 1. Схема взаимодействия компонентов системы «Умный дом»

2. Методы регулирования температуры

Существует большое количество методов автоматического регулирования: размещение полюсов [2], методы, основанные на использовании логарифмических частотных характеристик [2], на базе фильтра Калмана [3] и многие другие. Все они классифицируются [4, 5] по назначению, принципу действия, конструктивным особенностям, виду используемой энергии, характеру изменения регулирующего воздействия.

По принципу действия они подразделяются на регуляторы прямого и непрямого действия. Регуляторы прямого действия не используют внешнюю энергию для процессов управления, а используют энергию самого объекта управления

(регулируемой среды). Примером таких регуляторов являются регуляторы давления. Для работы автоматических регуляторов непрямого действия требуется внешний источник энергии.

По роду действия регуляторы делятся на непрерывные и дискретные. Дискретные регуляторы, в свою очередь, подразделяются на релейные, цифровые и импульсные.

По виду используемой энергии они подразделяются на электрические (электронные), пневматические, гидравлические, механические и комбинированные. Выбор регулятора по виду используемой энергии определяется характером объекта регулирования и особенностями автоматической системы. В современных системах управления используются цифровые программные регуляторы.

По закону регулирования они делятся на однопозиционные, двухпозиционные и трехпозиционные. Типовые регуляторы: интегральные – сокращенно И, пропорциональные – сокращенно П, пропорционально-интегральные – ПИ, пропорционально-дифференциальные – ПД и пропорционально-интегрально-дифференциальные – ПИД. Двухпозиционные регуляторы широко распространены благодаря своей простоте и малой стоимости.

По назначению регуляторы подразделяются на специализированные (например: регуляторы уровня, давления, температуры) и универсальные с нормированными входными и выходными сигналами, пригодные для управления различными параметрами.

По виду выполняемых функций они подразделяются на регуляторы автоматической стабилизации, программные, корректирующие, регуляторы соотношения параметров и другие.

Несмотря на развитые современные методы проектирования сложных регуляторов, подавляющее большинство промышленных систем управления основаны на П, ПД, ПИ и ПИД регуляторах (порядка 90 – 95 %) [6]. Регуляторы этого типа очень хорошо зарекомендовали себя в практических задачах. ПИД регулятор применяется в инерционных системах с относительно малым уровнем помех измерительного канала. Достоинством ПИД регулятора является быстрый выход на режим, точное удержание заданного параметра (например, температуры) и быстрая реакция на возмущающие воздействия. ПИД регулятор

может экономить до 35% мощности исполнительного механизма.

Существуют разнообразные способы реализации ПИД регулятора на базе различных микроконтроллеров с использованием разных языков программирования. В нашем случае Arduino дает возможность сделать это проще и дешевле – используя библиотеку PID со встроенной функцией PID (), реализующей ПИД регулятор.

Одним из способов выбора коэффициентов для ПИД регулятора является эксперимент с реальным объектом. Процесс настройки ПИД регулятора по экспериментальным правилам интуитивен и попытки настроить регулятор без начального приближенного расчета коэффициентов могут оказаться безуспешными. Основываясь на обзоре существующих методик настройки [7 –9], можно заключить, что методы Циглера-Никольса и Чина, Хронса, Ресвика наиболее оправданы для эксплуатации в промышленных условиях.

После расчета коэффициентов регулятора желательно сделать его подстройку. Подстройку можно выполнить на основе правил, которые используются для ручной настройки. Эти правила получены из опыта, теоретического анализа и численных экспериментов. Они сводятся к следующему: увеличение пропорционального коэффициента повышает быстродействие и снижает запас устойчивости; с уменьшением интегральной составляющей ошибка регулирования с течением времени уменьшается быстрее; уменьшение постоянной интегрирования снижает запас устойчивости; увеличение дифференциальной составляющей повышает запас устойчивости и быстродействие.

Сформулированные выше правила справедливы только в окрестности оптимальной настройки регулятора. Вдали от нее эффекты могут быть иными. При регулировке тепловых процессов настройка только по правилам может занять недопустимо много времени.

3. Регулирование температуры воздуха в помещении

Рассмотрим управление температурой воздуха в комнате. Выделяются три диапазона температур: ниже 20°C – воздух подогревается; от 20 до 24°C – действий никаких не нужно, это зона комфорта; выше 24°C – воздух охлаждается (подается сигнал на охлаждение). Например, необходимо поддерживать температуру на уровне 22°C. В данной работе рассматривается только регулирование

температуры при нагреве (рис. 2). Имеются объект регулирования (температура воздуха), цифровой датчик, измеряющий эту температуру. Датчик сообщает значение температуры микроконтроллеру, который [10] вычисляет необходимость регулирования температуры при помощи нагревания или охлаждения.



Рис. 2. Структурная схема для регуляции температуры

Таким образом, есть входные данные – текущая температура и температура, до которой необходимо нагреть объект. На выходе нужно получить значение мощности $u(t)$, которое следует передать на нагревающее устройство. Для реализации функции регулирования необходимо использовать ПИД-регулятор, закон которого представлен формулой

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt}. \quad (1)$$

где $u(t)$ – искомое выходное воздействие

$$P = K_p e(t). \quad (2)$$

– пропорциональная составляющая;

$$I = K_i \int_0^t e(t) dt. \quad (3)$$

– интегральная составляющая;

$$D = K_d \frac{de}{dt}. \quad (4)$$

– дифференциальная составляющая; $e(t)$ – текущая ошибка (отклонение температуры от заданного значения); K_p , K_i , K_d – пропорциональный, интегральный, дифференциальный коэффициенты соответственно.

Для дискретной реализации ПИД функции используются следующие выражения:

$$u(t) = P(t) + I(t) + D(t), \quad (5)$$

$$P(t) = K_p \cdot e(t), \quad (6)$$

$$I(t) = (t - 1) + K_i \cdot e(t), \quad (7)$$

$$D(t) = K_d \cdot (e(t) - (e(t - 1))). \quad (8)$$

Разница между текущей и заданной температурой $e(t)$ умножается на настраиваемый коэффициент K_p . Получается мощность, которую надо выдать в данный момент времени. Это пропор-

циональная составляющая. Она работает в момент появления рассогласования – т.е. моментально откликается на поведение объекта. Если объект начал нагреваться, мощность начинает снижаться. Если температура превышает заданный порог, нагреватель отключается. В реальных условиях эффект от воздействия проявляется с запаздыванием, так как на объект воздействует не только нагреватель, но и окружающая среда: разогретый нагреватель остывает, отдавая тепло комнате, поэтому, как только выключается мощность, он сразу начинает остывать. Использование только пропорциональной составляющей регулятора приводит к колебаниям температуры около заданного значения, и чем выше воздействие окружающей среды, тем сильнее эти колебания.

Чтобы компенсировать внешние воздействия на нагреватель, добавляют интегральную составляющую. Всё рассогласование, которое было в системе, накапливается в интеграторе (воздух перегревается – сумма уменьшается, воздух недостаточно нагрет – сумма увеличивается). И накопленный интеграл, со своим коэффициентом, даёт свою «прибавку-убавку» к мощности. В результате такого подхода, при стационарном процессе, через некоторое время интеграл подбирает такой вклад в сумму с мощностью, который компенсирует потери окружающей среды, и колебания исчезают – интеграл становится стабильным, поэтому величина выдаваемой мощности становится постоянной. Так как при этом держится нужная температура, рассогласование отсутствует, пропорциональная составляющая не работает вообще.

Для компенсации влияния задержек между воздействием и реакцией системы в систему добавляется дифференциальная составляющая. Просто пропорциональный регулятор даёт мощность всё время, пока температура не достигнет нужной точки. Пропорционально-дифференциальный регулятор начинает снижать подаваемую мощность раньше, чем температура достигла нужной точки (так как рассогласование уменьшается, имеется наличие отрицательной производной, уменьшающей воздействие). Это позволяет минимизировать перегрев при больших переходах.

Эксперимент проводился следующим образом. Для макета была сделана комната – фанерный короб. Нагревался воздух в этом коробе. Использовался нагреватель с регулируемой мощностью.

Для реализации ПИД регулятора в Arduino использовалась функция PID() из библиотеки PID. Заданное (желаемое) значение температуры $T_{\text{зад}}=22^{\circ}\text{C}$. Значение регулируемой температуры отображалось на компьютере для сохранения и дальнейшего построения графиков.

Далее был выполнен предварительный расчет коэффициентов регулятора по методу Циглера-Никольса. Для этого, вначале, коэффициентам K_i и K_d были назначены нулевые значения. Затем постепенно увеличивался K_p до критического значения $K_c = 12$, при котором возникают автоколебания. На рис. 3 графику 1 соответствует $K_p=1$, графику 2 – $K_p = 4$, графику 3 – $K_p = K_c = 12$. Далее был измерен период автоколебаний $\tau=102$ с, а после рассчитаны значения коэффициентов K_p , K_i и K_d : $K_p = 0,60 * K_c = 7,2$, $K_i = 2,0 * K_p / \tau = 0,14$, $K_d = K_p * \tau / 8 = 91,8$, которым соответствует график 2 на рис. 4.

Затем экспериментально в районе вычисляемых значений были подобраны коэффициенты, чтобы обеспечить необходимый переходной процесс. С фиксированными значениями K_p был выбран коэффициент K_i , потом с подобранными фиксированными значениями K_i и K_p подбирался K_d . В результате был получен набор коэффициентов, который обеспечивал переходной процесс, представленный графиком изменения температуры под номером 3 (см. рис. 4). Этому графику соответствуют коэффициенты $K_p = 7,2$, $K_i = 0,06$, $K_d=108$. На рис. 4 графику 1 соответствуют коэффициенты $K_p = 12$, $K_i = 0$, $K_d = 0$.

Качество регулирования оценивалось по характеристикам переходного процесса: перерегулирование, время нарастания, время переходного процесса, степень затухания (см. рис. 4): t_n – время нарастания; t_p – время переходного процесса; A_1 – первое перерегулирование; A_2 – второе перерегулирование; $d = A_1 / A_2$ – степень (декремент) затухания переходного процесса (отношение первого перерегулирования ко второму). График 3 (см. рис. 4) соответствует лучшим показателям качества регулирования: $t_n = 200$ с, $t_p = 540$ с, $A_1 = 1,3^{\circ}\text{C}$, $A_2 = 0,55^{\circ}\text{C}$, $d = A_1 / A_2 = 1,3 / 0,55 = 2,36$.

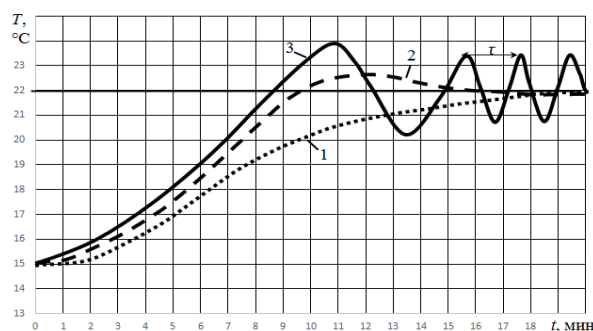


Рис. 3. Подбор коэффициента K_p

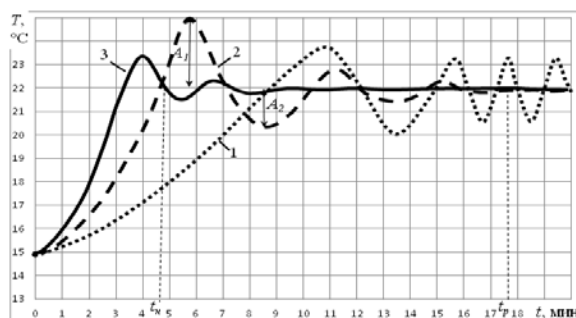


Рис. 4. Подбор коэффициентов K_p , K_i и K_d

4. Программно-аппаратная реализация эксперимента

Эксперимент был реализован с помощью устройства ArduinoUno на базе микроконтроллера ATmega328. Arduino - это открытая программируемая аппаратная платформа для работы с разными физическими объектами. Она включает в себя все необходимое для комфортной работы с микроконтроллером.

Для реализации необходимого модуля ПИД регулирования были использованы следующие модули: датчик температуры, нагревательный элемент, а также ArduinoUno.

Датчик температуры [11, 12] передавал данные непосредственно на ArduinoUno, а он в свою очередь обрабатывал их и выдавал необходимое значение на нагревательный элемент. Нагревательный элемент в зависимости от посылаемого на него значения меняет мощность нагревания от 0 до 100 процентов. После этого цикл повторяется заново.

Научная новизна: представлен расчетно-экспериментальный подбор коэффициентов для ПИД регулятора.

Практическая значимость: результаты исследования могут использоваться в системе «Умный дом», концепция которой построена на энергосбережении. Собственная разработка в части управления, а также собственная сборка и отладка системы позволят сделать ее дешевле и до-

ступнее в сравнении с готовыми системами «Умный дом», которые предлагаются рынком.

Литература: 1. Николаев П.Л. Архитектура интегрированной в облачную среду системы управления умным домом / П.Л. Николаев // Программные продукты и системы. Международный научно-технический журнал. 2015. №2(110). С.65-69. 2. Поляков К. Ю. Теория автоматического управления для чайников (часть 1) / К. Ю. Поляков: Санкт-Петербург, 2008. 80 с. 3. Шилов К.Е. Разработка системы автоматического управления беспилотным летательным аппаратом мультироторного типа / К. Е. Шилов // Математика, информатика, управление. Труды МФТИ. Том 6, № 4. 2014. С. 22-29. 4. Борисевич А. В. Теория автоматического управления: элементарное введение с применением MATLAB / А. В. Борисевич. СПб.: Изд-во Политехн. ун-та, 2011. 200 с. 5. Dorf P. Современные системы управления / Р. Dorf, Р. Бишоп. М.: Бинум, Лаборатория базовых знаний, 2004. 832 с. 6. Денисенко В.Д. ПИД-регуляторы: принципы построения и модификации. Ч. 1 // Современные технологии автоматизации. 2006. №4. С. 66-74. 7. Писарев, А. В. Сравнительные исследования расчетных методов определения параметров настроек промышленных ПИД-регуляторов / А.В. Писарев, С.И. Новиков // Энергетика и теплотехника. Сборник научных трудов. Вып. 11. НГТУ, 2007. С. 191-200. 8. Wen Tan. Comparison of some well-known PID tuning formulas / Wen. Tan, Jizhen Liu, Tongwen Chen, Horacio. J. Marquez // Computers and Chemical Engineering. 2006. № 30. P. 1416-1423. 9. Astrom K. J. Advanced PID control / K. J. Astrom, T. Hagglund – ISA (The instrumentation, Systems and Automation Society), 2006. 460 p. 10. Белов А. В. Конструирование устройств на микроконтроллерах. / А. В. Белов. СПб.: Наука и техника, 2005. 256 с. 11. Гордон М. К. Радиоэлектроника для «чайников» / М. К. Гордон, Б. Эрл: Пер. с англ. М., 2007. 400 с. 12. Иго Т. Arduino, датчики и сети для связи устройств / Т. Иго: Пер. с англ. СПб.: БХВ-Петербург, 2015. 544с.

Transliterated bibliography:

1. Nikolaev P.L. Arhitektura integrirovannoy v oblachnyuyu sredyu sistemy I upravleniya umnyim domom / P.L. Nikolaev // Programmnyye produkty I isistemyi. Mezhdunarodnyi ynauchno-tehnicheskij zhurnal #2(110). 2015. S. 65-69.
2. Polyakov K. Yu. Teoriya avtomaticheskogo upravleniya dlya chaynikov (chast 1) / K. Yu. Polyakov: Sankt-Peterburg, 2008. 80 s.
3. Shilov K.E. Razrabotka sistemyi avtomaticheskogo upravleniya bespilotnyim letatelnyim apparatom multiro-tornogotipa / K. E. Shilov // Matematika, informatika, upravlenie. Trudy MFTI. Tom 6, # 4. 2014. S. 22-29.

4. Borisevich A. V. Teoriya avtomaticheskogo upravleniya: elementarnoe vvedenie s primeneniem MATLAB / A. V. Borisevich. SPb.:Izd-vo Politehn. un-ta, 2011. 200 s.
5. Dorf R. Sovremennyye sistemyi upravleniya / R. Dorf, R. Bishop. M.: Binom, Laboratoriya bazovyih znaniy, 2004. 832 s.
6. Denisenko V.D. PID-regulyatoryi: printsipyi postroeniya I modifikatsii. Ch. 1 // Sovremennyye tehnologii avtomatizatsii. 2006. #4. S. 66-74.
7. Pisarev, A. V. Sravnitelnyie issledovaniya raschetnyih metodov opredeleniya parametrov nastroek promyshlennyih PID-regulyatorov / A.V. Pisarev, S.I. Novikov // Energetika I teplo-tehnika. Sbornik nauchnyih trudov. Vyip. 11. NGTU, 2007. S. 191-200.
8. Wen Tan. Comparison of some well-known PID tuning formulas / Wen. Tan, Jizhen Liu, Tongwen Chen, Horacio. J. Marquez // Computers and Chemical Engineering. 2006. # 30. P. 1416-1423.
9. Astrom K. J. Advanced PID control / K. J. Astrom, T. Hagglund – ISA (The instrumentation, Systems and Automation Society), 2006. 460 p.
10. Belov A. V. Konstruirovaniye ustroystv namikrokontrollerah / A. V. Belov. Spb.: Nauka i tehnika, 2005. 256 s.
11. Gordon M. K. Radio-elektronikadlya «chaynikov» / M. K. Gordon, B. Erl: Per. s angl. M., 2007. 400 s.
12. Igo T. Arduino, datchiki I seti dlya svyazi ustroystv / T. Igo: Per. s angl. SPb.: BHV-Peterburg, 2015. 544 s.:

Поступила в редколлегия 19.05.2017

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Филиппенко Инна Викторовна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: проектирование микроконтроллерных систем, сенсорных беспроводных систем. Адрес: Украина, 61166, Харьков, пр. Науки, 14.

Кондюков Станислав Эдуардович, студент, кафедра АПВТ ХНУРЭ. Научные интересы: проектирование микроконтроллерных систем управления. Адрес: Украина, 61166, Харьков, пр. Науки, 14.

Кулак Георгий Константинович, ученик средней школы, кандидат в члены Малой академии наук Украины. Научные интересы: проектирование микроконтроллерных систем управления. Адрес: Украина, 61166, Харьков, пр. Науки, 14.

Filippenko Inna Victorovna, PhD, Associate Professor, Design Automation Department, Kharkov National University of Radioelectronics. Scientific interests: microcontroller systems design, sensor wireless systems. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14.

Kondriukov Stanislav Eduardovich, student, Design Automation Department, Kharkov National University of Radioelectronics. Scientific interests: microcontroller control systems design. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14.

Kulak Georgiy Konstantinovich, schoolboy, candidate member of the Small Academy of Sciences of Ukraine. Scientific interests: designing of microcontroller control systems. Address: Ukraine, 61166, Kharkov, science, 14.

**СИНТЕЗ КУБИТНЫХ ПОКРЫТИЙ
ДЛЯ ЦИФРОВЫХ СИСТЕМ***БОЯДЖАН А.Г., КОТЛЯРОВ А.С.*

Описываются квантовые компьютеры, которые могут решать проблемы, трудно разрешимые даже для самых мощных современных суперкомпьютеров. Они имеют потенциал для решения проблем в области материаловедения, химии и математики, которые далеко не подходят для суперкомпьютеров. Их мощность обусловлена использованием квантовых битов, которые могут одновременно существовать как в 0, так и в квантовом состоянии суперпозиции. Синтез кубических покрытий для цифровых систем является основополагающей парадигмой для квантового компьютеринга [1,2].

Ключевые слова: кубит; квантовый компьютеринг; синтез; цифровая система.

Key words: qubit; quantum computing; synthesis; digital system.

1. Введение

Понятие адресного выполнения логических операций, реализованных на элементах памяти LUT в программируемых логических устройствах (PLD), дает потенциальную возможность создавать на кристалле только адресное пространство, максимально технологичное для встроенного восстановления работоспособности всех компонентов, участвующих в формировании функциональности [5, 6].

Тенденция к увеличению памяти влечет возможность встроенного восстановления работоспособности отказавших ячеек за счет выделенных дополнительных ресурсов для их ремонта (sparelogiccells). Проблема автономного устранения дефектов (самовосстановления работоспособности) логических элементов связана с отсутствием у них адресов. Но решить ее можно, если связи между элементами логики сделать гибкими с помощью программы описания структуры, помещенной в память, которая соединит логические компоненты в схему. Кроме структуры взаимодействия элементов, память должна содержать порядок их обработки. В случае возникновения дефекта в одном из адресуемых логических элементов система встроенного тестирования восстановит его работоспособность путем переадресации на заведомо исправный аналог из ремонтного запаса. Таким образом, решается проблема повышения качества и надежности цифровых систем на кристаллах путем создания инфраструктуры встроенного тестирования, диагностирования, оптимизации и восстановления работоспособности за счет аппаратной избыточности и уменьшения быстродействия выполнения функциональных операций [3,7].

Другими словами, любой вычислительный процесс сводится к адресным операциям считыва-

ния-записи на памяти. Практическая значимость данной парадигмы заключается в исключении аппаратной логики из компьютеринга, что дает возможность:

- 1) Сделать регулярной архитектуру компьютера на основе использования только матриц памяти.
- 2) Выполнять все арифметические, логические и специальные операции на матричной структуре памяти без обращения к внешнему модулю АЛУ, который значительно уменьшает быстродействие.
- 3) Технологично решать все проблемы надежности компьютерных систем на основе встроенного online-тестирования и ремонта отказавших модулей памяти путем их переадресации на исправные из ремонтного запаса.
- 4) Иметь неограниченные возможности в пределах одного кристалла для параллельного выполнения логических операций любой регистровой размерности.
- 5) Существенно уменьшать время проектирования специализированных процессоров за счет исключения сложных и нерегулярных блоков АЛУ с шинами обмена информацией.
- 6) Перепрограммировать в режиме online логику элементов памяти для выполнения других операций [5, 6].

Далее рассмотрим синтез кубических покрытий для цифровых устройств. На примерах будет показано, как можно проводить процесс синтеза схемы на памяти. Предлагается существенное уменьшение времени при синтезе цифровых схем.

Цель – показать технологичность и производительность методов и алгоритмов, использующих кубитные покрытия функциональных примитивов.

Задачи: 1) Синтез кубитных моделей цифровых схем. 2) Минимизация кубитных покрытий. 3) Верификация метода и разработка алгоритма синтеза моделей для их имплементации в программный сервис.

2. Кубитная форма описания примитивов

Кубит (n -кубит) есть векторная форма унитарного кодирования универсума из n примитивов для задания булеана состояний 2^{2^n} с помощью 2^n двоичных переменных. Если $n=2$, то 2-кубит задает 16 состояний с помощью четырех переменных, при $n=1$, кубит задает четыре состояния на универсуме из двух примитивов (10) и (01) с помощью двух двоичных переменных (00,01,10,11). При этом допускается суперпозиция в векторе 2^n состояний, обозначенных примитивами. Синонимом кубита при задании двоичного вектора логической функции является Q -покрытие (Q -вектор), как унифицированная векторная форма суперпозиционного задания вы-

ходных состояний, соответствующих адресным кодам входных переменных функционального элемента. Формат структурного кубитного компонента цифровой схемы $Q^* = (X, Q, Y)$ включает интерфейс (входные и выходную переменные), а также кубит-вектор Q , задающий функцию $Y = Q(X)$, размерность которого определяется степенной функцией от числа входных линий $k = 2^n$ [6]. Новизна кубитной формы заключается в замене неупорядоченных по строкам таблиц истинности функциональных элементов векторами упорядоченных состояний выходов. Например, если функциональный примитив имеет двоичную таблицу, то ему можно поставить в соответствие кубит или Q -покрытие: $Q = (1110)$ [4-6]:

	X_1	X_2	Y	
	0	0	1	
$C =$	0	1	1	$\rightarrow Q = (1110)$.
	1	0	1	
	1	1	0	

Таким образом, идея, положенная в основу исследований, заключается в замене множества вход-выходных соответствий таблицы истинности кубитным вектором адресуемых выходных состояний (рис. 1) [6].

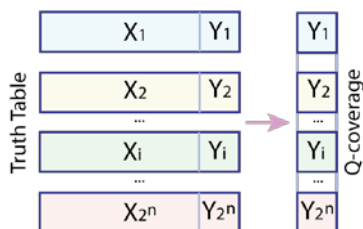


Рис. 1. Переход от таблицы истинности к Q-покрытию

Примитивизм и компактность кубитной векторной формы (см. рис.1) диктует применение только простых параллельных регистровых операций над его содержимым (not, shift, or, and, xor) для решения всех задач синтеза и анализа цифровых изделий.

3. Синтез кубитных моделей цифровых схем и минимизация кубитных покрытий

На рис. 2 представлена функция, аналитическая запись которой имеет вид:

$$f(x) = \sim X_2 \sim X_3 \vee X_1 X_2 X_3$$

$$1000 \vee 00000001 = \begin{matrix} & 1111 & 1111 & 0000 & 0001 & | & 0000 & 0001 & 0000 & 0001 \\ \text{a} & \boxed{1} & 0000 & 0000 & 0000 & 0000 & 1111 & 1111 & 1111 & 1111 \\ \text{д} & \boxed{2} & 0000 & 0000 & 1111 & 1111 & 0000 & 0000 & 1111 & 1111 \\ \text{р} & \boxed{3} & 0000 & 1111 & 0000 & 1111 & 0000 & 1111 & 0000 & 1111 \\ \text{е} & \boxed{4} & 0011 & 0011 & 0011 & 0011 & 0011 & 0011 & 0011 & 0011 \\ \text{с} & & 0101 & 0101 & 0101 & 0101 & 0101 & 0101 & 0101 & 0101 \end{matrix} \quad (1)$$

1) Записываем связь входных линий и выходного значения, получая таблицу истинности каждого элемента (табл.1).

2) Для каждого логического элемента составляем кубит или вектор Q -покрытия (табл.2).

3) Используя третий элемент логического или, формируем выходной вектор, используя кубиты заданных логических элементов 1000 и 00000001, которые являются входом для элемента 0111.

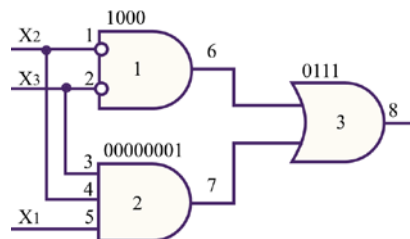


Рис. 2. Схема 1

4) Выполняем декартово произведение двух кубитов 1000 и 00000001.

Таблица 1. Таблица истинности элементов

3	2	6	3	4	5	7	6	7	8
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	1	1
1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	1

Таблица 2. Q-покрытия элементов

6	7	8
0	0	0
0	0	0
0	1	0
1	0	0
1	1	0
1	0	0
1	0	0
1	1	0
1	1	1

5) В итоге получаем результирующий вектор вида: 11111111 00000001 00000001 00000001. Для наглядности данный вектор разбит побайтово. Результат приведен в формуле:

6) Поскольку вектор получился 32-разрядным, то необходимо закодировать в двоичном коде 32 адреса от 00000... до 11111. Для этого записываем по вертикали адреса под каждым битом получившегося вектора.

7) Анализ схемы показывает, что линии 2 и 3 относятся к одному входу, как и линии 1 и 4. Так как одновременно на линиях не могут быть разные сигналы, то необходимо объединить линии в адресах 1,4 и 2,3.

8) В заданных линиях нужно оценить, где в них отличаются значения, и выделить их. После того как все столбцы с различающимися значениями будут выделены, смотрим на те столбцы, которые не были затронуты. Иначе говоря, используя операцию хог, необходимо обнаружить одинаковые значения и исключить различающиеся значения. После проведения операции хог получаем результирующий вектор: 11000001.

9) Для проверки результата необходимо посмотреть какой вектор получился на выходе линии 8. Стоит обратить внимание, что последовательность размещения входных сигналов имеет значение и она должна быть, согласно схеме, X_2, X_3, X_1 .

10) Проверяем результат, построив таблицу истинности (табл.3). Результат получается аналогичным 11000001, что показывает работоспособность данного метода.

Таблица 3. Таблица истинности элемента 0111

X_2	X_3	X_1	6	7	8
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Смоделируем исправное поведение на основе входного сигнала 100, используя структуры данных, представленных в табл. 4.

Таблица 4. Модель для анализа цифровой системы на основе структур данных

L	1	2	3	4	5	6	7	8
M	0	0	0	0	1	1	0	1
X	12	345	67
Q	1	0	0
	0	0	1
	0	0	1
	0	.
	0	.
	0	.
	1	.

Здесь представлены соответственно: L – вектор идентификаторов эквипотенциальных линий схемы цифровой системы, который ввиду своей тривиальности может быть исключен из модели, но при этом необходимо иметь число входных переменных устройства и общее количество линий; M – вектор моделирования состояний всех линий схемы; X – упорядоченная совокупность векторов входных переменных каждого примитива схемы, привязанных к номерам выходов; Q – совокупность q-покрытий примитивов, строго привязанных к номерам выходов и входным переменным примитивов.

Фактически метод позволяет реализовать синтез кубических покрытий для цифровых устройств. Адреса, по которым проводится операция хог, могут быть перенесены на память, поэтому скорость работы схемы увеличивается во много раз, так как процесс занимает не более нескольких тактов. На простых схемах разница между быстродействием может быть несущественной, но, взяв более сложную схему, можно увидеть разницу. К сожалению, более сложные схемы ручным методом довольно сложно описать, так как на примере схемы Шнейдера результирующий вектор будет 256-разрядным. Для того чтобы доказать работоспособность метода, на второй схеме (рис. 3) представлена другая, более сложная схема, аналитическая запись которой имеет следующий вид:

$$f = X_1 X_2 \vee (X_3 \text{ хог } X_4) \vee (X_1 \vee X_4).$$

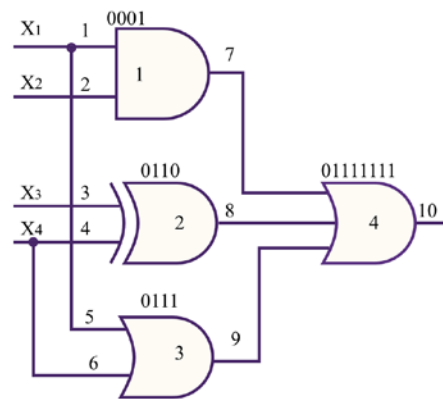


Рис. 3. Схема 2

1) Построение таблицы истинности (табл. 5).

Таблица 5. Таблица истинности элементов

1	2	7	3	4	8	1	4	9
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	0	1	1	1

2) Построение Q-покрытий (табл.6).

Таблица 6. Q-покрытия элементов

7		8		9	
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	0	1

3) Используя третий элемент логического или, формируем выходной вектор, используя кубиты заданных логических элементов 0001, 0110, 0111, которые являются входом для элемента 01111111.

4) Выполняем декартово произведение трёх векторов. В итоге получаем вектор вида: 01111111 11110111 01111111 11110111 11111111 11111111. Для наглядности данный вектор разбит побайтово. Результат приведен в формуле (2).

5) Вектор получился 64-разрядным, необходимо закодировать в двоичном коде 64 адреса от 000000... до 111111. Для этого записываем по вертикали адреса под каждым битом получившегося вектора.

6) Проанализировав схему, видим, что линия 1 и 5 относятся к одному входу, как и линии 4 и 6. Так как одновременно на линиях не могут быть разные сигналы, то необходимо объединить линии в адресах 1,5 и 4,6.

7) В заданных линиях нужно оценить, где в них отличаются значения, и выделить их.

8) Используя операцию хог, необходимо обнаружить одинаковые значения и исключить различающиеся. После проведения операции хог получается результирующий вектор: 0111 0111 1111 1111.

9) Построив таблицу истинности, видим, что результат получаем аналогичным 0111 0111 1111 1111, что является проверкой данного метода (табл.7).

В табл. 8 показано моделирование исправного поведения на основе входного сигнала 0011.

4. Алгоритм синтеза моделей

Для имплементации в программный сервис алгоритм будет иметь следующий вид (рис.4):

1. Получение таблицы истинности для каждого логического элемента, согласно входам и выходу, относящегося к элементу.
2. Составление кубита или вектора Q- покрытия для каждого логического элемента.

3. Формирование выходного вектора с помощью декартова произведения логических элементов.

4. Кодирование адресов в зависимости от разрядности результирующего вектора.

5. Изучение входов схемы и объединение линий, относящихся к одному и тому же элементу.

6. Обнаружение одинаковых значений и исключение отличающихся с использованием операции хог.

7. Проверка результата с помощью построения таблицы истинности на выход.

Таблица 7. Таблица истинности элемента 01111111

X ₁	X ₂	X ₃	X ₄	7	8	9	10
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	0	1	0	1
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	0	1	1
1	0	0	0	0	0	1	1
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	1
1	0	1	1	0	0	1	1
1	1	0	0	1	0	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	1

Таблица 8. Модель для анализа цифровой системы на основе использования кубитных структур данных

L	1	2	3	4	5	6	7	8	9	10
M	0	0	1	1	0	1	0	0	1	1
X	12	34	56	789
Q	0	0	0	0
	0	1	1	1
	0	1	1	1
	1	0	1	1
	1
	1
	1
	1

0001 v 0110 v 0111 =

01111111 11110111 01111111 11110111 01111111 11110111 11111111 11111111
 a 00000000 00000000 00000000 00000000 11111111 11111111 11111111 11111111
 д 00000000 00000000 11111111 11111111 00000000 00000000 11111111 11111111
 р 00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111 (2)
 е 00001111 00001111 00001111 00001111 00001111 00001111 00001111 00001111
 с 00110011 00110011 00110011 00110011 00110011 00110011 00110011 00110011
 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

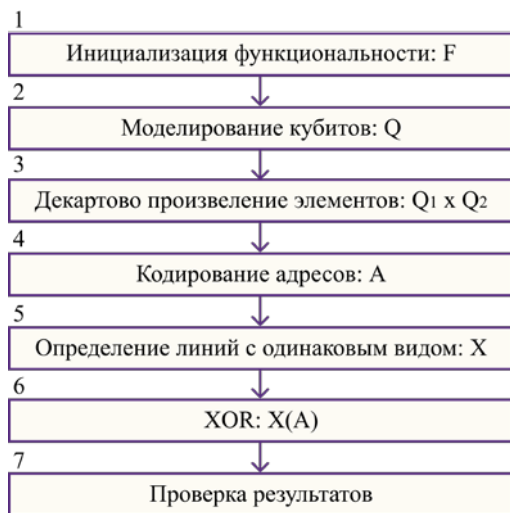


Рис. 4. Алгоритм синтеза моделей

5. Заключение

В результате исследования синтеза кубических покрытий для цифровых устройств был разработан алгоритм с существенно уменьшенным временем синтеза путем выполнения операций на матричной структуре памяти, без обращения к внешнему модулю АЛУ, которое значительно уменьшает быстродействие. Актуальность работы заключается в том, что вычисления строятся на векторе и выполняются за такт. Показано насколько технологичен и производителен данный алгоритм. Учитывая нехватку вычислительной мощности при обработке данных, к примеру, BigData, данный алгоритм поможет ускорить процессы обработки информации.

Литература: 1. *Almudever C. G., Lao TL., Fu X., Khammassi N., Ashraf I., Iorga D., Varsamopoulos S., Eichler C., Wallraff A., Geck L., Kruth A., Knoch J., Bluhm H., Bertels K.* The engineering challenges in quantum computing // 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne, Switzerland. 27-31 March 2017. DOI: 10.23919/DATE.2017.7927104. 2. *Lieven Vandersypen, Antoni van Leeuwenhoek.* 1.4 Quantum computing - the next challenge in circuit and system design // *Solid-State Circuits Conference (ISSCC), 2017 IEEE International.* San Francisco, CA, USA. 5-9 February. 2017. DOI: 10.1109/ISSCC.2017.7870244. 3. *Jasmeet Singh, Mohit Singh.* Evolution in Quantum Computing // *System Modeling & Advancement in Research Trends (SMART), International Conference.* Moradabad, India. 25-27 November 2016. DOI: 10.1109/SYSMART.2016.7894533. 4. *Travis S. Humble, Keith A. Britt.* Software systems for high-performance quantum computing // *High Performance Extreme Computing Conference (HPEC), 2016 IEEE.* Waltham, MA, USA. 13-15 September 2016. DOI: 10.1109/HPEC.2016.7761628 5. *Hahanov V., Litvinova*

E., Chumachenko S. Cyber Physical Computing for IoT Driven Services. *Springer International Publisher.* Switzerland. 2017. 150 p. 6. *Vladimir Hahanov, Svetlana Chumachenko, Eugenia Litvinova, Mykhailo Liubarskyi.* Qubit Description of the Functions and Structures for Computing // *Proc. of IEEE East-West Design and Test Symposium.* Yerevan. 14-17 October 2016. P.88-93.

7. <http://www.gartner.com/newsroom/id/3784363>.

Transliterated bibliography:

1. *Almudever C. G., Lao TL., Fu X., Khammassi N., Ashraf I., Iorga D., Varsamopoulos S., Eichler C., Wallraff A., Geck L., Kruth A., Knoch J., Bluhm H., Bertels K.* The engineering challenges in quantum computing. // 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne, Switzerland. 27-31 March 2017. DOI: 10.23919/DATE.2017.7927104.
2. *Lieven Vandersypen, Antoni van Leeuwenhoek.* 1.4 Quantum computing - the next challenge in circuit and system design // *Solid-State Circuits Conference (ISSCC), 2017 IEEE International.* San Francisco, CA, USA. 5-9 February. 2017. DOI:10.1109/ISSCC.2017.7870244.
3. *Jasmeet Singh, Mohit Singh.* Evolution in Quantum Computing // *System Modeling & Advancement in Research Trends (SMART), International Conference.* Moradabad, India. 25-27 November 2016. DOI: 10.1109/SYSMART.2016.7894533.
4. *Travis S. Humble, Keith A. Britt.* Software systems for high-performance quantum computing // *High Performance Extreme Computing Conference (HPEC), 2016 IEEE.* Waltham, MA, USA. 13-15 September 2016. DOI: 10.1109/HPEC.2016.7761628
5. *Hahanov V., Litvinova E., Chumachenko S.* Cyber Physical Computing for IoT Driven Services. *Springer International Publisher.* Switzerland. 2017. 250 p.
6. *Vladimir Hahanov, Svetlana Chumachenko, Eugenia Litvinova, Mykhailo Liubarskyi.* Qubit Description of the Functions and Structures for Computing // *East-West Design & Test Symposium (EWDTS), 2016 IEEE.* Yerevan, Armenia. 14-17 October 2016. P.88-93.
7. <http://www.gartner.com/newsroom/id/3784363>.

Поступила в редколлегию 17.05.2017

Рецензент: д-р техн. наук, проф. Хаханова И.В.

Бояджян Армен Гамлетович, студент 6 курса ХНУРЭ. Научные интересы: Internetofthings, квантовые вычисления. Адрес: Украина, 61166, Харьков, пр. Науки, 14, e-mail: z@fastlive.eu.

Котляров Антон Сергеевич, студент 6 курса ХНУРЭ. Научные интересы: Internetofthings, квантовые вычисления. Адрес: Украина, 61166, Харьков, пр. Науки, 14, e-mail: kotlyarovanton2@gmail.com.

Boiadzhian Armen, 6th year student of Kharkiv National University of Radioelectronics. Research interests: Internet of things, quantum computing. Address: Ukraine, 61009, Kharkiv, Nauki Ave., 14, e-mail: z@fastlive.eu.

Kotliarov Anton, 6th year student of Kharkiv National University of Radioelectronics. Research interests: Internet of things, quantum computing. Address: Ukraine, 61009, Kharkiv, Nauki Ave., 14.

КОМП'ЮТЕРНІЕ НАУКИ

УДК 004.4; 004.8

СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ – ОСНОВА СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЇХ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ЗАЯЦЬ В.М., ЗАЯЦЬ М.М.

На основі системного підходу (від простого до складного) робиться класифікація сучасних систем штучного інтелекту (СШІ), аналізуються способи подання даних і знань в СШІ, відзначаються особливості кожної з них та вказуються напрями доцільного застосування та подальшого розвитку. Проводиться порівняльний аналіз програмних засобів та мов програмування для їх реалізації.

Ключові слова: програмне забезпечення, системи штучного інтелекту, декларативні мови програмування, семантичні сітки, фрейми, продукційні системи, Лісп, Пролог, універсальні мови моделювання, логіко-лінгвістичні моделі.

Key words: software, artificial intelligence, declarative programming language, frames, Lisp, Prolog, universal modeling language

1. Вступ

З часу появи робіт зі штучному інтелекту їх розвиток пішов у двох напрямках.

Моделювання діяльності мозку, його психофізіологічних властивостей з метою відтворення на ЕОМ або за допомогою інших технічних засобів інтелекту чи штучного розуму – так званий біонічний напрям.

Імітація розумової поведінки людини з використанням засобів комп'ютерної техніки. Це основний прагматичний напрям розвитку робіт зі штучного інтелекту. За останні десятиліття досягнуто особливих успіхів. У цьому напрямку психофізіологічна діяльність мозку залишається в стороні. Тут комп'ютер розглядається як інструмент, на якому можуть бути написані добрі чи погані програми, що тією чи іншою мірою імітують розумову діяльність людини.

Поняття “штучний інтелект” в цьому напрямі розуміється метафорично, а дослідники зайняті розробленням програм, які дозволяють засобам комп'ютерної техніки відтворювати процеси, що є результатом розумової діяльності людини. Нажаль, на обчислювальну техніку нерідко поширюють ідею антропоморфізму. Можна часто зустріти вирази “машина прийняла рішення”, “машина розпізнає ситуацію”, “машина грає в шахи”.

Трапляються цілком невіглаські уявлення про “машини, що думають” в буквальному розумінні цих слів.

Насправді за всією цією “розумовою” діяльністю комп'ютера стоїть людина, яка для нього склала програму-інструкцію, реалізація якої і забезпечує імітацію розумової діяльності людини, але не більше.

Задачі, які розв'язуються в штучному інтелекті, є основою сучасних інформаційних технологій. Вчорашні інформаційні технології – це передовсім технології паперові, оскільки їх реалізація потребує колосальної кількості паперу. Традиційне використання комп'ютерної техніки в цьому плані становища не змінює. Старі інформаційні технології використовують комп'ютерну техніку через ряд посередників: фізиків, математиків, програмістів, спеціалістів з предметних областей. Знання про цьому в кожній предметній області фіксуються на паперових носіях.

У сучасних інформаційних технологіях спочатку, за літературними джерелами вивчають проблему, на її основі формують модель, створюють алгоритми її опрацювання на комп'ютері, відтак розв'язують задачу та аналізують її результати. При цьому забезпечується можливість отримання нових знань. Такі технології дозволяють можливість не лише ґрунтовно вивчити проблему, але й сприяють удосконаленню технічних і програмних засобів

У традиційних комп'ютерних інформаційних технологіях домінують моделі, які мають синтаксичний, описовий характер, а не суто процедурний, інструктивний характер. Для прикладу, математичні моделі у вигляді системи диференціальних рівнянь можуть описувати процеси в об'єктах найрізноманітнішої природи. Тобто для них характерний універсалізм. Якщо пред'явити користувачеві самі моделі (без коментарів), то він не зможе визначити, який об'єкт, які процеси описуються, а може зробити найзагальніші висновки про те, з якими класами об'єктів зіставимі ці моделі. Коментарі про конкретні предметні знання залишаються поза ЕОМ.

Становлення нової інформаційної технології зумовлено тим, що в теорії штучного інтелекту (а раніше в теорії управління, запропонованій академіком Д.А. Поспеловим) були розроблені логіко-

лінгвістичні моделі [2]. Вони дають змогу формалізувати конкретні змістовні знання про об'єкти і процеси, що в них відбуваються, і ввести в комп'ютер, поряд з математичними, логіко-лінгвістичні. До них належать семантичні сітки, фрейми, продукційні системи, які часто об'єднують як програмно-апаратні засоби систем штучного інтелекту (ПЗ СШ).

Мета даної роботи – на основі системного підходу від простого до складного провести класифікацію сучасних систем штучного інтелекту, дати їх порівняльну характеристику, проаналізувати способи подання даних і знань та вказати напрями подальшого їх вдосконалення.

2. Особливості сучасних систем штучного інтелекту

Системи штучного інтелекту, які функціонують за принципами нової інформаційної технології, поділяють на:

- Інтелектуальні інформаційно-пошукові системи (системи типу питання-відповідь), які забезпечують в процесі діалогу взаємодію користувача-непрограміста з базами даних і знань на професійній мові користувача, близькій до природної.
- Розрахунково-логічні системи, які дають змогу користувачам, що не є професійними програмістами і фахівцями в прикладній математиці, розв'язувати в діалоговому режимі професійні задачі, використовуючи складний математичний апарат і відповідне програмне забезпечення.
- Експертні системи, що дозволяють комп'ютеризувати області, в яких знання можна подати в експертній описовій формі, але використання математичних моделей, характерних для точних наук, ускладнене або й взагалі неможливе.

Саме завдяки експертним системам штучний інтелект набув стратегічного значення в розвитку науки і управління, проектування і екології, космосу і фізики Землі, освіти і матеріалознавства. Особливого значення набувають так звані гібридні експертні системи – об'єднання традиційних експертних систем з розрахунково-логічними, тобто в гібридних експертних системах логіко-лінгвістичні моделі використовуються поряд з математичними.

Зазначимо ще одну особливість нової інформаційної технології – адаптацію і гнучкість програмних систем відносно задач з тієї чи іншої предметної області. У всіх цих випадках не можна априорі сказати, за допомогою якої моделі, алгоритму і системи розв'язується задача чи розпізнається ситуація.

Так, коли встановлюють медичний діагноз, невідомо заздалегідь, яке правило буде використано з наявної бази знань. Це характерно для всіх інтелектуальних пакетів прикладних задач.

Отже, всі системи штучного інтелекту орієнтовані на знання, а тому подальший прогрес систем штучного інтелекту і комп'ютерних інформаційних технологій визначається розвитком трьох основних теоретичних проблем:

- представлення знань – центральна проблема штучного інтелекту;
- комп'ютерна лінгвістика – забезпечує процес природно-мовного спілкування з ЕОМ та процес автоматизованого перекладу текстів з іноземних мов;
- комп'ютерна логіка, яка має особливе значення для розвитку експертних систем. Мета її – моделювання людського мислення і перетворення програмування з мистецтва в науку.

Штучний інтелект можна визначити як комплексну науково-технічну проблему, для розв'язання якої дослідження ведуться у чотирьох напрямках:

- а) моделювання на комп'ютері окремих функцій творчих процесів: ігрові задачі, автоматичне доведення теорем, автоматичний синтез програм, аналіз музичних творів, розпізнавання та ідентифікація складних динамічних об'єктів;
- б) зовнішня інтелектуалізація комп'ютерів: фундаментальні дослідження із створення діалогового інтерфейсу (зовнішня в тому розумінні, що інтелектуальні системи на початковому етапі створюють на ЕОМ існуючої архітектури, враховуючи комп'ютери неймановської структури);
- в) внутрішня інтелектуалізація комп'ютерів – це створення комп'ютерів нової архітектури, основаної на принципах штучного інтелекту і призначеної для створення інтелектуальних систем;
- г) цілеспрямована поведінка роботів, що здатні автономно вирішувати завдання, які поставила людина.

Перш ніж аналізувати проблеми штучного інтелекту, розглянемо, як науково-технічний прогрес привів до становлення нових інформаційних технологій.

В основу науково-технічного прогресу покладено зростання знань, яке відбувається швидше, ніж за експонентою.

Якщо $x(t)$ – маса знань, то, вважаючи, що знання зростають за законом експоненти, маємо

$$x(t) = px(t),$$

де p – постійна величина. Звідси

$$x(t) = x(0)e^{pt}.$$

У наукознавстві вживають такий термін, як період подвоєння знань T . Безсумнівно, P і T є функціями часу і повинні бути уточнені. Величина T постійно падає, а P – зростає.

Дані про зміну T і P , як видно з таблиці, показують, що з 50-х років минулого століття спостерігається інформаційний вибух, який виявляється у збільшенні самого показника експоненти.

Дані про період подвоєння знань та коефіцієнт пропорційності

Рік	1900	1950	1970	1981	1990	2000
T (роки)	50	10	5	2.5	1.2	0.6
P (рік-1)	0.014	0.07	0.14	0.28	0.55	1.15

Зростання знань приводить до ускладнення народного господарства. Для ефективного управління господарськими механізмами виникає потреба в комп'ютеризації народного господарства та науки, яка має в наші дні таке саме значення, як індустріалізація країни в 20 – 30-ті роки ХХ ст.

Комп'ютеризація означає масове використання обчислювальної техніки, а тому професійних програмістів не вистачає, щоб забезпечити масове використання ЕОМ у народному господарстві. Традиційні комп'ютерні технології характеризуються наявністю посередників між ЕОМ і користувачами, до яких належать аналітики, програмісти, постановники задач, оператори ЕОМ.

Крок до позбавлення від посередників – це введення термінального доступу до ЕОМ, що значно зменшує кількість посередників у ланці ЕОМ - користувач. Для ефективного розв'язання своїх задач користувач повинен здійснити ще один крок – забезпечити прямий доступ до комп'ютера.

Інший шлях забезпечення прямого доступу до комп'ютера полягає у створенні засобів програмного забезпечення, які реалізуються всередині комп'ютерів, тобто через створення інтелектуального інтерфейсу ЕОМ. Його можна реалізувати на основі логіко-лінгвістичних моделей та інших методів й ідей, розроблених в області штучного інтелекту. Досягнення в області штучного інтелекту настільки значні, що якщо раніше йшлося про ін-

дустрію обробки даних, то тепер йдеться про індустрію інтелектуальних систем.

З використанням обчислювальної техніки пов'язані такі поняття, як програми та дані: на початкових етапах програміст розробляв програму, а потім вводив у неї дані. Відтак сталися значні зміни – дані були відділені від програми. З'явилися дані у вигляді різних структур: реляційні (табличні), мережі, ієрархічні та системи управління базами даних (СУБД), банки даних та знань.

Для відділення даних від програм використовувалися засоби опису даних, які містяться в мовах програмування. У таких мовах, як Фортран та Алгол розвивались порівняно нескладні засоби опису структур даних. Складніші засоби містять мови Паскаль, Сі, Сі++. Паралельно розвивалися способи запису даних на зовнішніх носіях інформації. Тут фундаментальним стало поняття файлу, що має ім'я і містить всі необхідні структуровані записи даних про різні об'єкти, з якими працює система.

Отже, файл можна розглядати як інформаційну модель об'єкта.

У процесі подання даних на зовнішніх носіях можна виділити такі етапи:

- способи формування записів даних у файлах, ведення файлів; організацію доступу до них повністю визначають користувачі;
- керування файлами та доступ до них забезпечується операційною системою;
- створення баз даних і систем управління ними з метою їх оптимальної обробки в інтересах усього підприємства.

З появою нових інформаційних технологій з'явилися і нові способи подання даних і програм у США.

Таким чином, можна зробити висновок, що нові інформаційні технології є основою сучасних США, розвиток яких своєю чергою є могутнім стимулом для створення як нового програмного забезпечення, так і появи нових мов програмування та засобів для їх ефективно технічної реалізації.

3. Способи подання даних та знань в США

Сьогодні можна говорити про новий етап представлення даних у пам'яті комп'ютера – створення інформаційно-розрахункових мереж, а на їх основі – розподілених баз колективного користувача.

З появою СШ з'явилися і нові поняття, такі як база знань. Безумовно, що дані й структура бази даних певною мірою відображають знання про предметну область і її структуру. Проте специфічними ознаками, що відрізняють знання від даних, пов'язаних з їх поданням у комп'ютері, можна вважати такі чотири ознаки [2, 22]:

- внутрішня інтерпретованість;
- структурованість;
- зв'язність;
- активність.

Якщо звернутися до структур даних, то деякі із ознак, властиві знанням, будуть справедливими і для даних. Так, інтерпретованість явно простежується в реляційній базі даних, де імена стовпців є атрибутами відношень, імена яких вказані в рядках.

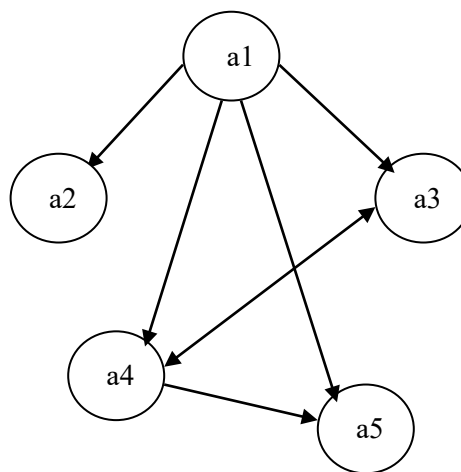
Другу ознаку – структурованість можна подати декомпозицією складного об'єкта на простіші і встановленням зв'язків між простими. Це зв'язки типу відношень клас-підклас, вид-підвид, список-підсписок. Ці зв'язки можна спостерігати в сіткових та ієрархічних моделях даних. Ці самі відношення при бажанні можна задати і в табличних базах даних.

Що стосується третьої ознаки – зв'язність, то практично неможливо знайти аналогів у згадуваних базах даних, оскільки наші знання зв'язані не лише в розумінні структури. Вони відображають закономірності у фактах, процесах, явищах та причинно-наслідкові відношення між ними.

Стосовно четвертої ознаки (активність) склалася така ситуація, що при використанні комп'ютерів нові знання породжуються програмами, а дані пасивно зберігаються в пам'яті комп'ютера. Людині властива пізнавальна активність. І це принципово відрізняє знання від даних. Таким чином, протиріччя у знаннях стимулюють здобуття нових знань. Таким же стимулом до знань є неповнота. Загалом, не завжди вдається розмежувати дані і знання. Однак вони існують і привели до появи спеціальних формалізмів у вигляді моделей подання знань в комп'ютері, які відображають всі чотири ознаки знань. Сьогодні є, принаймні, чотири види моделей і відповідних їм мов представлення знань: мови (моделі) семантичних мереж; системи фреймів; логічні мови (моделі) та продукційні системи.

Розглянемо моделі семантичних мереж. У семантичній сітці вершинам поставлені у відповідність

поняття (об'єкти, події, процеси, сигнали), а дугам – відношення на множині понять. З цієї причини мову семантичних мереж ще називають реляційною. Тут досить виразно проступає така ознака знань як зв'язність. У семантичній сітці можна відобразити знання, які виражені на природній мові. Візьмемо речення: “Мисливець сів на пліт, переїхав на другий берег, поклав у корзину качку і взяв корзину з качкою.” Можна виділити п'ять об'єктів: мисливець (a1), пліт (a2), другий берег (a3), корзина (a4), качка (a5). Ці об'єкти зв'язані відношеннями: сів на (Г1), переїхав на (Г2), поклав у (Г3) і взяв (Г4). Сітку, що відповідає цьому реченню, показано на рисунку.



Семантична мережа подання речень природною мовою

Згідно з логікою реального світу ми можемо доповнити вихідний текст відношеннями, які явно не містяться в тексті. Доповнений текст матиме вигляд: Мисливець сів на пліт і на плоті переїхав на другий берег. На другому березі була качка. Мисливець поклав качку в корзину. Мисливець взяв корзину з качкою.

Треба вказати на одну важливу обставину: в індоевропейських мовах, що підтвердили експериментальні дослідження, є не більше двохсот відношень, що не зводяться одне до одного. Комбінації цих відношень дозволяють виразити інші відношення, які фіксуються в текстах. Ця обставина дозволяє сподіватися, що будь-яку предметну область можна описати за допомогою скінченної множини відношень і здійснити автоматичну по-

будову семантичних мереж безпосередньо з тексту.

Частковим випадком семантичних мереж є сценарії або однорідні сітки. Це сітки, в яких об'єкти зв'язані єдиними відношеннями строгого або не строгого порядків з різною семантикою.

Для прикладу, якщо об'єктами-поняттями будуть роботи різного виду (чи окремі операції деякої роботи), а єдиними відношеннями строгого порядку – відношення слідування, то ми отримаємо сітковий графік комплексу робіт. Очевидно, сценарії є зручним засобом складання планів.

Семантична сітка загального виду дає змогу зрозуміти відмінності між поняттями бази даних і бази знань. Предметна область – це множина допустимих станів її компонент. Множина, представлена через загальні поняття і відношення між ними, утворює базу знань у вигляді інтенціональної семантичної сітки. З іншого боку, залежно від ситуації, компоненти предметної області мають конкретне значення, властивості, характеристики. Всі ці конкретні значення будуть відображені в так званій екстенціональній семантичній сітці або бази даних природної структури.

Ще один важливий аспект семантичних мереж – це представлення в них елементів загального виду за допомогою логічних міркувань та використання прикладних програм. Ці елементи можна реалізувати використовуючи так звані віртуальні відношення.

Але слід зазначити, що семантичні сітки, незважаючи на їх різноманіття, мають ряд недоліків. Надто довільна структура і різні типи вершин та зв'язків потребують потужного програмного забезпечення для їх обробки. Це зумовило появу різних модифікацій семантичних мереж: синтагматичні ланки, сценарії, фрейми тощо. Коротко зупинимося на фреймових моделях.

Термін фрейм (рамка) асоціюється з природним уявленням людини про об'єкт як щось довершене, обрамлене. Формально під фреймом розуміють структуру такого вигляду:

$$f[<v_1, g_1>, <v_2, g_2>, \dots, <v_k, g_k>],$$

де f – ім'я фрейма; пара $<v_i, g_i>$ – це i є слот, де v_i – ім'я слота, а g_i – його значення.

Часто фрейми, свою чергою, поділяють на дві групи: фрейми-описи та рольові фрейми. Приклади фреймів:

1) описовий фрейм:

[<фрукти>, <виноград, болгарський 20 кг>, <вишня, володимирська 200 кг>];

2) рольовий фрейм:

[<перевезти>, <що, прокат 300 кг>, <куди, Москву 200 кг>, <звідки, Череповець>, <чим, залізничним транспортом>].

Якщо в наведених прикладах і в загальному виразі забрати всі значення слотів, то отримаємо фрейм, який названо прототипом фрейму: фреймом-інтенціоналом. Фрейми з конкретними значеннями слотів називаються фреймами-прикладними, фреймами-екземплярами.

Фрейми властивість вкладеності, тобто значення слоту може виступати ціла система імен слотів глибшого рівня. Така структура запису фреймів дає змогу успішно оперувати з гігантськими масивами даних, реалізовувати і ефективно управляти базами даних та створювати нові бази знань. Очевидно, для ефективного використання таких вкладених даних в одну структуру слід використовувати мови програмування та відповідне програмне забезпечення, що оптимально пристосоване для цих цілей.

4. Характеристика програмних засобів та мов програмування для реалізації СШ

Для розроблення та реалізації сучасних СШ розроблено відповідні технічні та програмні засоби, які можуть бути реалізовані за допомогою спеціальних мов програмування, які пристосовані для цих цілей. Для створення СШ найбільш вдалим є декларативні мови програмування, які задають не послідовність виконання дій для досягнення поставленої мети, а описують об'єкт дослідження і взаємозв'язки між його складовими. В результаті цього маємо логічну або функціональну модель об'єкта досліджень. Використовуючи вбудовані механізми опрацювання даних і програм в таких мовах, можемо отримувати нові знання про об'єкт дослідження, оперативно аналізувати його динаміку і управляти його поведінкою.

В основу сучасних мов логічного програмування покладено мову Пролог [7-13], що з'явилася в 1970

році водночас з такими поширеними мовами як Паскаль і Сі. Орієнтація мов логічного програмування – це нетрадиційне застосування комп'ютерної техніки: розуміння природної мови, створення баз знань, інтелектуальних систем прийняття рішень [4, 22-25], систем розпізнавання об'єктів, експертних систем [1, 3, 5, 6], систем перекладу текстів, інформаційно-аналітичних систем та інших задач, які прийнято відносити до проблем штучного інтелекту [2, 5, 22, 33]. Принципова відміна мови Пролог від традиційних мов у тому, що програма на мові логіки описує не процедуру розв'язання задачі, а створює логічну модель предметної області – деякі факти відносно властивостей даної предметної області і відношення між цими властивостями, а також правила вислуду нових властивостей і відношень на основі заданих. Отже, Пролог є описовою мовою, а не процедурною. Основні поняття мови досвідченими програмістами сприймаються без особливих затруднень, а от практичне втілення цих знань в корисні і ефективні програми викликає певні ускладнення.

В основу сучасних мов функціонального програмування покладено мову Лісп, яка запропонована Дж. Маккарті [32] в 1960 р. і орієнтована на розв'язання задач нечислового характеру. Англійська назва мови Lisp є скороченням виразу опрацювання списків і підкреслює область її застосування. У вигляді списків зручно відображати алгебраїчні вирази, множини, графи, правила виводу, граматику природних мов та інші складні об'єкти. Списки є однією з найбільш гнучких форм представлення інформації в пам'яті комп'ютера.

Після появи Ліспу [13-19] рядом авторів розроблено інші алгоритмічні мови, орієнтовані на розв'язання задач штучного інтелекту, серед яких слід відзначити Пленер, Снобол, Рефал, Пролог [5]. Тим не менше мови функціонального програмування успішно розвиваються і майже за сорокарічну історію існування з'явилося ряд діалектів функціональної мови: Common Lisp [14], MacLisp, R-Lisp [21], InterLisp, Standart Lisp, Reduce [22] та ряд інших.

Відзначимо, що відміни між ними не носять принципового характеру і зводяться, в основному, до несуттєвої різниці у формі запису програми та набору вбудованих функцій. Тому програміст, який освоїв одну із мов функціонального програмування, без особливих зусиль в стані освоїти будь-яку іншу, володіючи основними методами та алгоритмами побудови функціональних програм.

Незважаючи на очевидні переваги декларативних мов перед процедурними, довгий час ці мови роз-

вивалися і застосовувалися у вузькому колі спеціалістів з проблем штучного інтелекту. Лише протягом останніх двох десятиліть декларативні мови програмування викликали зацікавлення широкого кола спеціалістів з інформатики. Зростання популярності цих мов зумовлено виходом теорії штучного інтелекту в сферу прикладного програмування.

На сьогоднішній день з'явилися престижні національні проекти створення ЕОМ нових поколінь, в яких інтелектуальний інтерфейс, орієнтований на користувача-непрофесіонала в інформатиці, займає центральне місце.

Пролог був прийнятий як базова мова в японському проекті ЕОМ п'ятого покоління, який орієнтований на дослідження методів логічного програмування і штучного інтелекту.

Теоретичною основою Прологу є символічна логіка, яка дає механізм обчислення предикатів. Цій мові притаманний ряд властивостей, які відсутні в традиційних мовах програмування. До них відносяться: механізм логічного доведення з пошуком і поверненням, вбудований механізм співставлення зі зразком, проста і виразна структура даних з можливістю її зміни. Пролог відрізняє однакову структуру даних і програм. Оскільки вони не відрізняються, то їх можна змінювати під час роботи Пролог-системи. В Пролозі, як і у функціональних мовах, відсутні вказівники, оператори присвоєння та переходу. Природним методом програмування тут є рекурсія. Ряд важливих рис теорії програмування впливає з декларативності мов, що дозволяє розуміти програму не досліджуючи динаміки її виконання. Однак істинна потужність цих мов ґрунтується не на одній якійсь властивості, а в сукупному і грамотному поєднанні всіх. Мови декларативного характеру орієнтовані не на розробку розв'язків, а на систематичний і формалізований опис задачі з тим, щоб розв'язок впливав із складеного опису. В сучасному програмуванні позначився перехід від пошуку розв'язку до визначення того, який розв'язок потрібний. Якщо мови логічного програмування ідеально пристосовані для реалізації процедури логічного висновку, то функціональні мови [14-21] дозволяють будь-яку синтаксично правильно побудовану програму на довільній мові програмування записати у вигляді функції, тим самим оптимальним чином побудувати інтерпретатор досліджуваної мови програмування.

5. Системний підхід до освоєння та застосування декларативних мов програмування

Традиційні (алгоритмічні) мови програмування порівняно з декларативними (описовими) є доволі об'ємні і громіздкі, оскільки не дають змоги:

- максимально використовувати можливості сучасної комп'ютерної техніки для забезпечення ефективності програмних засобів;
- ясно і наочно відображати алгоритми програми, щоб забезпечити легкість перевірки та модифікації останніх.

Строго функціональні та мови логічного програмування, що відносяться до декларативних мов програмування [7-21], є доволі простими, а тому забезпечують достатньо високий ступінь виразності програм, порівняно з традиційними мовами (Algol, Fortran, Pascal, C, Delphi, C++). Ряд функціональних програм можуть ефективно працювати на сучасних комп'ютерах, тим не менше не так ефективно, як відповідні програми з оператором присвоєння. Це пов'язано зі структурою архітектури сучасних комп'ютерів. Окрім того, вибір дещо іншої структури представлення даних, ніж це прийнято в інструментальному Ліспі, забезпечує як більшу ясність представлення програм, так і підвищення їх ефективності з використанням сучасних комп'ютерів старої архітектури.

З одного боку, сучасні мови програмування повинні ефективно використовувати сучасні машини, а з іншого – ясно виражати програмні алгоритми, щоб полегшити перевірку останніх. Строго функціональна мова, будучи простою за своєю структурою, демонструє вищий ступінь виразності порівняно з традиційними мовами, де існує оператор присвоєння. Це зв'язано, великою мірою, зі способом вибору структур даних. Дещо інший їх вибір можна забезпечити підвищенням ефективності функціональних програм і на сучасних комп'ютерах. Одна з фундаментальних властивостей мови програмування, що дає можливість ясно описати обчислення на цій мові, – простота семантики мови. Велика перевага функціональної мови, як і логічної, в тому, що тут є кілька основних понять, кожне з яких має просту семантику. Зокрема, семантика функціональної мови розуміється в термінах значень, які мають вирази, але аж ніяк не в термінах дій і послідовності їх використання. Але з практичного погляду було б справедливіше зробити висновок, що строго функціональна мова є надзвичайно елементарною і її розширення значно б підвищили ефективність і ясність деяких класів вирахань. Очевидно, необхідно розрізнити суто синтаксичні розширення і розширення, які вима-

гають зміни семантики мови. Зміна семантики вимагає обережності, оскільки це ускладнює розуміння (ясність) вже відлагоджених функціональних програм.

Будь-яку систему позначень для опису алгоритмів і структури даних можна назвати мовою програмування, хоча, як правило, вимагається ще й реалізація мови на комп'ютері.

Сьогодні розроблено сотні різних мов програмування. Ще в 1969р. Саммет [32] наводить список з 120 мов, які доволі широко застосовуються. Ця приголомшлива кількість мов програмування протирічить тому, що більшість програмістів у своїй практиці використовують декілька мов програмування, а значна їх частина – одну або дві мови. Виникає питання про доцільність освоєння різних мов програмування, якщо навряд чи буде можливість їх реалізації. Тим не менше, якщо не обмежуватися поверховим ознайомленням з мовою, а мати глибокі уявлення про поняття, які лежать в основі конструювання програм на даній мові, то поза всяким сумнівом можна переконатися у доцільності освоєння різних мов програмування, виходячи з таких міркувань:

1. Внаслідок вивчення різних мов програмування покращується розуміння конкретної мови, її понять та основних методів і прийомів, що в ній використовуються. Типовим прикладом може бути рекурсія. При правильному її використанні можна отримати елегантну ефективну програму, а застосування її до простого алгоритму могло б привести до астрономічного збільшення часових затрат. З іншого боку, недоступність використання рекурсії в таких мовах, як Фортран, Кобол та розуміння основних принципів і методів реалізації рекурсії може внести ясність в певні обмеження мови, які, на перший погляд, є надуманими.
2. Значення мов програмування розширює запас корисних програмістських конструкцій і сприяє розвитку мислення. Працюючи з структурами даних однієї мови, виробляють і відповідну структуру мислення. Вивчаючи конструкції інших мов та методи їх реалізації, розширюють програмістський тезаурус.
3. Знання великої кількості мов програмування дає змогу обгрунтовано вибирати певну мову для розв'язання конкретної задачі.
4. Освоєння нової мови програмування, як і природної мови людського спілкування, завжди легше, якщо відомі декілька мов.
5. Знання принципів побудови різноманітних мов програмування полегшує розробку нової мови.

Конструювання сучасних мов програмування сьогодні далеко від досконалості. Кожна з відомих мов має свої недоліки та переваги.

Більшість сучасних мов програмування є універсальними, оскільки дають змогу записати будь-який алгоритм цією мовою, якщо не накладати обмежень на час виконання програми та місткість пам'яті, алгоритмічну складність тощо. Якщо хто-небудь запропонує нову мову програмування, то вона, очевидно, буде універсальною, якщо ігнорувати обмеження на пам'ять або час. Порівнюючи різні мови програмування, слід виходити не з кількісного співвідношення того, що вони дозволяють зробити, а з якісних відмінностей, що визначають елегантність (короткість і наочність), легкість (прозорість) та ефективність (швидкодія та технічні засоби) програмування на них. Це порівняння слід здійснювати в контексті конкретної сфери застосування.

Очевидно для детального розуміння особливостей, синтаксису, семантики, діапазону прикладних застосувань декларативних мов програмування (Lisp [14-21, 24, 28], Prolog [7-13, 28], UML [31-33]), які покликані обслуговувати задачі штучного інтелекту [2, 4-6, 8, 10-12, 22, 33], при їх вивченні слід застосовувати системний підхід. Суть підходу зводиться до викладу матеріалу від простого до складного з наведенням численних прикладних задач, які в змозі розв'язати на даній мові програмування. Поряд з цим слід чітко уявляти обмеження, які накладені на використовувану мову програмування та сферу її прикладних застосувань.

При системному підході [30] забезпечується можливість постійного порівняння різних описових мов програмування [27], постійна зміна і оновлення прикладних задач та освоєння нових можливостей логічних предикатів у Пролозі чи функціональних описів в Універсальному Ліспі.

Авторам видається, що при викладанні декларативних мов програмування такий системний підхід можна забезпечити шляхом викладання лекційного, практичного матеріалу чи постановки лабораторних робіт у вигляді аплікацій, виконаних на мові програмування UML [31] чи іншій об'єктно-орієнтованій мові моделювання [5, 6, 12, 22-29], яка забезпечує можливість як постійного поновлення підготовленого матеріалу, так і постійного порівняння з іншими підходами, їх оптимізації і подальшого вдосконалення.

Можна дійти висновку, що досягти прогресу для отримання і освоєння нового інформаційного матеріалу можна лише у процесі постійного удосконалення та оновлення набутих знань, вмінь та на-

вичок. Коли такого матеріалу накопичується значний обсяг, то без системного його впорядкування від простого до складного, класифікації за алгоритмічною складністю, функціональними можливостями та простотою подання і зберігання даних подальший розвиток неможливий.

6. Висновки

При побудові СШ доцільно застосовувати системний підхід, рухаючись від простого і очевидного до складного і малоімовірного на перший погляд. При цьому слід враховувати особливості як самої системи, так і програмних засобів, які розробляються або використовуються.

Видається доцільним, поряд з процедурними, застосування декларативних мов програмування, зокрема, логічного та функціонального, які дозволяють будувати логічну модель об'єкта досліджень та застосовувати ефективні механізми її аналізу й експлуатації.

Для ефективного подання даних і програм найбільш продуктивними є логіко-лінгвістичні моделі [2, 22, 32] (семантичні мережі; системи фреймів; логічні моделі; продукційні системи; функціональне подання [28]; реляційні бази і банки даних та знань).

Суттєвий прогрес в галузі створення СШ неможливий без побудови принципово нових комп'ютерів, які відходять від неймановської структури, а пов'язані з нейронною структурою [34], що наближена до природних потреб користувача.

Література: 1. Вендров А. М. CASE-технологии: современные методы и средства проектирования информационных систем / А. М. Вендров. М.: Финансы и статистика, 1998. 175 с. 2. Поспелов Г. С. Искусственный интеллект – основа новой информационной технологии / Г. С. Поспелов. К.: Наукова думка, 1995. 360 с. 3. Заяць В. М. Методи, алгоритми та програмні засоби для моделювання і аналізу динаміки складних об'єктів на основі дискретних моделей: Монографія / В. М. Заяць. Львів: Новий світ – 2000, 2009. 400 с. 4. Катренко А. В. Теорія прийняття рішень: Підручник / А. В. Катренко, В. В. Пасічник, В. П. Пасько. К.: Видавнича група ВНУ, 2009. 448 с. 5. Робертс Ф. С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. М.: Наука, 1986. 386 с. 6. Томашевський В. М. Моделювання систем: Підручник / В. М. Томашевський. К.: Видавнича група ВНУ, 2004. 352 с. 7. Клоксин У., Меллиш К. Программирование на языке Пролог. М.: Мир, 1987. 336 с. 8. Хоггер К. Введение в логическое программирование. М.: Мир. 1988. 348 с. 9. Братко И. Программирование на языке Пролог. 10. Макаллистер Дж. Искусственный интеллект и Пролог на микро-ЭВМ, М.: Машиностроение, 1990. 240 с. 11. Ин Ц., Соломон Д. Использование Тур-

бо-Пролога: Пер. с англ. М.: Мир, 1993. 608 с. **12.** Бакаев А.А., Гриценко В.И., Экспертные системы и логическое программирование. К.: Вища школа, 1995. 246 с. **13.** Заяць В.М. Логічне програмування: Ч. 1: Конспект лекцій з дисципліни «Логічне програмування» для студентів базового напрямку 6.08.04 «Програмне забезпечення автоматизованих систем». Львів: Видавництво Національного університету «Львівська політехніка», 2002. 48 с. **14.** Хендерсен П. Функциональное программирование. Применение и реализация. М.: Мир, 1983. 349 с. **15.** Бадаев Ю.И. Теория функционального программирования. Мови CommonLisp та AutoLisp. Київ, 1999. 150 с. **16.** Заяць В.М. Конспект лекцій з курсу функціональне програмування. Львів, 1999. 55 с. **17.** Маурер У. Введение в программирование на языке Лисп. М.: Мир, 1976. 104 с. **18.** Хювенен Э., Слептян Й. Мир Лиспа. Т.1. Введение в язык Лисп и функциональное программирование: Пер. с финск. М.: Мир, 1990. 447 с. **19.** Хювенен Э., Слептян Й. Мир Лиспа. Т.2 Методы и системы программирования. Пер. с финск. М.: Мир, 1990. 447 с. **20.** Еднерал В.Ф., Крюков А.П., Родионов А.Я. Язык аналитических вычислений REDUCE. М.: Из-во МГУ, 1984. 176 с. **21.** Лавров С.С., Силигадзе Г.С. Автоматическая обработка данных. Язык Лисп и его реализация. М.: Наука, 1978. 176 с. **22.** Уинстон П. Искусственный интеллект. М.: Мир, 1980. 513 с. **23.** Грин Д., Кнут Д. Математические методы анализа алгоритмов. М.: Мир, 1987. 120 с. **24.** Фостер Дж. Обработка списков.. М.: Мир, 1974. 280 с. **25.** Вирт Н. Алгоритмы + структуры данных = программы: Пер. с англ. М.: Мир, 1985. 213 с. **27.** Бердж В. Методы рекурсивного программирования. М.: Машиностроение, 1983. 248 с. **28.** Заяць В.М., Заяць М.М. Логічне і функціональне програмування. Навч. посібник. Рівне: 216. 400 с. **29.** Erich Gamma, Richard Helm, Ralih Johnson, John Vlissides. Wzorce projektowe, 2012. **30.** Заяць В.М., Маєвський Яцек. Системний підхід до освоєння та викладання мов логічного і функціонального програмування //21-й Міжнародний молодіжний форум «Радіоелектроніка і молодь в XXI столітті». Харків, ХНУРЕ., 2017. С. 4-6. **31.** Stanislaw Wrycha i inni. UML 2.1. Cwiczenia. ISBN: 978-83-246-0612-2012. **32.** Maccarthy J. Recursive functions of symbolic expressions and their computation by machine // Comm. ACM.: 1960. Vol.3. P. 184-195. **33.** Nilsson N.J. Principles of Artificial Intelligence. Tioga. Springer-Verlag, 1960. 164 с. **34.** Бондаренко М. Ф., Хаханов В. И., Гузь О. А., Шабанов-Кушнарченко Ю. П. Инфраструктура мозкоподібних обчислювальних процесів. Харків: ХНУРЕ, 2010. 169 с.

Надійшла до редколегії 03.05.2017

Рецензент: д-р фіз.-мат. наук, проф. Яворський І.М.

Заяць Василь Михайлович, д-р техн. наук, професор Технологічно-Природничого Університету (м. Бидгощ, Польща). Наукові інтереси: числові методи в прикладних застосуваннях, математичне і комп'ютерне моделювання динамічних систем, розпізнавання об'єктів зі складною динамічною природою, електрорадіотехніка та інформатика. Захоплення і хобі: туризм, велоспорт, театр; щоденні піші прогулянки 7 км. Адреса: Україна, 79070, Львів, просп. Червона Калина, 39, кв. 4. Тел.: дом. (+38) (032) 222-43-48; роб.: (+48) 52-340-53-30; моб.: (+38)(032) 0678362473.

Заяць Марія Михайлівна, ст. викладач НУ «Львівська політехніка» (м. Львів, Україна). Наукові інтереси: інформаційні технології, інженерія знань, математичне і комп'ютерне моделювання складних об'єктів, логічне програмування. Захоплення і хобі: туризм, театр, якісне приготування вишуканих страв. Адреса: Україна, 79070, Львів, просп. Червона Калина, 39, кв. 4. Тел.: дом. (+38) (032) 222-43-48; роб.: (+38) (032) 258-25-38 моб.: (+38)(032) 0989327526.

Zaiatc Vasyl M., DSc., Professor, Professor University of Science and Technology (m. Bydgoszcz, Poland). Research interests include numerical methods in applied applications, mathematical and computer modeling of dynamic systems, recognition of objects of complex dynamic nature, electro - radiotehnika and informatics. Hobbies: hiking, biking, theater; daily walks 7 km. Address: Ukraine, 79070, Lviv, avenue Chervona Kalyna, 39, flat 4. Tel.: Home (38) (032) 222-43-48; Job: (+48) 52-340-53-30; Mob.: (+38) (032) 0678362473.

Zaiats Mary M., senior teacher National University "Lviv polytechnic" (m. Lviv, Ukraine). Research interests include information technology, engineering knowledge, mathematical and computer modeling of complex objects, logic programming. Interests and hobbies: hiking, theater, high-quality gourmet food. Address: Ukraine, 79070, Lviv, avenue Chervona Kalyna, 39, flat 4. Tel.: Home. (38) (032) 222-43-48; Job.: (+38) (032) 258-25-38 Mob.: (+38) (032) 0989327526.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.9

BIG DATA: ПРОБЛЕМЫ, МЕТОДЫ АНАЛИЗА, АЛГОРИТМЫ

*МАГЕРРАМОВ З.Т., АБДУЛЛАЕВ В.Г.,
МАГЕРРАМОВА А.З.*

Проводится обзор развития, характеристики и применения технологий Big Data, показывается взаимосвязь технологии Big Data с промышленными предприятиями на примере металлургического производства. Описываются принципы обработки Big Data на основе модели Apache Hadoop и компании Oracle, а также методы анализа массивов данных. Некоторые методы анализа данных сопровождаются алгоритмами кластеризации и в них применяется функция конкурентного сходства (FRiS-функция).

Ключевые слова: большие массивы данных, металлургическое производство, обработка и методы анализа, многомерный анализ, регрессия, классификация, кластеризация, алгоритмы кластеризации, FRiS-функция.

1. Введение

Термин Big Data относится к наборам данных, размер которых превосходит возможности типичных баз данных по хранению, управлению и анализу информации. Сами по себе алгоритмы Big Data возникли при внедрении первых высокопроизводительных серверов (мэйнфреймов), обладающих достаточными ресурсами для оперативной обработки информации и пригодных для компьютерных вычислений и дальнейшего анализа.

Предпринимателей и ученых волнуют вопросы качественной интерпретации данных, разработки инструментов для работы с ними и развития технологий хранения. Этому способствует внедрение и активное использование облачных моделей хранения и вычислений.

За последние три года человечество произвело информации больше, чем за всю историю своего существования до 2008 года. И рост продолжается экспоненциально. В настоящее время множество компаний следят за развитием технологий Big Data. В отчете, представленном компанией IDC [1] в декабре 2012 г., предсказывалось, что объемы информации будут удваиваться каждые 2 года в течение следующих 8 лет. За ближайшие 7 лет количество данных в мире достигнет 40 зеттабайт ($1 \text{ ЗБ} = 10^{21}$ байт), а это значит, что на каждого жителя Земли будет приходиться по 5200 ГБ данных.

В настоящее время большинство крупнейших поставщиков информационных технологий для организаций в своих деловых стратегиях используют понятие «большие данные», а основные ана-

литики рынка информационных технологий посвящают концепции выделенные исследования. Если обратить внимание на динамику роста данных, то обнаружим рост вычислительных средств, приложений и пользователей – от миллионов в эпоху мэйнфреймов до сотен миллионов в эпоху ПК и миллиардов пользователей в эпоху мобильных устройств, мобильного Интернета, социальных сетей, «облачных» технологий и построения всевозможных решений «умной» экономики. Отдельная тема – Интернет и мобильные технологии. Поток информации генерируется всё новыми интернет-сервисами, социальными сетями, приложениями электронной торговли, приложениями о местонахождении абонентов сетей. Количество e-mail, отправляемых каждую секунду в мире – 2,9 млн, объем видео, зачисляемого на YouTube каждую минуту – 20 часов. Объем данных, обрабатываемых Google за день – 24 петабайт. Количество сообщений на Твиттере в день – 50 млн. Количество минут, проведенных в Facebook в месяц – около 700 млрд. Объем данных, переданных/полученных на мобильные устройства – 1,3 экзбайт. Количество продуктов, заказываемых в Amazon в секунду – 72,9 [2].

2. Постановка задачи

Методику и инструменты работы со структурированными данными ИТ-индустрия создала давно – это реляционная модель данных и системы управления БД. Но современной тенденцией является потребность обработки большого объема неструктурированных данных, и это та область, где прежние подходы работают плохо или вообще не работают. Именно эта потребность требует новой методики обращения с данными, и сейчас все более популярной становится модель работы с Big Data. Занимаясь проблемами Big Data, перед разработчиками и учеными стоит задача – найти программное и техническое решение, способное легко интегрироваться в существующую инфраструктуру ЦОД и обеспечить все три этапа обработки информации: сбор, ее организацию и анализ.

3. Этапы и методы решения

3.1. Характеристики технологии Big Data. В качестве характеристик для больших данных Forrester определяет понятие Big Data как технологию в области аппаратного и программного обеспечения, которая объединяет, организует, управляет и анализирует данные, характеризующиеся «четырьмя V»: объемом (Volume), разнообразием (Variety), изменчивостью (Variability) и скоростью (Velocity) [2].

Эти характеристики являются существенными проблемами технологии Big Data. Рассмотрим каждую из этих составляющих.



Объем накопленных данных в корпорациях из разных сфер деятельности (источник: McKinsey)

Объем данных (Volume). Во введении мы описали лавинообразный рост объема данных в научных и персональных приложениях. Дополним картину информацией о том, какие объемы данных накопили корпорации. Только в США это более 100 Тбайт данных. При этом в разных вертикальных индустриях объем данных существенно различается, следовательно, актуальность применения технологии Big Data в них различна (рисунок) [2].

Разнообразие форматов данных (Variety). Способность приложения обрабатывать большие массивы данных, поступающих из разных источников в различных форматах, является главным критерием отнесения его к технологии Big Data. Многие бизнес-задачи и научные эксперименты требуют совместной обработки данных различных форматов – это могут быть табличные данные в СУБД, иерархические данные, текстовые документы, видео, изображения, аудиофайлы. Пример подобного рода задачи из области медицины: как найти оптимальный курс лечения для конкретного пациента, базируясь на огромном количестве историй болезней пациентов (которые постоянно меняются), а также на базе данных медицинских исследований и геномных данных? Другой пример – из области оптимизации бизнес-процессов: как провести анализ структурированных данных из ERP (*Enterprise Resource Planning*) – приложения, а также слабоструктурированных данных в виде логфайлов и неструктурированного текста из отзывов покупателей? Третий пример – из сферы прогнозирования погоды: как выполнить анализ климата на базе многолетних метеорологических данных и данных, поступающих со спутника в реальном времени?

Скорость поступления и обработки информации (velocity). В области Big Data выделяют ещё одну проблему: недостаточно высокая скорость обработки данных. В ряде задач эта скорость должна быть очень высокой. Например, биржевым игрокам иногда нужно мгновенно принять решение, основываясь на большом количестве данных о состоянии рынка — за пару секунд ситуация уже может измениться. Очень большая скорость поступления данных характерна для многих научных задач. Например, только один экспериментальный синхротрон Advanced Photon Source (APS) в Аргоннской лаборатории, используемый в числе прочего для томографической съемки объектов на субмикронных разрешениях, может ежедневно генерировать 150 Тбайт информации.

Ценность для бизнеса (Value). Компания IDC тоже выделяет «четыре V», характеризующие данные, однако параметр Variety (изменчивость), который применяет компания Forrester, она заменяет на параметр Value (ценность). IDC подчеркивает, что параметр Value — один из основных, позволяющих выделить Big Data как новое явление. Он относится к экономическому эффекту, который технология Big Data обеспечивает пользователям. Информация – это главный аспект успешного прогнозирования роста и составления маркетинговой стратегии в умелых руках маркетолога. Big Data является точнейшим инструментом маркетолога для предсказания будущего компании.

Применение технологии Big Data может быть полезно для решения следующих задач:

- лучше узнавать своих потребителей, привлекать аналогичную аудиторию в Интернете;
- оценивать уровень удовлетворенности клиентов;
- находить и внедрять новые способы, увеличивающие доверие клиентов;
- создавать проекты, пользующиеся спросом по прогнозированию рыночной ситуации;
- маркетинг и оптимизация продаж;
- эффективно сегментировать клиентов;
- совершенствовать качество товаров и услуг;
- принимать более обоснованные управленческие решения на основе анализа Big Data;
- оптимизировать портфель инвестиций;
- повышать производительности труда.

3.2. Big Data и промышленные предприятия.

Сфера использования технологий Big Data обширна. Известно, что основной поток информации генерируют не люди. Источником служат роботы, находящиеся в постоянном взаимодействии друг с другом. Это приборы для мониторинга,

сенсоры, системы наблюдения, операционные системы персональных устройств, смартфоны, интеллектуальные системы, датчики и прочее. Все они задают бешеный темп роста объема данных, что приводит к появлению потребности наращивать количество рабочих серверов (и реальных, и виртуальных) – как следствие, расширять и внедрять новые data-центры. На промышленных предприятиях объем информации неуклонно растет за счет данных, полученных с датчиков, измерительных и «умных» устройств. Самыми перспективными устройствами считаются датчики, которые могут передавать данные в режиме реального времени. Все устройства на предприятии с помощью таких датчиков могут быть объединены в сеть, а технологии Big Data позволят обрабатывать информацию, поступающую с них, и проводить необходимые мероприятия в автоматическом режиме. Например, предприятия могут с помощью датчиков получать ежеминутные данные о состоянии своего оборудования и на основе этих данных предсказывать оптимальное время для замены и обслуживания. Слишком ранняя замена приведет к дополнительным расходам, а поздняя – к потере прибыли вследствие простоя оборудования.

Рассмотрим взаимосвязь технологии Big Data с промышленными предприятиями на примере металлургического производства. В настоящее время металлургические компании стремятся к сокращению издержек, что позволит им быть конкурентоспособными на металлургическом рынке. Использование технологии Big Data может дать значительный экономический эффект в виде сокращения затрат на обработку информации в производстве, логистике и управлении.

Современное металлургическое производство представляет собой сложный комплекс различных переделов, базирующийся на месторождениях руд, коксующихся углей, энергетических мощностях.

Металлургическое производство включает в себя следующие комбинаты, заводы, цеха:

- шахты и карьеры по добыче руд и каменных углей;
- горно-обогатительные комбинаты для подготовки руды к плавке;
- коксохимические заводы или цеха для подготовки углей к коксованию и извлечению полезных химических продуктов;
- энергетические цеха для получения сжатого воздуха, кислорода, а также очистки газов металлургических производств;
- доменные цеха для выплавки чугуна и ферросплавов;
- заводы для производства ферросплавов;

– сталеплавильные цеха (конвертерные, мартеновские, электросталеплавильные) для производства стали;

– прокатные цеха для получения сортового проката (листы, балки, рельсы, прутки, проволока) [4].

Автоматизированные системы управления металлургическими комбинатами ежесекундно порождают данные о процессах:

- технологических (АСУ ТП);
- логистических (АСУ транспортной логистики);
- управления (MES – Manufacturing execution system и ERP – Enterprise Resource Planning системы).

Системы АСУ ТП собирают данные с датчиков агрегатов о состоянии и режимах технологических процессов. С систем контроля качества могут поступать видеоизображения полос прокатки и дефектов на полосе, карты ультразвукового контроля. АСУ транспортной логистики содержат данные о перемещении материалов. ERP и MES владеют информацией о заказах, планировании, оперативном управлении обработкой материалов, о состоянии запасов на складах.

Например, только цепочка производства от выплавки металла до выпуска автолиста может включать в себя от 7000 до 15000 источников разнородных неструктурированных данных, поступающих в реальном масштабе времени. Высокая степень автоматизации производства порождает у персонала предприятий «иллюзию доступности данных» [5].

Оснащение производства современными системами автоматизации приводит к оцифровке всех получаемых данных, и это создает у персонала предприятия иллюзию их доступности. Но «оцифровано» – не значит «доступно».

Данные о технологических процессах есть в АСУ ТП агрегатов, данные о производстве – в MES системах, данные о заказах – в ERP.

Перечисленные выше характеристики Big Data («четыре V») хорошо подходят для структурированных и неструктурированных данных металлургического производства:

- множество сигналов с датчиков контроля технологических процессов,
- карты ультразвукового контроля,
- изображения полос прокатки, содержащих дефекты на полосе,
- данные о перемещении продукции и материалов,
- данные о заказах и поставщиках.

Технология Big Data позволит свести данные из АСУ ТП, АСУ транспортной логистики и систем класса ERP и MES воедино, тратя на это в разы меньшее количество времени, по сравнению с

традиционным подходом. Экономия времени, соответственно, принесет экономический эффект, делая системы обработки Big Data выгодным делом для металлургического бизнеса в целом.

В перспективе металлургические компании, благодаря Big Data, получают возможность заниматься предиктивным анализом, к примеру, с большой вероятностью предсказывать долю брака в металлургическом переделе на основе технологии машинного обучения.

3.3. *Обработка и методы анализа Big Data.* С точки зрения обработки в основу технологий Big Data положены два основных принципа:

- 1) распределенного хранения данных;
- 2) распределенной обработки, с учетом локальности данных.

Распределенное хранение решает проблему большого объема данных, позволяя организовывать хранилище из произвольного числа отдельных простых носителей. Хранение может быть организовано с разной степенью избыточности, обеспечивая устойчивость к сбоям отдельных носителей.

Распределенная обработка с учетом локальности данных означает, что программа обработки доставляется на вычислитель, находящийся как можно ближе к обрабатываемым данным. Это принципиально отличается от традиционного подхода, когда вычислительные мощности и подсистема хранения разделены и данные должны быть доставлены на вычислитель. Таким образом, технологии Big Data опираются на вычислительные кластеры из множества вычислителей, снабженных локальной подсистемой хранения.

Доступ к данным и их обработка осуществляются специальным программным обеспечением. Наиболее известным и интенсивно развиваемым проектом в области Big Data является Apache Hadoop [6,7]. В настоящее время на рынке информационных систем и программного обеспечения синонимом Big Data является технология Hadoop, которая представляет собой программный фреймворк, позволяющий хранить и обрабатывать данные с помощью компьютерных кластеров, используя парадигму MapReduce.

Основными составляющими платформы Hadoop являются:

- отказоустойчивая распределенная файловая система Hadoop Distributed File System (HDFS), при помощи которой осуществляется хранение;
- программный интерфейс Map Reduce, который является основой для написания приложений, обрабатывающих большие объемы структурированных и неструктурированных данных параллельно на кластере, состоящем из тысяч машин;
- Apache Hadoop YARN, выполняющий функцию управления данными.

В соответствии с подходом MapReduce обработка данных состоит из двух шагов: Map и Reduce. На шаге Map выполняется предварительная обработка данных, которая осуществляется параллельно на различных узлах кластера. На шаге Reduce происходит сведение предварительно обработанных данных в единый результат.

В основе модели работы Apache Hadoop лежат три основных принципа. Во-первых, данные равномерно распределяются на внутренних дисках множества серверов, объединенных HDFS. Во-вторых, не данные передаются программе обработки, а программа – к данным. Третий принцип – данные обрабатываются параллельно, причем эта возможность заложена архитектурно в программном интерфейсе Map Reduce. Таким образом, вместо привычной концепции «база данных+сервер» у нас имеется кластер из множества недорогих узлов, каждый из которых является и хранилищем, и обработчиком данных, а само понятие «база данных» отсутствует. Платформа Hadoop позволяет сократить время на обработку и подготовку данных, расширяет возможности по анализу, позволяет оперировать новой информацией и неструктурированными данными.

Компания Oracle разбивает жизненный цикл обработки информации на три этапа и использует для каждого из них собственное решение:

- 1) Сбор, обработка и структурирование данных. В качестве решения применяется Oracle Big Data Appliance – это предустановленный Hadoop-кластер, Oracle NoSQL Database и средства интеграции с другими хранилищами данных. Задача Oracle Big Data Appliance состоит в хранении и первичной обработке неструктурированной или частично структурированной информации, т.е. как раз в том, что у систем на базе Hadoop получается лучше всего.

- 2) Агрегация и анализ данных. Для работы со структурированными данными используется комплекс Oracle Exadata. Модули интеграции Oracle Big Data Appliance позволяют оперативно загружать данные в Oracle Exadata, а также получать доступ к данным «на лету» из Oracle Exadata.

- 3) Аналитика данных в реальном времени. Для максимально оперативного анализа полученных данных используется Oracle Exalytics Database Machine, которая позволяет решать аналитические задачи фактически в режиме «online».

Существует множество разнообразных методов анализа массивов данных, в основе которых лежит примерно одинаковый набор инструментов анализа данных [3]: многомерный анализ (OLAP), регрессия, классификация, кластеризация и поиск закономерностей. Некоторые из перечисленных методик вовсе не обязательно применимы исключительно к большим данным и могут с успехом

использоваться для меньших по объему массивов (например, А/В-тестирование, регрессионный анализ).

Многомерный анализ — суть метода заключается в построении многомерного куба и получении его различных срезов. Результатом анализа, как правило, является таблица, в ячейках которой содержатся агрегированные показатели (количество, среднее, минимальное или максимальное значение и так далее). В зависимости от реализации, существуют три типа системы многомерного анализа (OLAP): многомерная OLAP (Multidimensional OLAP – MOLAP); реляционная OLAP (Relational OLAP – ROLAP); гибридная OLAP (Hybrid OLAP – HOLAP).

Среди них ROLAP-системы являются наиболее прозрачными и изученными, поскольку основываются на широко распространенных реляционных СУБД, в то время как внутреннее устройство MOLAP и HOLAP обычно более закрыто и относится к области «ноу-хау» конкретных коммерческих продуктов.

MOLAP представляет информацию в виде «честной» многомерной модели, но внутри используются те же подходы, что и в ROLAP: схемы «звезда» и «снежинка». С точки зрения СУБД база данных ROLAP — это обыкновенная реляционная база, и для нее необходимо поддерживать весь перечень операций. Однако это не позволяет, во-первых, жестко контролировать этапы ввода данных. Во-вторых, собирать статистику и подбирать оптимальные структуры для хранения индексов. В-третьих, оптимизировать размещение данных на диске для обеспечения высокой скорости ввода/вывода. В-четвертых, при выполнении аналитических запросов из-за высоких требований к быстродействию нет возможности произвести глубокий статистический анализ и выработать оптимальный план выполнения. В ROLAP используются «родные» реляционные оптимизаторы запроса, которые никак не учитывают «многомерность» базы данных. Технологии MOLAP лишены перечисленных недостатков и благодаря этому позволяют добиться большей скорости анализа.

Выбор технологии MOLAP/ROLAP/HOLAP при анализе Big Data зависит от частоты обновления базы данных. С точки зрения распараллеливания обработки, на первый взгляд, все просто — любой многомерный куб может быть «разрезан» по делениям одного из измерений и распределен между несколькими серверами. Например, можно разделить куб на временные периоды (по годам и месяцам), по территориальному признаку (каждый сервер отвечает за свой регион) и так далее. Критерием для деления куба является следующий принцип: выполнение многомерного запроса

должно ложиться не на один сервер, а на несколько, после чего полученные результаты собираются в единое целое. Например, если пользователь запрашивает статистику продаж по стране за указанный промежуток времени, а данные распределены по нескольким региональным OLAP-серверам, то каждый сервер возвращает свой собственный ответ, которые затем собираются воедино. Если же данные будут распределены по временному критерию, то при выполнении рассматриваемого примера запроса вся нагрузка ляжет на один сервер.

Проблема в том, что, во-первых, очень трудно заранее определить оптимальное распределение данных по серверам, а во-вторых, для части аналитических запросов может быть заранее неизвестно, какие данные и с каких серверов понадобятся.

Применительно к Большим Данным это означает, что существующие подходы для многомерного анализа могут хорошо масштабироваться и что они допускают распределенный сбор информации — каждый сервер может самостоятельно собирать информацию, осуществлять ее очистку и загрузку в локальную базу.

Регрессия — под регрессией понимают построение параметрической функции, описывающей изменение указанной числовой величины в указанный промежуток времени. Эта функция строится на основе известных данных, а затем используется для предсказания дальнейших значений этой же величины. На вход метода поступает последовательность пар вида «время — значение», описывающая поведение этой величины при заданных условиях, например, количество продаж конкретного вида товара в конкретном регионе. На выходе — параметры функции, описывающей поведение исследуемой величины.

Независимо от вида используемой параметрической функции подбор значений ее параметров осуществляется одним и тем же способом. Вычисляется суммарная разница между наблюдаемыми значениями и значениями, которые дает функция при текущих значениях ее параметров. Затем определяется, как следует подкорректировать значения параметров для того, чтобы уменьшить текущую суммарную разницу. Эти операции повторяются до тех пор, пока суммарная разница не достигнет необходимого минимума или ее дальнейшее уменьшение станет невозможным.

С точки зрения обработки данных при регрессионном анализе ключевыми операциями являются вычисление текущей суммарной разницы и корректировка значений параметров. Если первая операция распараллеливается очевидным образом (сумма вычисляется по частям на отдельных серверах, а затем суммируется на центральном

сервере), то со второй сложнее. В наиболее общем случае при корректировке весов используют общеизвестный математический факт: функция нескольких параметров возрастает в направлении градиента и убывает в направлении, обратном градиенту. В свою очередь, вычисление градиента состоит в вычислении частных производных функции по каждому из параметров, что сводится к дискретному дифференцированию, основанному на вычислении взвешенных сумм. В результате корректировка значений параметров также сводится к суммированию, которое может быть распараллелено.

Если регрессионный анализ сводится к вычислению взвешенных сумм, то он обладает примерно той же степенью применимости и при работе с Big Data, что и многомерный анализ. Таким образом, системы регрессионного анализа вполне могут масштабироваться и работать в условиях распределенного сбора информации.

Классификация – ее задача отчасти похожа на задачу регрессии и заключается в попытке построения и использования зависимости одной переменной от нескольких других. Например, имея базу данных о цене объектов недвижимости, можно построить систему правил, позволяющую на основе параметров нового объекта предсказать его примерную цену. Отличие классификации от регрессии состоит в том, что анализируется не временной ряд – подаваемые на вход значения никак не могут быть упорядочены.

На текущий момент разработано множество методов классификации (функции Байеса, нейронные сети, машины поддерживающих векторов, деревья решений и т. д.), каждый из которых имеет под собой хорошо проработанную научную теорию. Вместе с тем все методы классификации строятся по одной и той же схеме. Сначала производится обучение алгоритма на сравнительно небольшой выборке, а затем – применение полученных правил к остальной выборке. На первом этапе возможно копирование массива данных на один сервер для запуска «классического» алгоритма обучения без распараллеливания работы. Однако на втором этапе данные могут обрабатываться независимо — система правил, полученная по итогам самообучения, копируется на каждый сервер, и через нее прогоняется весь массив данных, хранящийся на этом сервере. Полученные результаты могут либо сохраняться там же на сервере, либо отправляться для дальнейшей обработки.

Таким образом, на этапе обучения классификаторов о работе с Big Data пока речи не идет – не существует выборки такого объема, подготовленных для обучения систем, а на этапе классификации отдельные порции данных обрабатываются

независимо друг от друга. *Кластеризация* – ее задача состоит в разбиении множества информационных сущностей на группы, при этом члены одной группы более похожи друг на друга, чем члены из разных (классификация относит каждый объект к одной из заранее определенных групп). В качестве критерия схожести используется функция-расстояние, на вход которой поступают две сущности, а на выход – степень их схожести. Известно множество различных способов кластеризации (графовые, иерархические, итеративные, сети Кохонена).

Проблема кластеризации Big Data состоит в том, что имеющиеся алгоритмы предполагают возможность непосредственного обращения к любой информационной сущности в исходных данных (заранее невозможно предугадать, какие именно сущности понадобятся алгоритму). В свою очередь, исходные данные могут быть распределены по разным серверам, и при этом не гарантируется, что каждый кластер хранится строго на одном сервере. Если распределение данных по серверам делать прозрачным для алгоритма кластеризации, то это неизбежно приведет к копированию больших объемов с одного сервера на другой.

Решение проблемы может быть следующим. На каждом сервере запускается свой алгоритм, который оперирует только данными этого сервера, а на выходе дает параметры найденных кластеров и их веса, оцениваемые исходя из количества элементов внутри кластера. Затем полученная информация собирается на центральном сервере и производится метакластеризация – выделение групп близко расположенных кластеров с учетом их весов. Этот метод универсален, хорошо распараллеливается и может использовать любые другие алгоритмы кластеризации, однако он требует проведения серьезных научных исследований, тестирования на реальных данных и сравнения полученных результатов с другими «локальными» методами.

Таким образом, для анализа Big Data подавляющая часть методов кластеризации неприменима в чистом виде и необходимы дополнительные исследования.

Поиск закономерностей – суть метода заключается в нахождении правил, описывающих взаимозависимости между внутренними элементами данных. Классическим примером является анализ покупок в супермаркете и выявление правил вида «если человек покупает фотоаппарат, то обычно он покупает еще к нему аккумулятор и карту памяти». На вход задачи поиска закономерностей поступает неупорядоченное множество сущностей, для каждой из которых известен набор присутствующих информационных признаков; например, такими сущностями могут быть чеки

на покупки, а признаками – купленные товары. Задача поиска закономерностей сводится к выявлению правил вида «если присутствуют признаки A_1, A_2, \dots, A_n , то присутствуют и признаки B_1, B_2, \dots, B_m », при этом каждое правило характеризуется двумя параметрами: вероятностью срабатывания и поддержкой. Первый параметр показывает, как часто выполняется данное правило, а второй – как часто применимо данное правило, т.е. как часто встречается сочетание признаков A_1, A_2, \dots, A_n .

A/B тестирование — методика, в которой контрольная выборка поочередно сравнивается с другими. Тем самым удается выявить оптимальную комбинацию показателей для достижения, например, наилучшей ответной реакции потребителей на маркетинговое предложение. Большие данные позволяют провести огромное количество итераций и таким образом получить статистически достоверный результат.

Краудсорсинг – методика сбора данных из большого количества источников: категоризация и обогащение данных силами широкого, неопределённого круга лиц.

Смешение и интеграция данных – набор техник, позволяющих интегрировать разнородные данные из разнообразных источников для возможности глубинного анализа.

Машинное обучение («искусственный интеллект») – преследует цель создания алгоритмов самообучения на базе статистического анализа данных или машинного обучения для получения комплексных прогнозов.

Генетические алгоритмы – в этой методике возможные решения представляют в виде «хромосом», которые могут комбинироваться и мутировать. Как и в процессе естественной эволюции, выживает наиболее приспособленная особь.

Оптимизация – набор численных методов для редизайна сложных систем и процессов для улучшения одного или нескольких показателей. Помогает в принятии стратегических решений, например, состава выводимой на рынок продуктовой линейки, проведении инвестиционного анализа.

Визуализация аналитических данных – методы для представления информации в виде рисунков, графиков, схем и диаграмм с использованием интерактивных возможностей и анимации как для результатов, так и для использования в качестве исходных данных.

3.4. Алгоритмы кластеризации Big Data. При выполнении кластеризации основной проблемой является определение числа кластеров. Число методов разбиения множества на кластеры довольно велико. Все их можно подразделить на иерархические и неиерархические.

В неиерархических алгоритмах характер их работы и условие останковки необходимо заранее регламентировать часто довольно большим числом параметров, что иногда затруднительно, особенно на начальном этапе изучения материала. Но в таких алгоритмах достигается большая гибкость в варьировании кластеризации и обычно определяется число кластеров. С другой стороны, когда объекты характеризуются большим числом признаков (параметров), то приобретает большое значение задача группировки признаков.

В иерархических алгоритмах фактически отказываются от определения числа кластеров, строя полное дерево вложенных кластеров (дендрограмму). Их число определяется из предположений, не относящихся к работе алгоритмов, например, по динамике изменения порога расщепления (слияния) кластеров. Трудности таких алгоритмов хорошо изучены: выбор мер близости кластеров, проблема инверсий индексации в дендрограммах, негибкость иерархических классификаций, которая иногда весьма нежелательна. Тем не менее, представление кластеризации в виде дендрограммы позволяет получить наиболее полное представление о структуре кластеров.

3.4.1. Алгоритм *k*-средних (*k-means*).

Алгоритм *k-means* строит *k* кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм *k*-средних, – наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Общая идея алгоритма: заданное фиксированное число *k* кластеров наблюдения сопоставляется кластерам так, что средние в кластере максимально возможно отличаются друг от друга. Описание алгоритма [8]:

Этап 1. Первоначальное распределение объектов по кластерам. Выбирается число *k*, и на первом шаге эти точки считаются "центрами" кластеров. Каждому кластеру соответствует один центр. Выбор начальных центроидов может осуществляться следующим образом: выбор *k*-наблюдений для максимизации начального расстояния; случайный выбор *k*-наблюдений; выбор первых *k*-наблюдений. В результате каждый объект назначен определенному кластеру.

Этап 2. Вычисляются центры кластеров, которыми далее считаются по координатным средние кластеров. Объекты опять перераспределяются. Процесс вычисления центров и перераспределения объектов продолжается до тех пор, пока не выполнено одно из условий: кластерные центры стабилизировались, т.е. все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации; число итераций равно максимуму.

Выбор числа кластеров является сложным вопросом. Если нет предположений относительно этого числа, рекомендуют создать 2 кластера, затем 3, 4, 5 и более, сравнивая полученные результаты.

Достоинства метода: простота использования; быстрота использования; понятность и прозрачность алгоритма.

Недостатки метода: алгоритм слишком чувствителен к выбросам, которые могут исказить среднее; медленная работа на больших базах данных; необходимо задавать количество кластеров.

3.4.2. Алгоритм HCM (Hard C – Means).

Метод *Hard C – Means* применяется, в основном, для кластеризации больших наборов числовых данных. Описание алгоритма [9]:

Шаг 1. Инициализация кластерных центров c_i ($i = 1, 2, \dots, c$). Это можно сделать, выбрав случайным образом c – векторов из входного набора.

Шаг 2. Вычисление рядовой матрицы M . Она состоит из элементов m_{ik} :

$$m_{ik} = \begin{cases} 1, & \|u_k - c_i\|^2 \leq \|u_k - c_j\|^2, \text{ для всех } i \neq j, \\ & i = 1, 2, \dots, c, k = 1, 2, \dots, K, \\ 0, & \text{остальное} \end{cases}$$

где K – количество элементов во входном наборе данных. Матрица M обладает следующими свойствами:

$$\sum_{i=1}^c m_{ij} = 1, \quad \sum_{j=1}^K m_{ij} = K.$$

Шаг 3. Расчет объектной функции:

$$J = \sum_{i=1}^c J_i = \sum_{i=1}^c \left(\sum_{k, u_k \in C_i} \|u_k - c_i\|^2 \right).$$

Шаг 4. Пересчет кластерных центров уравнением:

$$c_i = \frac{1}{|C_i|} \sum_{k, u_k \in C_i} u_k,$$

где $|C_i|$ – количество элементов в i -м кластере.

Шаг 5. Переход на шаг 2.

Достоинствами метода являются: легкость реализации, вычислительная простота, а недостатками – задание количества кластеров, отсутствие гарантии в нахождении оптимального решения.

3.4.3. Алгоритм нечеткой кластеризации Fuzzy C-means.

Метод нечеткой кластеризации *Fuzzy C-means* так же применяется, в основном, для кластеризации больших наборов числовых данных.

Описание алгоритма [10]:

Пусть нечеткие кластеры задаются матрицей разбиения:

$$F = [\mu_{ki}], \quad \mu_{ki} \in [0, 1], \quad k = \overline{1, M}; \quad i = \overline{1, c},$$

где μ_{ki} – степень принадлежности объекта k к кластеру i ; c – количество кластеров; M – количество элементов. При этом:

$$\left. \begin{aligned} \sum_{i=1}^c \mu_{ki} &= 1, \quad k = \overline{1, M}; \\ 0 < \sum_{k=1}^M \mu_{ki} &< M, \quad i = \overline{1, c}. \end{aligned} \right\} \quad (1)$$

Этап 1. Установить параметры алгоритма: c – количество кластеров; m – экспоненциальный вес, определяющий нечеткость, размазанность кластеров ($m \in [1, \infty)$); ε – параметр останова алгоритма.

Этап 2. Генерация случайным образом матрицы нечеткого разбиения с учетом условий (1).

Этап 3. Расчет центров кластеров:

$$V_i = \frac{\sum_{k=1}^M \mu_{ki}^m |X_k|}{\sum_{k=1}^M \mu_{ki}^m}, \quad i = \overline{1, c}.$$

Этап 4. Расчет расстояния между объектами X и центрами кластеров:

$$D_{ki} = \sqrt{\|X_k - V_i\|^2}, \quad k = \overline{1, M}; \quad i = \overline{1, c}.$$

Этап 5. Пересчет элементов матрицы разбиения с учетом следующих условий:

$$\text{если } D_k > 0: \mu_k = \frac{1}{\left(D_{jk}^2 * \sum_{j=1}^c \frac{1}{D_{jk}^2} \right)^{1/(m-1)}}, \quad j = \overline{1, c},$$

$$\text{если } D_k = 0: \mu_k = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}, \quad j = \overline{1, c}.$$

Этап 6. Проверить условие $\|F - F^*\| < \varepsilon$, где F^* – матрица нечеткого разбиения на предыдущей итерации алгоритма. Если «Да», то переход к этапу 7, иначе к этапу 3.

Этап 7. Конец.

Достоинства: нечеткость при определении объекта в кластер позволяет определять объекты, которые находятся на границе, в кластеры. Недостатки: вычислительная сложность, задание количества кластеров, неопределенность с объектами, которые удалены от центров всех кластеров.

3.4.4. Алгоритм FRiS-Cluster.

Одной из основных проблем, возникающих при решении задачи классификации каких-либо объектов, является проблема выбора меры схожести. Чаще всего в этой роли выступает расстояние (евклидово, Чебышева, «городских кварталов»).

В основе алгоритма лежит FRiS-функция (Function of Rival Similarity – функция конкурентного сходства). FRiS – мера схожести двух объектов относительно некоторого третьего объекта. Она, в отличие от других существующих мер схожести, позволяет не просто оценивать понятия «далеко» или «близко», «похож» или «не похож», но и давать количественную оценку схожести. Такой подход позволяет учитывать большее число факторов при классификации. Исследования показали, что FRiS-функция хорошо имитирует человеческий механизм восприятия сходства и различия. Это позволяет использовать ее как базовый элемент для различных типов задач, включая задачи кластеризации документов.

Подробное описание алгоритма FRiS-Cluster можно найти в [11]. Алгоритм кластеризации можно описать следующим образом. Он последовательно ищет решение задачи кластеризации для значений $k = 1, \dots, K$, где K – заданное пользователем максимальное число кластеров. Затем из полученных решений выбирается лучшее. В описании алгоритма используются FRiS-функция:

$$F(a, S) = \frac{r_2(a, S) - r_1(a, S)}{r_2(a, S) + r_1(a, S)}$$

и редуцированная FRiS-функция:

$$F^*(a, S) = \frac{r_2^* - r_1(a, S)}{r_2^* + r_1(a, S)}.$$

Здесь: r_1 – минимальное расстояние до ближайшего столпа «своего» кластера, r_2 – минимальное расстояние до ближайшего столпа кластера-конкурента; r_2^* – расстояние до виртуального столпа (используется при $k=1$, когда настоящие конкуренты ещё не были найдены).

Шаг 1. При $k=1$ для каждого объекта a обучающей выборки находится значение средней редуцированной FRiS-функции $\bar{F}^*\{a\}$:

$$\bar{F}^*(S) = \frac{1}{M} \sum_{a \in A} F^*(a, S).$$

Шаг 2. Объект с максимальным значением средней редуцированной FRiS-функции принимается за первый столп – s_1 .

Шаг 3. Каждый объект a , за исключением s_1 , пробуете на роль второго столпа, столпом s_1 назначается объект, для которого значение функции $\bar{F}^*\{a, s_1\}$ оказывается максимальным.

Шаг 4. Все объекты исходной выборки распределяются между столпами s_1, s_2 , образуя кластеры A_1, A_2 , соответственно. Объект a относится к кластеру A_1 , если расстояние от a до s_1 меньше, чем для любого s_i .

Шаг 5. После добавления кластера A_2 , эталонный объект для кластера A_1 выбирается снова, но уже среди объектов кластера A_1 . Столпом s_{12} назначается объект, на котором достигает своего максимума функция $\bar{F}^*\{a, s_2\}$, описанная ниже. Аналогично, находится столп s_{22} для второго кластера. Полученная кластеризация представляет собой решение задачи для $k=2$:

$$\bar{F}(S) = \frac{1}{M} \sum_{a \in A} F(a, S).$$

Шаг 6. Далее для каждого нового столпа повторяются те же операции, что и при добавлении объекта два. Шаг 5 выполняется для всех кластеров текущей кластеризации.

Данный алгоритм имеет высокую трудоёмкость – $O(n^3)$, а также алгоритму требуется более ячеек n^2 памяти, что обусловлено необходимостью хранения матрицы парных расстояний.

3.4.5. Алгоритм FRiS-Stolp.

Другим применением FRiS-функции является один из алгоритмов отбора эталонных образцов

(столпов) для метрического классификатора, именуемый FRiS-Stolp. Выбор эталонов делается с помощью алгоритма FRiS-Stolp. Его идея состоит в том, что все объекты первого образа по очереди назначаются эталонами. Он нацелен на выбор минимального числа столпов, которые защищают не только самих себя, но обеспечивают заданную надежность защиты всех остальных объектов обучающей выборки. Первыми выбираются столпы, защищающие максимально возможное количество объектов с заданной надежностью. По этой причине при нормальных распределениях в первую очередь будут выбраны столпы, расположенные в точках математического ожидания [12].

Алгоритм FRiS-Stolp:

Шаг 1. Проверим вариант, при котором первый случайно выбранный объект a_i является единственным столпом образа S_1 , а все другие образы в качестве столпов имеют свои объекты. Для всех объектов $a_j \neq a_i$ первого образа находим расстояние r_{1j} до своего столпа a_i и расстояние r_{2j} до ближайшего объекта чужого образа. По этим расстояниям вычисляется значение FRiS-функции для каждого объекта a_j первого образа. Находим те m_i объектов первого образа, значение функций принадлежности F которых выше заданного порога F^* , например, $F^*=0$. По этим m_i объектам вычисляем суммарное значение FRiS-функции F_i , которое характеризует пригодность объекта a_i на роль столпа.

Шаг 2. Аналогичную процедуру повторяем, назначая в качестве столпа все M объектов первого образа по очереди.

Шаг 3. Находим объект a_i с максимальным значением F_i и объявляем его первым столпом A_{11} первого кластера S_{11} первого образа S_1 .

Шаг 4. Исключаем из первого образа m_i объектов, входящих в первый кластер.

Шаг 5. Для остальных объектов первого образа находим следующий столп повторением шагов 1–4.

Шаг 6. Процесс останавливается, если все объекты первого образа оказались включенными в свои кластеры.

Шаг 7. Восстанавливаем все объекты образа S_1 и для образа S_2 выполняем шаги 1–6.

Шаг 8. Повторяем шаги 1–7 для всех остальных образов.

На этом шаге заканчивается первый этап поиска столпов. Каждый столп A_i защищает подмножество объектов m_i своего кластера S_i .

4. Выводы

Эпоха Big Data уже наступила – объемы данных, генерируемых в науке, бизнесе, промышленности и управлении ИТ, растут экспоненциально. Однако существующие приложения обработки Big Data

не позволяют контролировать этапы ввода данных, собирать статистику и подбирать оптимальные структуры для хранения индексов, оптимизировать размещение данных на диске для обеспечения высокой скорости ввода/вывода, для выполнения аналитических запросов нет возможности произвести глубокий статистический анализ и выработать оптимальный план выполнения.

Важнейшее значение для масштабируемости и быстродействия приложений имеют характеристики сетей ЦОД — оптимизация качества обслуживания (QoS) требует активного обмена информацией между вычислительными узлами. Однако большинство нынешних ЦОД не способны обеспечить высокие скорости переноса данных, сопоставимые с показателями коммуникационных сетей высокопроизводительных компьютеров.

Что касается методов анализа для обработки Big Data, существующие на сегодня инструменты и наиболее распространенные методы анализа массивов данных пока не полностью удовлетворяют требованиям приложений обработки Big Data. В одном случае они не пригодны для обработки больших данных, в другом — затрудняется их применимость при построении автоматической классификации множества объектов в условиях отсутствия априорной информации о числе классов, в третьем — алгоритм имеет высокую трудоёмкость.

В данный момент можно прогнозировать высокоскоростную доставку данных из распределенных источников, оптимизацию переноса данных можно осуществить, например, с помощью развивающихся сейчас методов управления ресурсами с соблюдением гарантий качества обслуживания (QoS). В промышленной сфере можно прогнозировать аппаратные средства со специализированными датчиками для точного снятия показателей данных, а также развитие приложений, которые будут эти данные собирать, обрабатывать и структурировать, передавать в центры обработки, визуализировать для легкого восприятия, что, в свою очередь, облегчит принятие правильного решения.

В [13] разработана параллельная реализация алгоритма FRiS для кластеризации научных статей на основе технологии параллельных вычислений Message Passing Interface (MPI). В качестве меры близости при кластеризации принята мера конкурентного сходства. Для настройки весовых коэффициентов при вычислении меры сходства используется генетический алгоритм.

Литература: 1. Gantz John, Reinsel David. [http://www.emc.com/collateral/analyst-reports/idc-the-](http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf)

[digital-universe-in-2020.pdf](http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf). 2. Найдич А. www.compress.ru. 3. Селезнев К. Проблемы анализа Больших Данных // Открытые системы. СУБД №07, 2012 с.25-30. 4. Бабич В.К. и др. Основы металлургического производства. М.: Металлургия, 1988. 272 с. 5. Федин М.В. Перспективы использования систем обработки больших данных (bigdata) в металлургической промышленности // Economics. 2015, № 8(9). С. 52-54. 6. Проект Apache Hadoop. <https://hadoop.apache.org/>. 7. Артемов С. Big Data: новые возможности для растущего бизнеса.

<http://www.pcweek.ru/upload/iblock/d05/jet-big-data.pdf>.

8. Daniel Fasulo «An Analysis of Recent Work on Clustering Algorithms». Электронное издание. 9. Паклин Н. Алгоритмы кластеризации на службе Data Mining.

<http://www.basegroup.ru/clusterization/datamining.htm>.

10. Jan Jantzen «Neurofuzzy Modelling». Электронное издание. 11. Борисова И.А., Загоруйко Н.Г. Труды Всероссийской Конференции «Знания-Онтология-Теория». Новосибирск, 2007. Том II. С. 67-76. 12. Борисова И.А. и др. Труды Всероссийской конференции «Знания – Онтология – Теория», Новосибирск, 2007. Том I. С.37–44. 13. Мансурова М.Е. и др. Parallel computational technologies (PCT) 2016. <http://ceur-ws.org/Vol-1576/128.pdf>.

Transliterated bibliography:

1. Gantz John, Reinsel David. <http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf>.

2. Andrey Naydich. www.compress.ru.

3. Konstantin Seleznev. Problemy analiza Bolshih Danyih // Otkryitye sistemyi. SUBD #07, 2012 s.25-30.

4. Babich V.K. i dr. Osnovy metallurgicheskogo proizvodstva. M: Metallurgiya, 1988. 272 s.

5. Fedin M.V. Perspektivy ispol'zovaniya sistem obrabotki bol'shih danyih (bigdata) v metallurgicheskoy promyshlennosti // Economics. 2015, № 8(9). С. 52-54. 6. Project Apache Hadoop. <https://hadoop.apache.org/>.

7. Artemov S. Big Data: novyye vozmozhnosti dlya rastuschego biznesa.

<http://www.pcweek.ru/upload/iblock/d05/jet-big-data.pdf>.

8. Daniel Fasulo «An Analysis of Recent Work on Clustering Algorithms». Elektronnoe izdanie.

9. Paklin N. Algoritmyi klasterizatsii na sluzhbe Data Mining.

<http://www.basegroup.ru/clusterization/datamining.htm>.

10. Jan Jantzen «Neurofuzzy Modelling». Elektronnoe izdanie.

11. Borisova I.A., Zagoruyko N.G. Trudyi Vserossiyskoy Konferentsii «Znaniya-Ontologii-Teorii», Novosibirsk, 2007, Tom II, s. 67-76.

12. Borisova I.A. i dr. Trudyi Vserossiyskoy konferentsii «Znaniya-Ontologiya-Teoriya», Novosibirsk, 2007. Tom I. S.37–44.

13. Mansurova M.E. i dr. Parallel computational technologies (PCT) 2016. <http://ceur-ws.org/Vol-1576/128.pdf>.

Поступила в редколлегию 12.04.2017

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Магеррамов Закир Тулуевич, канд. техн. наук, доцент кафедры «Прикладная информатика» Азербайджанского Технического Университета. Научные интересы: численные методы, моделирование и оптимальное управление, информационные технологии, объектно-ориентированное программирование. Увлечения: научные книги, художественная литература (классика), мир животных. Адрес: Азербайджан, AZ1114, Баку, ул. И. Джумшудова, 1/7, кв. 110, тел. (99412)5689951, (050)3212595, e-mail: zakirmaharramov@rambler.ru

Абдуллаев Вугар Гаджимамудович, канд. техн. наук, доцент кафедры «Компьютерная инженерия технологии и программирование» Азербайджанской Государственной Нефтяной Академии (АГНА), Институт Кибернетики НАНА. Научные интересы: информационные технологии, веб-программирования, мобильные приложения. Увлечение: электронная коммерция, B2B, B2C проекты, научные книги, спорт. Адрес: Азербайджан, AZ1129, Баку, ул. М. Гади, 53, кв. 81, тел. (99412)5712428, (050)3325483, e-mail: abdulvugar@mail.com

Магеррамова Айнур Закировна, инженер-экономист, магистрант Aix-Marseille School of Economics (Франция). Научные интересы: численные методы в экономике, экономика развивающихся стран. Увлечения: конкурсы kaggle.com, йога. Адрес: Франция, 13100, Экс-ан-Прованс, ул. Маршал Леклерк, Л'Эстелан, кв. 205. E-mail: aynur.maharramova@etu.univ-amu.fr

Magerramov Zakir Tuluyevich, Cand. tech. Sci., Associate Professor of Applied Informatics at the Azerbaijan Technical University. Scientific interests: numerical methods, modeling and optimal control, information technologies, object-oriented programming. Hobbies: scientific books, artistic literature (classics), animal world. Address: Azerbaijan, AZ1114, Baku, I. Jumshudova, 1/7, apt. 110, tel. (99412) 5689951, (050) 3212595, e-mail: zakirmaharramov@rambler.ru

Abdullaev Vugar Gadzhimakhmudovich, Cand. tech. Sci., Associate Professor of Computer Engineering and Technology Programming at the Azerbaijan State Oil Academy (ASAN), Institute of Cybernetics of ANAS. Scientific interests: information technology, web programming, mobile application. Hobbies. e-commerce, B2B, B2C projects, science books, sports. Address: Azerbaijan, AZ1129, Baku, M. Gadi, 53, apt. 81, tel. (99412) 5712428, (050) 3325483, e-mail: abdulvugar@mail.com

Magerramova Ainur Zakirovna, engineer-economist, master student of the Aix-Marseille School of Economics (France). Scientific interests: numerical methods in economics, economics of developing countries. Hobbies: competitions kaggle.com, yoga. Address: France, 13100, Aix-en-Provence, Marshal Lek-lerk, L'Estelan, ap. 205. E-mail: aynur.maharramova@etu.univ-amu.fr

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ МЕТОДОВ УПРАВЛЕНИЯ СОВРЕМЕННЫМ ПРЕДПРИЯТИЕМ

ГВОЗДИНСКИЙ А.Н., СЕРИК Е.Э.

Исследуются интеллектуальные системы управления в производственном планировании современного промышленного предприятия. Предлагается оптимизационный метод решения производственных задач для распределения ресурсов предприятия.

Введение

В настоящее время новейшие достижения математики и современной вычислительной техники все более широко применяются в экономических исследованиях и планировании. Накоплен достаточный опыт постановки и решения экономических задач с помощью математических методов. Особенно успешно развиваются методы оптимального планирования.

Высокий уровень капиталовложений, необходимых при создании высокотехнологичных производств, требует эффективного использования средств, которое в первую очередь достигается путем оптимизации структуры и параметров таких производств на основе применения новых информационных технологий и математического моделирования на базе современной вычислительной техники. Особое место в этом комплексе задач занимают проблемы выбора и принятия решений. При этом процесс решения происходит в условиях ограничений, обусловленных спецификой функционирования системы: сложность математической модели, большое число рассматриваемых альтернативных вариантов, значительная неопределенность исходной информации. В этих условиях решающую роль должны сыграть модели и методы поддержки принятия решений, основанные на последних достижениях математического моделирования и теории принятия решений.

Актуальность исследования. Производственное планирование современного промышленного предприятия является тем инструментом, который обеспечивает необходимой информацией остальные процессы управления. Успешность решения подавляющего большинства экономических задач зависит от наилучшего, самого выгодного способа использования ресурсов. В процессе экономической деятельности приходится распределять такие важные ресурсы, как деньги, товары, сырье, оборудование, рабочая сила. От распределения ресурсов зависит конечный результат деятельности предприятия [1, 7].

Цель исследования – изучение эффективных методов принятия решения, определение интеллектуального оптимизационного метода для реали-

зации задач управления производственного планирования.

Описание объекта исследования и постановка задачи

Целью работы является исследование деятельности предприятия, составление математической модели задачи в классе экстремальных задач и определение на основе решения оптимальной стратегии работы фирмы на определенный срок, а также анализ возможных колебаний затрат и цен реализации, которые не приведут к изменению стратегии.

Для построения математической модели введем такие условные обозначения: x_i – число изделий, закупленных в месяце i ; y_i – число проданных изделий; Z – чистая прибыль; u, v – переменные двойственной задачи; n – месяц, $n = 1, 2, 3, \dots, 6$:

$$Z = 27y_1 + 24y_2 + 24y_3 + 26y_4 + 22y_5 + 22y_6 - (1) \\ - 26y_1 - 23 - 25x_3 - 27x_4 - 21x_5 - 20x_6.$$

С любой задачей ЛП тесно связана другая линейная задача, называемая двойственной. Первоначальная задача называется прямой.

Прямая задача:

$$Z = \sum_{j=1}^n (c_j x_j) \rightarrow \max, \quad (2)$$

$$\sum_{j=1}^n (a_{ij} x_j) \leq b_i, \quad i = 1, \dots, m, \quad x_j \geq 0, \quad j = 1, \dots, n. \quad (3)$$

Двойственная задача:

$$F = \sum_{j=1}^m (b_j y_j) \rightarrow \min, \quad (4)$$

$$\sum_{j=1}^m (a_{ij} y_j) \geq c_i, \quad i = 1, \dots, n, \quad y_j \geq 0, \quad j = 1, \dots, m. \quad (5)$$

Прямая задача: сколько и какой продукции $x_j \geq 0$, $j = 1, n$, надо произвести, чтобы при заданных стоимостях единицы продукции $c_j \geq 0$, $j = 1, n$, объемах имеющихся ресурсов $b_j \geq 0$, $i = 1, \dots, m$, и нормах расходов $a_{ij} \geq 0$, $i = 1, \dots, m$, $j = 1, \dots, n$, максимизировать выпуск продукции в стоимостном выражении [1]?

Двойственная задача: какова оценка единицы каждого из ресурсов $y_i \geq 0$, $i = 1, \dots, m$, чтобы при заданных количествах ресурсов $b_i \geq 0$, $i = 1, \dots, m$, величинах стоимости единицы продукции $c_j \geq 0$, $j = 1, \dots, n$, и нормах расходов $a_{ij} \geq 0$, $i = 1, \dots, m$, $j = 1, \dots, n$, минимизировать общую расценку затрат на все ресурсы?

Двойственный симплекс-метод, применяемый к задаче в двойственной базисной форме, приводит к последовательности задач с возрастающим значением целевой функции, неотрицательными коэффициентами c_j , $j = 1, \dots, n$, и значением b_i , $i = 1, \dots, m$, любого знака. Такой метод называют методом последовательного улучшения оценок. Преобразования задачи выполняются до тех пор,

пока не будет установлено, что исходная задача не имеет допустимого решения или будет получена задача с допустимым базисным планом (все $b_i \geq 0$), который одновременно будет и оптимальным.

Решение задачи линейного программирования двойственным симплекс-методом включает следующие этапы:

1. Привести исходную задачу к каноническому виду.
2. Исключить базисные переменные из целевой функции z .
3. Проверить приведенные коэффициенты целевой функции: если все приведенные коэффициенты $c_j^1 \geq 0$, $j = 1, \dots, n$, а среди значений b_i , $i=1, \dots, m$, есть отрицательные, то задача решается двойственным симплекс-методом. Если среди приведенных коэффициентов c_j^1 есть положительные, то в системе ограничений следует преобразовать свободные члены в неотрицательные (умножив на число (-1) строки, содержащие отрицательные b_i) и решать задачу прямым симплекс-методом [2].

Применение двойственного симплекс-метода для решения поставленной задачи

Одним из универсальных методов решения задач ЛП является симплекс-метод, или метод последовательного улучшения плана. Если задача разрешима, то ее оптимальный план совпадает, по крайней мере, с одним из опорных решений системы ограничений. Именно этот опорный план и отыскивается симплекс-методом в результате упорядоченного перебора опорных решений. Упорядоченность понимается в том смысле, что при переходе от одного плана к другому соответствующие им значения целевой функции возрастают. Так как общее число опорных решений конечно, то через определенное число шагов будет найден оптимальный опорный план, либо установлена неразрешимость задачи. Чтобы получить новый опорный план, первоначальный базис преобразовывают в новый.

С геометрической точки зрения перебор опорных планов можно толковать как переход по ребрам от одной вершины многогранника решений к другой, по направлению к вершине X_1 , в которой целевая функция достигает оптимального решения.

Разработан программный продукт, куда были введены исходные данные: месяцы, затраты и цена реализации за каждый месяц. Решение предполагает использование итераций симплекс-метода для нахождения оптимального решения [2].

Анализ методов эволюционной оптимизации в системах управления производством

Средства эволюционной оптимизации включают следующие методы: нейронные сети, деревья решений, генетические алгоритмы, а также их комбинации [3,4].

Нейронные сети относят к классу нелинейных адаптивных систем, строением они условно напоминают нервную ткань из нейронов. Это набор связанных друг с другом узлов, получающих входные данные, которые осуществляют их обработку и вырабатывают на выходе определенный результат. На узлы нижнего слоя подаются значения входных параметров, на их основе производятся вычисления, необходимые для принятия решений, прогнозирования развития ситуации. Эти значения рассматривают как сигналы, которые передаются в вышележащий слой, усиливаясь или ослабляясь в зависимости от числовых значений (весов), приписываемых межнейронным связям. На выходе нейрона самого верхнего слоя вырабатывается значение, которое рассматривается как ответ, реакция всей сети на введенные начальные значения. Так как каждый элемент нейронной сети частично изолирован от своих соседей, у таких алгоритмов имеется возможность для распараллеливания вычислений. Нейроны сети обрабатывают входные данные, для которых известны и значения входных параметров, и правильные ответы на них. Обучение состоит в подборе весов межнейронных связей, которые обеспечивают наибольшую близость ответов в сети к известным правильным ответам. После обучения на имеющихся данных сеть готова к работе и может быть использована для построения прогнозов поведения объекта в будущем. Нейронные сети используются для решения задач прогнозирования, классификации или управления. Достоинство – сети могут аппроксимировать любую непрерывную функцию, нет необходимости заранее принимать какие-либо предположения относительно модели. Исследуемые данные могут быть неполными или зашумленными. Удобны при работе с нелинейными зависимостями.

Недостаток – необходимость иметь большой объем обучающей выборки. Окончательное решение зависит от начальных установок сети. Данные должны быть обязательно преобразованы к числовому виду. Полученная модель не объясняет обнаруженные знания.

Деревья решений используют разбиение данных на группы на основе значений переменных. В результате получается иерархическая структура операторов "Если..., то...", которая имеет вид дерева. Для классификации объекта или ситуации нужно ответить на вопросы, стоящие в узлах это-

го дерева, начиная от его корня. Если ответ положительный, переходят к правому узлу следующего уровня, если отрицательный – к левому узлу. Заканчивая ответы, доходят до одного из конечных узлов, где указывается, к какому классу надо отнести рассматриваемый объект. Деревья решений предназначены для решения задач классификации и поэтому весьма ограниченно применяются в области финансов и бизнеса.

Достоинство – простое и понятное представление признаков для пользователей. В качестве целевой переменной используются как измеряемые, так и не измеряемые признаки – это расширяет область применения метода.

Недостаток – проблема значимости. Данные могут разбиваться на множество частных случаев, возникает "кустистость" дерева, которое не может давать статически обоснованных ответов. Полезные результаты получают только в случае независимых признаков.

Генетические алгоритмы (ГА) на данный момент являются наиболее известным представителем эволюционных методов оптимизации. Они содержат все основные генетические операции. ГА получены в процессе обобщения и имитации в искусственных системах свойств живой природы:

- приспособляемость к изменениям среды;
- естественный отбор;
- наследования потомками наиболее "ценных" свойств родителей.

С помощью ГА можно улучшать работу поисковых систем, которые требуют обработки больших массивов информации; оптимизировать работу нефтяных трубопроводов; распределять инструменты в металлообрабатывающих цехах; осуществлять оптимизацию профилей балок в строительстве. Решение задач комбинаторной оптимизации является одной из основных областей применения ГА. Генетические алгоритмы имитируют процесс естественного отбора в природе. Все решения описываются набором чисел или величин нечисловой природы. Поиск оптимального решения похож на эволюцию популяции индивидов, которые представлены наборами их хромосом. В этой эволюции действует три механизма:

- отбор сильнейших наборов хромосом, которым соответствуют наиболее оптимальные решения;
- скрещивание – получение новых индивидов при помощи смешивания хромосомных наборов отобранных индивидов;
- мутация – преобразование хромосомы, случайное изменение одного или несколько генов (чаще – одного).

Наиболее популярное приложение генетического алгоритма – оптимизация многопараметрических

функций. Реальные задачи формируются как поиск оптимального значения сложной функции, зависящей от некоторых n – входных параметров. Сила ГА – в способности манипулировать одновременно многими параметрами. В одних случаях получено точное решение функции, в других – решением считается любое значение, лучшее некоторой заданной величины.

Формулировка задачи эволюционной оптимизации

Для построения математической модели за основу возьмем один из методов интеллектуального анализа данных – генетические алгоритмы, и на их базе будем решать поставленную задачу [5].

Формально, генетический алгоритм – это любой алгоритм поиска приближенного решения поставленной задачи, построенный по следующей схеме.

Шаг 0. Создать случайную начальную популяцию $\{S_k(0)\}$.

Шаг 1. Вычислить приспособленность $f(S_k)$ каждой особи (S_k) популяции $\{S_k(t)\}$.

Шаг 2. Производя отбор особей из $\{S_k(t)\}$ в соответствии с их приспособленностями $f(S_k)$ и применяя генетические операторы (скрещивания и мутации) к отобранным особям для получения потомства, сформировать популяцию следующего поколения $\{S_k(t+1)\}$.

Шаг 3. Повторять шаги 1,2 для $t = 0, 1, 2, \dots$ до тех пор, пока не выполнится некоторое условие окончания эволюционного поиска (прекращается рост максимальной приспособленности в популяции, число поколений t достигает заданного предела).

Возможны различные варианты генетического алгоритма, которые отличаются по схемам отбора особей из текущего поколения в новое, операторам скрещивания и мутации хромосом особей, по форме их представления. Традиционный вид генетического алгоритма базируется на следующей частотной схеме:

- цепочки символов в хромосомах бинарные (символы S_{kj} принимают значения 0 либо 1), длина цепочек постоянна ($N = \text{const}$);
- метод отбора пропорционально-вероятностный;
- скрещивание производится по схеме одноочечного скрещивания.

Рассмотрим задачу распределения ресурсов на примере решения задачи оптимального раскроя. На заготовочный участок радиощахи поступили листы фольгированного гетинакса размером $1_1 * 1_2$ мм. Его нужно разрезать на заготовки для печатных плат: Π_1 размером $1_3 * 1_4$ мм, Π_2 - $1_5 * 1_6$ мм, Π_3 - $1_7 * 1_8$ мм, Π_4 - $1_9 * 1_{10}$ мм, необходимость в которых $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ единиц соответственно.

Построить модель и на ее основании составить экстремальную задачу, решение которой дает минимальное количества разрезанных выходных листов. Найти решение задачи в условиях минимализации отходов раскроя. Предусмотреть возможность максимизации количество комплектов печатных плат при условии выполнения комплектности $\Pi_1 : \Pi_2 : \Pi_3 : \Pi_4 = k_1 : k_2 : k_3 : k_4$ (таблица).

Параметры	Значение	Параметры	Значение	Параметры	Значение
l_1	136	l_6	88	Π_1	270
l_2	158	l_7	46	Π_2	200
l_3	92	l_8	40	Π_3	310
l_4	36	l_9	50	Π_4	90
l_5	51	l_{10}	50	Комплектность	3:2:1:5

Исходные данные: $l_1 * l_2$, Π_1 , Π_2 , Π_3 , Π_4 , k_1 , k_2 , k_3 , k_4 .

Используемые переменные: $\Pi_i = A_i$ – необходимое количество заготовок i -го типа; C_j – отход при раскрое единиц исходного материала j -м способом; $j = 1, m$ – количество вариантов раскроя; a_{ij} – количество заготовок i -го типа ($i = 1, 4$), полученных при раскрое единицы исходного материала j -м вариантом; i – вид заготовки; q – количество изделий ($\Pi_1, \Pi_2, \Pi_3, \Pi_4$); a – количество разрезанных листов; K – количество типов листов, поступающих на раскрой ($K = 1$); l_i – комплектность – количество заготовок i -го вида в одном комплекте; x_j – количество единиц исходного материала, который раскраивается j -м вариантом.

Формализация исследуемой задачи

Сформируем ограничения для данной задачи:

$$\sum_{j=1}^m x_j = a, \quad (6)$$

$$x_j \geq 0, \quad j = 1, m, \quad (7)$$

$$\sum_{j=1}^m a_{ij} x_j = \Pi_i. \quad (8)$$

Функция цели имеет вид:

$$F(x) = \sum_{j=1}^m x_j \rightarrow \min, \quad (9)$$

$$F(x) = a \rightarrow \min. \quad (10)$$

Минимизация отходов. Функция цели имеет вид:

$$\sum_{j=1}^n C_j x_j \rightarrow \min. \quad (11)$$

Максимизация комплектов. Необходимо предусмотреть возможность максимизации комплектов печатных плат при условии выполнения комплектности $\Pi_1 : \Pi_2 : \Pi_3 : \Pi_4 = k_1 : k_2 : k_3 : k_4$.

Функция цели имеет вид:

$$F(x) = X \rightarrow \max, \quad (12)$$

где $X = \frac{1}{\Pi_1 k_1} (\sum_{j=1}^m a_{j1} x_j)$, а ограничения (8) примут

следующий вид:

$$\sum_{j=1}^p a_{j1} x_j = \Pi_1 k_1 X, \quad \sum_{j=1}^p a_{j2} x_j = \Pi_2 k_2 X,$$

$$\sum_{j=1}^p a_{j3} x_j = \Pi_3 k_3 X, \quad \sum_{j=1}^p a_{j4} x_j = \Pi_4 k_4 X. \quad (13)$$

Алгоритмы решения исследуемых задач

Для начала рассмотрим решение данной задачи на основе методов линейного программирования. Алгоритм симплекс-метода разделяют на два этапа нахождения: начального опорного

плана задачи и оптимального плана, или установление факта неограниченности целевой функции на множество планов задачи.

Алгоритм симплекс-метода применяется лишь для канонической формы задач, поэтому задачу в других формах нужно изображать также в канонической.

Общая постановка задачи линейного программирования – определить величины $x_j \geq 0, j = 1, n$, которые максимизируют линейную форму

$$Z = \sum_{j=1}^n x_j C_j \rightarrow \max \quad (14)$$

при условиях существования ограничений вида

$$AX \leq B, \quad (15)$$

$A = \| a_{ij} \|_{mn}, X = \| x_1, x_2, \dots, x_n \|, B = \| b_1, b_2, \dots, b_n \|$.

Симплекс-метод применяется лишь для канонической формы задачи линейного программирования, поэтому, чтобы начать решение с помощью симплекс-метода, необходимо превратить систему неравенств в эквивалентную систему уравнений путем введения новых неотъемлемых переменных $x_{n+1}, x_{n+2}, \dots, x_{n+m}$.

Вследствие этого система приобретает вид:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} &= b_2, \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} &= b_m. \end{aligned} \quad (16)$$

Анализ методов традиционной оптимизации имеет ряд недостатков, связанных с ограничением на размерность задач. Поэтому предлагается использовать генетические алгоритмы для решения совместной задачи [4].

Требуется подобрать алгоритм, с помощью которого можно эффективно решить задачу. Был выбран генетический алгоритм по следующим причинам:

во-первых, задача раскроя является ОТ-полной;

во-вторых, время получения решения для заказчиков предполагается порядка десяти секунд для размерности 100, это достаточно высокие требования по времени.

Метод муравьиных колоний

Имитация самоорганизации муравьиной колонии составляет основу муравьиных алгоритмов оптимизации. Колония муравьев может рассматриваться как многоагентная система, в которой каждый агент (муравей) функционирует автономно по очень простым правилам.

Основу поведения муравьев составляет самоорганизация, механизмы которой обеспечивают теоретически оптимальное поведение. Принципы его состоят в достижении системой некоторой глобальной цели в результате низкоуровневого взаимодействия ее элементов. Здесь имеется в виду использование системой только локальной информации, при этом исключается любое централизованное управление и обращение к внешнему образу системы [3].

Генетический алгоритм

Генетические алгоритмы используют аналогию между естественным отбором и процессом выбора наилучшего решения из множества возможных. Его суть состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения [2].

В каждой клетке животного содержится вся информация этой особи. Эта информация записана в виде набора очень длинных молекул ДНК. Каждая молекула ДНК – это цепочка, состоящая из молекул нуклеотидов четырех типов, обозначаемых А, Т, С и G. Собственно, информацию несет порядок следования нуклеотидов в ДНК. В животной клетке каждая молекула ДНК окружена оболочкой – такое образование называется хромосомой. Каждое врожденное качество особи кодируется определенной частью хромосомы, которая называется геном этого свойства. При размножении животных происходит слияние двух родительских половых клеток и их ДНК взаимодействуют, образуя ДНК потомка. Основным способом взаимодействия – кроссовер (скрещивание). При наследовании возможны мутации из-за радиоактивности или других влияний, в результате чего могут изменяться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Выбор пар родителей из популяции для порождения потомков производит оператор отбора, а выбор особей для уничтожения – оператор редукции.

Генетический алгоритм – это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются как ана-

лог механизма генетического наследия, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

В процессе работы генетического алгоритма все указанные выше операторы применяются многократно и ведут к изменению исходной популяции. Поскольку операторы отбора, скрещивания, мутации и редукции по своей сути направлены на улучшение отдельной особи, то результатом их работы является постоянное улучшение популяции.

Преимуществом генетических алгоритмов перед другими является простота их реализации, относительно высокая скорость работы, параллельный поиск решения.

Блочное программирование

Математическое программирование – это прикладная отрасль математики, которая является теоретической основой решения задач оптимального планирования. К этому классу относят алгоритмические методы. Наиболее часто используются линейные и нелинейные преобразования. Значительная вычислительная сложность алгоритмических методов зачастую препятствует их эффективному применению в практических задачах. Блочное программирование – метод решения сложных задач линейного программирования путем разложения модели на блоки. Крупноразмерная модель (включающая много показателей в исходной таблице) сводится к нескольким моделям меньшей размерности. Получившиеся задачи решаются вместе по специальным правилам согласования. Необходимость такого подхода обосновывается тем, что с ростом размерности трудоемкость решения задач растет невероятно быстро [4].

Среди теоретических схем блочного программирования наиболее известны две: метод декомпозиции Данцига-Вульфа и метод планирования на двух уровнях Корнай-Липтака. Обе они представляют собой последовательные (итеративные) пересчеты, взаимно увязывающие решения главной, “отраслевой” задачи и локальных задач предприятий. Различие же между ними состоит в том, что схема Данцига-Вульфа построена по принципу “централизованное определение цен – децентрализованное определение наилучших возможностей”, а схема Корнай-Липтака – по принципу “централизованное имитирование возможностей – децентрализованное выявление эффекта от их использования”. В обоих случаях важную роль играют двойственные оценки, причем их оптимальный уровень выявляется вместе с оптимальным распределением ресурсов, т.е. собственно планом (именно в этом состоит принцип оптимального планирования).

Характерной особенностью этого метода является использование так называемой «координирующей задачи», которая по сравнению с исходной имеет небольшое число строк (не намного превышающее число связывающих ограничений) и очень много столбцов. При этом для решения координирующей задачи не требуется задание всех столбцов в явном виде. Они генерируются в процессе использования симплекс-метода. Поэтому такой подход называется методом генерации столбцов. Алгоритм включает итерационный обмен между множеством независимых подзадач, целевые функции которых включают варьируемые параметры, и координирующей задачей. В подзадачу вводится ряд параметров (двойственные переменные, оценки), получаемых при решении координирующей задачи. В свою очередь в координирующую задачу вводятся решения подзадач, которые оптимальным образом комбинируются и служат для получения новых оценок. Последние вновь вводятся в подзадачу, и итеративный процесс продолжается вплоть до этапа, на котором удовлетворяется критерий оптимальности. Такая процедура имеет изящную экономическую интерпретацию: ее, например, можно понимать как процедуру децентрализованного планирования, когда основной планирующий (управляющий) орган системы координирует функционирование отдельных подсистем с помощью цен на ограниченные ресурсы.

Выводы

В результате данного исследования можно сделать вывод, что наиболее распространенным методом остается блочное программирование, поскольку позволяет находить наилучшие решения за короткое число итераций. Генетический алгоритм реже используется, так как он имеет значительный недостаток в процессе кодировки, хотя является самым быстрым. Муравьиный алгоритм – самый молодой, однако при всей своей сложности он продуктивен.

Исследованы проблемы организационного управления на предприятиях, разработана модель объекта управления и процессов управления, выявлены наиболее значимые, с точки зрения повышения эффективности, процессы принятия решений в системах управления раскроем. Исследованы методики поддержки принятия решений в процессах управления поступлением, хранением ресурсов, основанных на предложенной модели затрат. Разработана методика принятия решений в целях оптимизации системы управлением материально-технических ресурсов, используемых на предприятиях. Проведен анализ эффективности разработанной методики управления запасами.

Научная новизна исследования заключается в обосновании экономического механизма управления запасами материально-технических ресурсов и разработки методического обеспечения процесса принятия решения по оптимизации объемов запасов.

Практическим значением исследования являются: обоснования места, роли и рациональной структуры запасов в системе планирования сбытовой деятельности предприятия в современных условиях; разработка концептуальных подходов к решению оптимизационных задач в области управления материально-техническими ресурсами, определение факторов стратегии управления запасами, постоянная партия объема пополнения заказа, уровень восполнения запаса.

Литература: 1. Гвоздинский А.Н. Методы аналитической обработки информации [Текст] / А.Н. Гвоздинский, Е.Г. Климов // Радиоэлектроника и информатика. 2000. №4. С.111-112. 2. Гвоздинский А.Н. Принципы и методы оптимизации в современных организационных системах управления // Научно-технический журнал [Текст] / А.Н. Гвоздинский, А.Е. Козлова, А.О. Дроздов // Х: Бионика интеллекта, 2013. №2 (81). С.66-70. 3. Гвоздинский А.Н. Методы оптимизации в системах принятия решений [Текст] / А.Н. Гвоздинский, Н.А. Якимова, В.О. Губин // Х: ХНУРЭ, 2006. 327 с. 4. Гвоздинский А.Н. Методы оптимизации в системах организационного управления [Текст] / А.Н. Гвоздинский, В.Л. Шергин, В.О. Губин // Х: ХНУРЭ, 2014. 395 с. 5. JASA: Глава 2,6. Двойственный метод <http://jasa.org.ua/> 6. Лэддон Л.С. Оптимизация больших систем. М: Наука, 1975. С.65-83. 7. Гвоздинский А.Н., Корниенко Ю.Н. Исследование интеллектуальных систем принятия решений в производственном планировании // АСУ и приборы автоматизации. 2009. Вып. 147. С. 81-84.

Transliterated bibliography:

1. Gvozinskiy A.N. Metodyi analiticheskoy obrabotki informatsii [Tekst] / A.N. Gvozinskiy, E.G. Klimko // Radioelektronika i informatika. 2000. #4. S.111-112.
 2. Gvozinskiy A.N. Printsipy i metodyi optimizatsii v sovremennyih organizatsionnyih sistemah upravleniya: Nauchno-tehnicheskij zhurnal [Tekst] / A.N. Gvozinskiy, A.E. Kozlova, A.O. Drozdov // H: Bionika intellekta, 2013. #2 (81). S.66-70.
 3. Gvozinskiy A.N. Metodyi optimizatsii v sistemah prinyatiya resheniy [Tekst] / A.N. Gvozinskiy, N.A. Yakimova, V.O. Gubin // H: HNURE, 2006. 327 s.
 4. Gvozinskiy A.N. Metody optimizatsii v sistemah organizatsionnogo upravleniya [Tekst] / A.N. Gvozinskiy, V.L. Shergin, V.O. Gubin // H: HNURE, 2014. 395 s.
 5. JASA: Glava 2, 6. Dvoystvennyiy metod <http://jasa.org.ua/>
 6. Lesdon L.S. Optimizatsiya bolshih sistem. M: Nauka, 1975. S.65-83.
 7. Gvozinskiy A.N., Kornienko Ju.N. Issledovanie intellektual'nyh sistem prinjatija reshenij v proizvodstvennom planirovanii // ASU i pribory avtomatiki. 2009. Vyp. 147. S. 81-84.

Поступила в редколлегию 17.08.2017

Рецензент: д-р техн. наук, проф. Куземин А.Я.

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решения в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Академика Ляпунова, 7, кв.9, тел.702-38-23.

Серик Екатерина Эдуардовна, студентка группы КН-13-б кафедры искусственного интеллекта ХНУРЭ. Научные интересы: методы принятия решения в системах искусственного интеллекта. Адрес: Украина, 61000, Харьков, пер. Афанасьевский, дом 36, тел. 066-340-08-76.

Anatoly Gvozdinsky, PhD, Professor, Artificial Intelligence Department KNURE. Scientific interests: optimization of decision-making procedures in complex control systems. Address: Ukraine, 61166, Kharkov, ul. Academician Lyapunov, 7, ap. 9, tel.702-38-23.

Serik Ekaterina Eduardovna, student of group KH-13-6, Artificial Intelligence Department of KNURE. Scientific interests: methods of decision making in artificial intelligence systems. Address: Ukraine, Kharkiv, per. Afanasyevsky, house 36, tel. 066-340-08-76.

Описуються методи і реалізація апаратно-програмного забезпечення безумовного паралельного синтезу тестів на основі булевих похідних для логіки, поданої у вигляді чорної скриньки і описаної кубітним покриттям. Наводяться теоретичні основи застосування методів і оцінка їх ефективності для широкого класу цифрових схем, реалізованих в програмованих логічних пристроях. Пропонуються інноваційні методи обчислення логічних похідних і дедуктивного моделювання несправностей для функціональних елементів, описаних кубітним покриттям.

Ключові слова: синтез тестів, проектування і верифікація SoC, кубітне покриття, цифрова схема, моделювання несправностей, булева похідна.

1. Реалізація логічної функціональності на елементах пам'яті

Поняття адресного виконання логічних операцій, реалізованих на елементах пам'яті LUT в програмованих логічних пристроях (PLD), дає потенційну можливість створювати на кристалі тільки адресний простір, максимально технологічний для вбудованого відновлення працездатності всіх компонентів, що беруть участь у формуванні функціональності [1-3]. Актуальність створення адресного простору для компонентів підтверджується розподілом логіки і пам'яті на кристалі, представленим на рис. 1, де після 2020 року на чіпі буде тільки один відсоток логіки і 99 відсотків пам'яті.

Тенденція до збільшення пам'яті дає можливість вбудованого відновлення працездатності відмовлених осередків за рахунок виділених додаткових ресурсів для їх ремонту (spare logic cells). Проблема автономного усунення дефектів (самовідновлення працездатності) логічних елементів пов'язана з відсутністю у них адреси. Але вирішити її можна, якщо зв'язки між елементами логіки зробити гнучкими за допомогою програми опису структури, вміщеної в пам'яті, яка з'єднає логічні компоненти в схему. Крім структури взаємодії елементів, пам'ять повинна містити порядок їх обробки. У разі виникнення дефекту в одному з адресованих логічних елементів система вбудованого тестування відновить його працездатність шляхом переадресації на завідомо справний аналог з ремонтного запасу. Просто вирішується проблема підвищення якості та надійності цифрових систем на кристалах шляхом створення інфраструктури вбудованого тестування, діагностування, оптимізації та відновлення працездатності за рахунок апаратної надмірності і зменшення швидкодії виконання функціональних операцій [2-5].

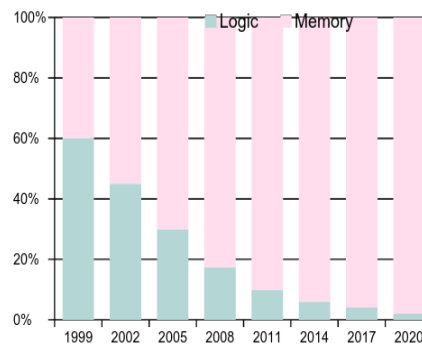


Рис. 1. Розподіл пам'яті і логіки на кристалі
Мета – істотне підвищення швидкодії синтезу тестів і дедуктивної верифікації для blackbox функціональності логічних компонентів за рахунок використання компактних описів у формі кубітних покриттів і паралельного виконання мінімального числа реєстрових логічних операцій (shift, or, not, pxor).

Задачі:

- 1) Метод генерації тестів для blackbox функціональності логічних схем на основі використання кубітних покриттів і паралельного виконання реєстрових логічних операцій (shift, or, not, pxor).
- 2) Метод обчислення булевих похідних для синтезу тестів на основі використання кубітних покриттів.
- 3) Метод синтезу тестів на основі використання булевих похідних, поданих векторами в форматі кубітних покриттів.
- 4) Метод дедуктивного моделювання несправностей для функціональних елементів, поданих векторами кубітних покриттів.
- 5) Процесор моделювання несправностей і справної поведінки на основі кубітного опису функціональності, інтегруючий розроблені методи в хмарний або SoC-інфраструктурний сервіс тестування.

Сутність дослідження – розробка методів генерування вхідних тестових послідовностей і оцінки їх якості для функціональних логічних компонентів шляхом паралельного виконання реєстрових логічних операцій (shift, or, not, pxor) над кубітним покриттям і його похідними в структурі процесора кубітного моделювання.

Тестування і діагностування запам'ятовуючих пристроїв реалізується за допомогою спеціальних алгоритмів, що генерують тести: марш; нуль і одиниця, що біжать; логарифмічний розподіл [1-5]. Якщо елементи пам'яті виконують функції логічних схем (reusable logic), то для них слід генерувати тести перевірки функціональності, не прив'язані до фізики процесів зберігання даних. Тому далі розглядається метод синтезу тестів для функціональних схем, представлених у формі black box, опис яких задається кубітним вектором [6-7].

Ринкова привабливість застосування квантових методів обчислень при створенні комп'ютерних структур в кіберпросторі заснована на використанні кубітних моделей даних, орієнтованих на паралельне вирішення завдань проектування, тестування, дискретної оптимізації [5,8], завдяки збільшенню витрат пам'яті. Не вдаючись в деталі фізичних основ квантової механіки, що стосуються недетермінованої взаємодії атомних часток [9,10], далі використовується поняття кубіта як двійкового вектора для спільного і одночасного завдання булеана станів у дискретній області кіберпростору на основі лінійної суперпозиції унітарних кодів, орієнтованих на паралельне виконання операцій. У теорії квантових обчислень, яка швидко розвивається, вектори станів утворюють квантовий регістр з n кубітів, що формують унітарний або гільбертовий [11] простір H , розмірність якого має ступеневу залежність від числа кубітів.

Мотивація нового підходу для проектування комп'ютерних систем обумовлена появою хмарних сервісів у рамках нової кіберкультури Internet of Things, яка представляє собою спеціалізовані системи, що реалізуються в апаратурі або в програмному продукті. Будь-який компонент функціональності, так само як і структура системи, представляється векторною формою, впорядкованою за адресами, таблицею істинності, що реалізується за допомогою пам'яті. Логічні функції в традиційному виконанні reusable logic не розглядаються. Від цього частково зменшується швидкодія, але, з огляду на те, що 94% SoC-кристала становить пам'ять [2,3], останні 6% будуть імплементуватися в пам'яті, що не критично для більшості хмарних сервісів. Практично для створення ефективних комп'ютерних структур слід використовувати теорію, засновану на обчислювальних компонентах високого рівня абстракції: адресована пам'ять і транзакція.

Особливість організації даних в класичному комп'ютері полягає в адресації біта, байта або іншого компонента. Адресація створює проблему в обробці асоціації елементів множини, які не адресуються та не мають порядку за визначенням. Рішенням може бути процесор, де образ універсуму, з n унітарно кодovаних примітивів, використовує суперпозицію для формування булеана $|B(A)|=2^n$ усіх можливих станів [4,12].

Коректність використання прикметника «кубітна» для моделей цифрових пристроїв заснована на порівнянні лінійної і булевої алгебри Кантора з алфавітом $A^k = \{0,1, X, \emptyset\}$. Тут перші два символи – примітиви. Третій визначається суперпозицією: $X=0\cup 1$. У гільбертовому просторі лінійна алгебра оперує примітивами $|\alpha|0\rangle, |\beta|1\rangle$ кубіта, третій символ також є суперпозицією двох

складових: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. У лінійній алгебрі немає символу, відповідного порожній множині, він є похідною від функції, зворотної по відношенню до суперпозиції. В теорії множин такою операцією є перетин, який дає порожню множину на примітивах: $\emptyset = 0 \cap 1$. У гільбертовому просторі такою операцією є скалярний добуток або функція Дірака $\langle \alpha | \beta \rangle$ [11], яка має геометричну інтерпретацію: $\langle \alpha | \beta \rangle = |\alpha| |\beta| \cos \angle(\alpha, \beta)$. Якщо проєкції a і b вектора квантового стану ортогональні, виходить:

$$\langle \alpha | \beta \rangle = |\alpha| |\beta| \cos \angle(\alpha, \beta) = |\alpha| |\beta| \cos 90^\circ = 0.$$

Скалярний добуток ортогональних векторів дорівнює нулю, що є аналогом символу порожньої множини в алгебрі Кантора. Таблиця відповідності алгебри множин та лінійної алгебри, що зображена нижче, підтверджує властивості ізоморфізму між символами булеана і станами кубіт-вектора:

Boolean $A^k =$	0	1	$X = 0 \cup 1$	$\emptyset = 0 \cap 1$
Qubit $ \psi\rangle =$	$ 0\rangle$	$ 1\rangle$	$\alpha 0\rangle + \beta 1\rangle$	$\alpha 0\rangle \beta 1\rangle$

Отже, структуру даних «булеан» можна розглядати як детермінований образ квантового кубіта в алгебрі логіки, елементи якої унітарно кодуються двійковими векторами і мають властивості суперпозиції, паралелізму і змішування. Це дає можливість використовувати запропоновані кубітні моделі для підвищення швидкодії аналізу цифрових пристроїв на класичних обчислювачах, а також без модифікації – у квантових комп'ютерах, які з'являться через кілька років на ринку електроніки.

2. Кубітний метод синтезу тестів

Пропонується метод синтезу тестів, що використовує кубітні вектори або Q -покриття функціональних примітивів цифрових пристроїв, який характеризується компактністю опису даних і паралелізмом виконання логічних операцій. Q -покриття є векторна форма опису поведінки цифрового пристрою, де кожен розряд має адресу, що формується двійковими станами його вхідних змінних:

$$Q = (Q_1, Q_2, \dots, Q_i, \dots, Q_n), \quad Q_i = \{0,1\}, \quad Q_i = Q(i), \quad i = (X_1 X_2, \dots, X_i, \dots, X_n).$$

Q -тест є векторна форма неявного завдання тестових послідовностей цифрового пристрою, де координати вектора формують впорядковану послідовність двійкових наборів, що подаються на вхідні змінні за правилом:

$$Q_i = Q(i) = \begin{cases} 1 \rightarrow (X_1 X_2, \dots, X_i, \dots, X_n) = i, \\ 0 \rightarrow (X_1 X_2, \dots, X_i, \dots, X_n) = \emptyset. \end{cases}$$

Іншими словами, якщо координата вектора $Q(i)=1$, то тестовий набір, складений з двійкових розрядів, які формують десяткову адресу i , подається на входи пристрою. В іншому випадку, при нульовому значенні координати $Q(i)=0$, такий тестовий набір відсутній.

Абстрагуючись від поняття таблиці істинності, пропонується формальний безумовний алгоритм кубітного синтезу тестів для функціональних примітивів на основі Q-покриття, стосовно раніше розглянутого прикладу:

1) Інвертування всіх розрядів Q-покриття функціонального елемента, що має три вхідні змінні:

Q	1	0	0	1	0	0	1	0
\bar{Q}	0	1	1	0	1	1	0	1

2) Логічний зсув номерів розрядів інвертованого кубітного вектора відповідно до послідовності номерів:

$$\begin{aligned} Q_1(X_1) &= 45670123, \\ Q_2(X_2) &= 23016745, \\ Q_3(X_3) &= 10325476. \end{aligned}$$

Тут діє просте логічне правило: для першої вхідної (молодшої) змінної виконується паралельний обмін даними між сусідніми координатами, для другої вхідної змінної реалізується обмін даними, але вже між сусідніми парами координат, для третьої змінної виконується обмін даними між сусідніми тетрадами координат вектора адрес.

Узагальнення операцій зсуву для функціональності, що містить 4 змінних, представлено на рис. 2.

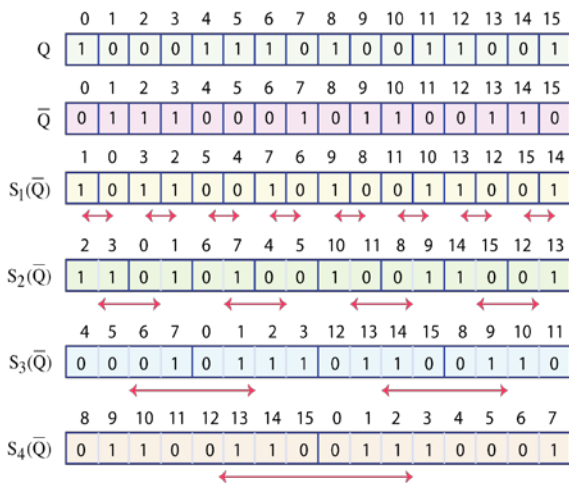


Рис. 2. Операції зсуву для 4-вхідової функціональності

Збільшення кількості змінних принципово не змінює сутностей зсуву даних: зустрічний зсув двох бітів, пар бітів, тетрад бітів, вісімок, ...

Процедура зустрічного зсуву даних в інвертованому кубітному векторі є найбільш часозатратною в алгоритмі синтезу тестів для функціональних елементів. Тому її швидкодія матиме максимальне значення при апаратній реалізації зсувних операцій.

3) Порівняння за допомогою операції еквівалентності отриманих, в даному випадку, трьох інвертованих і переупорядкованих Q-векторів з вихідним Q-покриттям функціонального елемента:

Q	1	1	0	1	0	1	1	0
\bar{Q}	0	0	1	0	1	0	0	1
$S_1(\bar{Q})$	1	0	0	1	0	0	1	0
$S_2(\bar{Q})$	1	0	0	0	0	1	1	0
$S_3(\bar{Q})$	0	0	0	1	0	1	1	0
$T_1 = Q \oplus S_1(\bar{Q})$	1	0	1	1	1	0	1	1
$T_2 = Q \oplus S_2(\bar{Q})$	1	0	1	0	1	1	1	1
$T_3 = Q \oplus S_3(\bar{Q})$	0	0	1	1	1	1	1	1
$T = T_1 \vee T_2 \vee T_3$	1	0	1	1	1	1	1	1

4) Диз'юнкція отриманих трьох векторів формує Q-тест, одиничні координати яких визначають тестові набори для перевірки всіх поодиноких константних несправностей зовнішніх входів і виходів.

З огляду на істотність зустрічних реєстрових зсувів, нижче пропонується універсальний алгоритм отримання перестановок кубітних фрагментів залежно від номера вхідної змінної, що має три вкладених цикли:

1) Завдання і-номера вхідної змінної або кроку 2^i для зустрічного зсуву даних в кубіті: $i = \overline{1, n}$.

2) Формування циклу обробки кубіта з вже заданим кроком 2^i (2,4,8,16...), кратним ступені двійки: $j = \overline{0, 2^n - 1, 2^i}$. Залежно від номера вхідної змінної і формуються такі послідовності індексу $j=f(i)$: [(0,2,4,6...), (0,4,8,12...), (0,8,16,32...)].

3) Задання циклу зустрічного зсуву даних для пари сусідніх груп: $t = j, j + 2^{i-1} - 1$. Значення індексу $t=f(i,j)$: обробка кубіта першої змінної – (0,0; 2,2; 4,4...), другої змінної – (0,1; 4,5; 8,9...), третьої змінної – (0,3; 8,11; 16,19...). Виконання операцій зсуву за допомогою використання буферного регістра В (рис. 3):

$$(B_t = Q_{i,t+j}) \rightarrow (Q_{i,t+j} = Q_{i,t}) \rightarrow (Q_{i,t} = B_t).$$

Кінець алгоритму зсуву даних в регістрах.

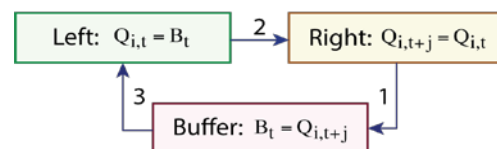


Рис. 3. Зустрічний зсув сусідніх частинкубіта
Обчислювальна складність послідовного алгоритму для обробки кубітного покриття функціональності, що має n вхідних змінних, дорівнює $q = 3n2^n$.

Приклад 1. Синтезувати тест для перевірки функціональності, заданої логічним рівнянням:

$$Y = \underline{X}_1 \underline{X}_2 \underline{X}_3 \vee \underline{X}_1 \underline{X}_2 \underline{X}_3 \vee \underline{X}_1 X_2 X_3 \vee \underline{X}_1 X_2 X_3.$$

Результати виконання алгоритму синтезу тесту за Q-вектором на основі послідовного виконання чотирьох реєстрових логічних операцій (not, shift, xor, or) представлені у вигляді:

\bar{Q}	0 0 1 1 1 0 0 1
Q	1 1 0 0 0 1 1 0
$S_1(Q)$	1 1 0 0 1 0 0 1
$S_2(Q)$	0 0 1 1 1 0 0 1
$S_3(Q)$	0 1 1 0 1 1 0 0
$T_1 = Q \oplus S_1(Q)$	0 0 0 0 1 1 1 1
$T_2 = Q \oplus S_2(Q)$	1 1 1 1 1 1 1 1
$T_3 = Q \oplus S_3(Q)$	1 0 1 0 1 0 1 0
$T = T_1 \vee T_2 \vee T_3$	1 1 1 1 1 1 1 1

На рис. 4 представлений секвенсор синтезу тестів для функціональних елементів, заданих кубітними покриттями. Він містить модуль управління, який розподіляє чотири синхроімпульси, створюючи цикл генерації тесту, що включає 4 паралельні операції, які виконуються послідовно: 0) Початкова стартова операція, що ініціюється сигналом Start, передбачає завантаження в регістр кубітного покриття. 1) Синхроімпульс Clk N активує виконання операції інверсії Not над вмістом регістра кубітного покриття. 2) Синхросигнал Clk S активує виконання операцій зустрічного зсуву над вмістом, в даному випадку трьох, регістрів, отримуючи такі результати: S_1 (not Q), S_2 (not Q), S_3 (not Q). 3) Потім синхросигнал Clk C ініціює виконання паралельних операцій порівняння (в даному випадку для трьох регістрів) отриманих кандидатів в тест з початковим кубітним покриттям: $pxor(notQ, S_1)$, $pxor(notQ, S_2)$, $pxor(notQ, S_3)$. 4) Синхросигнал Clk U активує виконання ог-операції над кандидатами в тест, в даному випадку реалізацію логічного об'єднання вмісту трьох регістрів:

$$T = or[nxor(notQ, S_1), nxor(notQ, S_2), nxor(notQ, S_3)]$$

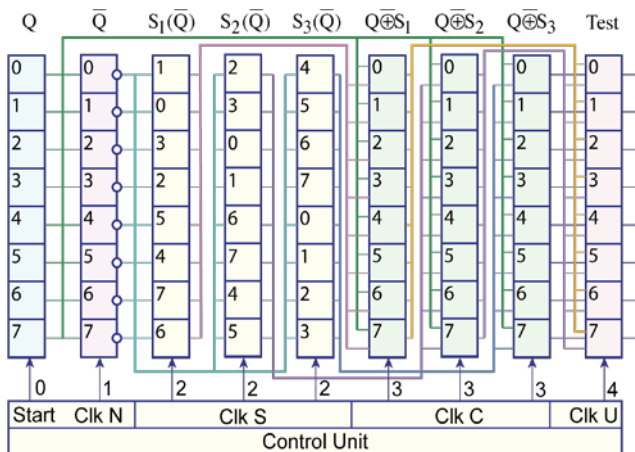


Рис. 4. Секвенсор синтезу тестів

Якщо не економити на апаратурі, то швидкодію тестового генератора можна довести до чотирьох автоматних тактів $q=4$ паралельного виконання логічних реєстрових операцій, що дає можливість вбудовувати секвенсор в BIST-інфраструктуру цифрових систем на кристалах для online тесту-

вання функціональних елементів. При цьому сумарна кількість 2^n -розрядних регістрів дорівнюватиме $N(n)=1+1+n+n+1=(2n+3)$, а загальний обсяг реєстрової пам'яті в бітах буде дорівнювати $N(R)=(2n+3) 2^n$, де n – число вхідних змінних функціонального елемента.

3. Обчислення булевих похідних для Q-синтезу тестів

Розглянемо метод обчислення булевих похідних по кубітному покриттю для створення умов активізації вхідних змінних при синтезі кубітних тестів. Проведемо аналогію між двома формами булевих функцій для взяття похідних: аналітичною та векторною. Дослідження методу виконаємо на прикладі: $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$. Питання, що підлягають розв'язанню: 1) Визначення похідних першого порядку за аналітичною кубітною формами завдання логічної функції. 2) Верифікація отриманих умов активізації шляхом їх моделювання на одній з форм опису функціональності. 3) Синтез тестів активізації змінних логічної функції на основі обчислення похідних.

Кубітний метод взяття булевої похідної:

- 1) Визначити кубітний вектор функціональності.
- 2) Виконати хог-операцію для зсунутих назустріч один одному сусідніх частин кубіта: біти, пари, тетради.
- 3) Результат записати в обидві сусідні частини: біти, пари, тетради:

Y	Y' ₁	Y' ₂	Y' ₃
0	0	0	1
1	1	0	1
0	1	0	1
1	0	0	1
0	0	1	0
0	1	1	0
1	1	1	0
1	0	1	0

Для розглянутого прикладу виконання процедури обчислення похідної за першою змінною має вигляд: $\{a,b\}=a \oplus b$, $(0110,0110)=0101 \oplus 0011$. Для отримання похідної за другою змінною величиною слід послідовно хог-скласти сусідні пари кубіт-вектора Y, а загальний результат записати в кожен біт сусідів: $\{a,b\}=a \oplus b$: $(00,00)=01 \oplus 01$, $(11,11)=00 \oplus 11$. Для отримання похідної по третій змінній слід послідовно хог-скласти сусідні біти кубіт-вектора Y, а загальний результат записати в кожен біт сусідів: $\{a,b\}=a \oplus b$: $(1,1)=0 \oplus 1$, $(1,1)=0 \oplus 1$, $(0,0)=0 \oplus 0$, $(0,0)=0 \oplus 0$.

Звичайно, що кубіт-похідна за будь-якою вхідною змінною, як вектор, має відносну симетрію рівності підвектора з побудови: похідна першої змінної має симетричну рівність двох тетрад, похідна другої змінної має симетричну рівність

кожних сусідніх пар, похідна третьої змінної має симетричну рівність кожних сусідніх бітів.

Визначити всі похідні для функції трьох змінних:
 $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$.

Нижче представлений виключно простий процес обчислення похідних на кубітному покритті без розгляду таблиці істинності, що на формальному рівні має такий вигляд:

Y	Y ₁ '	Y ₂ '	Y ₃ '
1	0	1	1
0	0	0	1
0	0	1	0
0	1	0	0
1	0	1	1
0	0	1	1
0	0	1	1
1	1	1	1

Інтерпретація отриманих кубіт-похідних. Природно, що похідні є функції, задані векторами. Вони можуть бути записані в аналітичній формі (ДНФ) за одиничними значеннями змінних, які формують адреси осередків кубіт-вектора:

$$Y_1' = 011 \vee 111 = \bar{X}_1 X_2 X_3 \vee X_1 X_2 X_3 = X_2 X_3 (\bar{X}_1 \vee X_1) = X_2 X_3.$$

$$Y_2' = 000 \vee 010 \vee 100 \vee 101 \vee 110 \vee 111 = \bar{X}_1 \bar{X}_3 \vee X_1.$$

$$Y_3' = 000 \vee 001 \vee 100 \vee 101 \vee 110 \vee 111 = \bar{X}_1 \bar{X}_2 \vee X_1.$$

Мінімізація булевих функцій, відповідних похідним, приводить до аналітичних виразів, де відсутні змінні, за якими береться похідна. Таким чином, всі результати по обчисленню похідних від трьох форм (аналітична, таблична, векторна) завдання функції є ідентичними. Найбільш технологічним є метод обчислення похідної за кубітним покриттям. Метод має меншу обчислювальну складність в силу компактного представлення функціональності.

Як альтернатива, далі пропонується технологічно простий метод синтезу тестів на основі взяття похідних по кубітних покриттях функціональних елементів.

- 1) Початкове завдання функціональності кубітним покриттям.
- 2) Виконання операцій зустрічного зсуву частин кубіт-вектора і подальшого покоординатного хогпідсумовування для отримання векторів похідних для кожної вхідної змінної.
- 3) Логічне об'єднання векторів похідних, яке формує тест-вектор, що дорівнює за розміром кубітному покриттю.
- 4) У разі необхідності отримання мінімального тесту вирішується завдання покриття (вже на матриці кубіт-похідних) шляхом знаходження мінімального числа пар одиничних координат

кубіт-вектора всіх змінних, де пара одиниць повинна перевіряти поодинокі константні несправності кожного входу.

Рис. 5 ілюструє таблиці процесу отримання тесту відповідно до пунктів алгоритму для функціональності $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$, представленої схемною структурою.

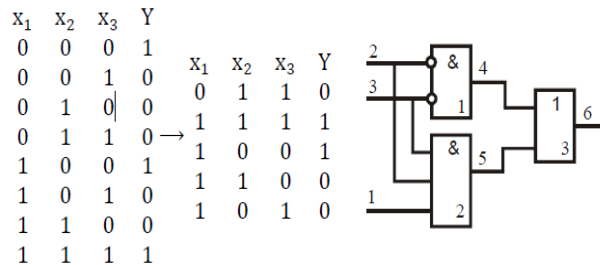


Рис. 5. Отримання тесту для структури булевої функції 3 з урахуванням виконання пункту 3 алгоритм синтезу дає мінімальний тест перевірки вхідних змінних, який містить 5 наборів, що представлено стовпцем T, а також продубльовано в явному вигляді правою таблицею. Інтерес представляє той факт, що результат виконання процедури взяття похідної по кубіт-вектору вже містить тест активізації кожної змінної. Об'єднаний тест перевіряє всі константні несправності вхідних змінних, а також може бути використаний для діагностування несправностей, оскільки для істотних входів усі похідні-вектори будуть різними. Фактично взяття похідної за змінною на кубіт-покритті формує Q-тест, не більше і не менше.

4. Дедуктивний аналіз несправностей цифрових структур

Використовується для визначення якості тесту щодо введеного класу несправностей, як правило, одиночних константних. Існує розвинена теорія дедуктивного аналізу [4], орієнтована на паралельну обробку списків несправностей. Пропонується технологічна реалізація дедуктивного моделювання на кубітній формі завдання функціональності, яка відрізняється від наведеної вище паралелізмом виконання логічних операцій, а також можливістю застосування методу для будь-яких цифрових структур.

Сукупність кубіт-похідних для всіх вхідних змінних, обчислених по кубітному покриттю, являє собою кубітну матрицю для реалізації дедуктивного методу моделювання несправностей. Рядок матриці формує умови для транспортування списків несправностей від зовнішніх входів до виходу за правилом: поодинокі значення створюють об'єднання вхідних списків, а нульові сигнали вказують на входи, списки яких повинні бути відняті з результату об'єднання. Наявність всіх нульових сигналів у рядку створює умови перетину вхідних списків між собою.

Як приклад пропонується побудова дедуктивних формул транспортування списків несправностей

від вхідних змінних до виходу функціональності, заданої вхідними наборами, кубітним покриттям з векторними похідними:

x_1	x_2	x_3	Y	X_1	X_2	X_3
0	0	0	1	0	1	1
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	0	0
1	0	0	1	0	1	1
1	0	1	0	0	1	1
1	1	0	0	0	1	1
1	1	1	1	1	1	1

У загальному випадку формула дедуктивного моделювання логічних функціональностей, представлених у вигляді кубітних векторів, має такий вигляд:

$$L = \bigvee_{i=1}^{2^n} (x_{i1} \wedge x_{i2} \wedge \dots \wedge x_{ij} \wedge \dots \wedge x_{in}) \wedge (X_{i1} \vee \dots \vee X_{ij} \vee \dots \vee X_{in}),$$

$$L = \bigvee_{i=1}^{2^n} (x_{i1} \wedge \dots \wedge x_{in}) \wedge (X_{i1} \vee \dots \vee X_{in}),$$

$$L = x \wedge X, x = x_{ij}, X = X_{ij}, i = \overline{1, 2^n}, j = \overline{1, n}.$$

Тут L – список вихідних несправностей; x – матриця вхідних тестових наборів; X – матриця кубітних похідних від кубітного покриття; n – число вхідних змінних.

Алгоритм побудови дедуктивної формули для заданої функціональності включає такі пункти:

- 1) Завдання кубіт-вектора функціональності.
- 2) Обчислення кубіт-похідних для вхідних змінних з метою отримання відповідної матриці.
- 3) Формування аналітичної або матрично-векторної форми обчислення вихідних списків несправностей шляхом логічного множення матриць вхідних тестових впливів і матриці похідних.

Нижче представлений процес обчислення аналітичної і векторної форм для дедуктивного моделювання несправностей логічної функціональності:

$$T = (000, 001, 010, 011, 100, 101, 110, 111).$$

$$Q = (011, 001, 010, 100, 011, 011, 011, 111)].$$

$$L = (000 \wedge 011) \vee (001 \wedge 001) \vee (010 \wedge 010) \vee (011 \wedge 100) \vee (100 \wedge 011) \vee (101 \wedge 011) \vee (110 \wedge 011) \vee (111 \wedge 111).$$

$$L = (\bar{x}_1 \bar{x}_2 \bar{x}_3 \wedge \bar{X}_1 X_2 X_3) \vee (\bar{x}_1 \bar{x}_2 x_3 \wedge \bar{X}_1 \bar{X}_2 X_3) \vee (\bar{x}_1 x_2 \bar{x}_3 \wedge \bar{X}_1 X_2 \bar{X}_3) \vee (\bar{x}_1 x_2 x_3 \wedge \bar{X}_1 \bar{X}_2 \bar{X}_3) \vee (x_1 \bar{x}_2 \bar{x}_3 \wedge \bar{X}_1 X_2 X_3) \vee (x_1 \bar{x}_2 x_3 \wedge \bar{X}_1 X_2 X_3) \vee (x_1 x_2 \bar{x}_3 \wedge \bar{X}_1 X_2 X_3) \vee (x_1 x_2 x_3 \wedge \bar{X}_1 X_2 X_3).$$

Обчислення векторних форм для дедуктивного моделювання несправностей основних логічних примітивів: or, and, xor представлено в наступній таблиці:

x_1	x_2	Y^V	X_1^V	X_2^V	Y^\wedge	X_1^\wedge	X_2^\wedge	Y^\oplus	X_1^\oplus	X_2^\oplus
0	0	0	1	1	0	0	0	0	1	1
0	1	1	0	1	0	1	0	1	1	1
1	0	1	1	0	0	0	1	1	1	1
1	1	1	0	0	1	1	1	0	1	1

Тут визначені два вектор-стовпці табличного задання вхідних змінних (x_1, x_2), кубіт-вектор функції or – Y^V , дві похідні (X_1^V, X_2^V) для кожної вхідної змінної. Далі показані: кубіт-вектор задання функції and і дві колонки похідних, а також кубіт-вектор функції xor і два стовпці похідних по вхідним змінним. Формули дедуктивного моделювання тривіально записуються по рядках:

$$L^V = \bar{x}_1 \bar{x}_2 (X_1 \vee X_2) \vee \bar{x}_1 x_2 (\bar{X}_1 \vee X_2) \vee x_1 \bar{x}_2 (X_1 \vee \bar{X}_2) \vee x_1 x_2 (\bar{X}_1 \vee \bar{X}_2),$$

$$L^\wedge =$$

$$= \bar{x}_1 \bar{x}_2 (\bar{X}_1 \vee \bar{X}_2) \vee$$

$$\bar{x}_1 x_2 (X_1 \vee \bar{X}_2) \vee x_1 \bar{x}_2 (\bar{X}_1 \vee X_2) \vee x_1 x_2 (X_1 \vee X_2),$$

$$L^\oplus =$$

$$= \bar{x}_1 \bar{x}_2 (X_1 \vee X_2) \vee$$

$$\bar{x}_1 x_2 (X_1 \vee X_2) \vee x_1 \bar{x}_2 (X_1 \vee X_2) \vee x_1 x_2 (X_1 \vee X_2)$$

$$= (X_1 \vee X_2).$$

Тут вхідні змінні x_i з'єднуються між собою знаком кон'юнкції, а похідні – змінні транспортування списків вхідних несправностей, позначені символом X_i , об'єднуються між собою знаком диз'юнкції, відповідно до стану координат стовпців-похідних.

Таким чином, запропонована технологія моделювання несправностей, що заснована на використанні векторних кубітних форм задання функціональностей і похідних, не має аналогів за доступністю розуміння, простотою реалізації та швидкістю. На рис. 6 зображено процесор кубітного моделювання цифрових пристроїв, що включає структури: справного інтерпретативного моделювання, дедуктивного аналізу несправностей, призначеного для оцінки якості тесту і побудови таблиці несправностей, а також модулів тестування і діагностування дефектів на стадіях проектування і експлуатації. Основна відмінність від існуючих рішень полягає у використанні Q-покриття, представленого у формі вектора станів функціональності, що дає можливість істотно підвищити швидкість моделювання за рахунок виконання паралельних реєстрових операцій.

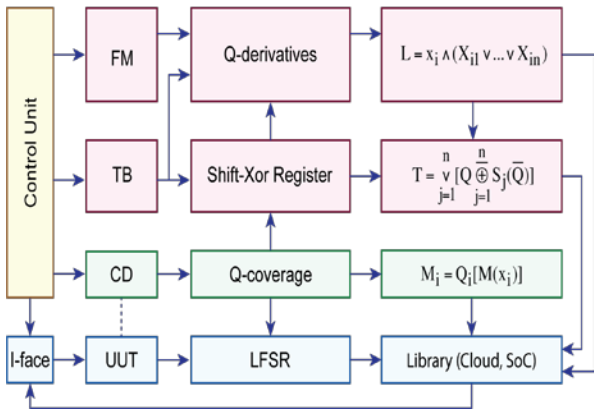


Рис. 6. Процесор кубітного моделювання цифрових пристроїв

Тут: Control Unit – пристрій управління симулятором, який синхронізує роботу блоків справного моделювання та структурних компонентів дедуктивного аналізу несправностей; FM – Fault Matrix, матриця вхідних несправностей розглянутої функціональності цифрового пристрою; TB – Test Bench, упорядкована сукупність вхідних перевіряючих послідовностей, де поточний вхідний набір ідентифікується як x_i ; CD – Circuit Description – схемний опис цифрового пристрою, де функціональні елементи представлені кубітними покриттями Q-coverage. Обробка останніх здійснюється блоком справного моделювання $M_i = Q_i[M(x_i)]$, який реалізує адресні транзакції між кубітним покриттям і вектором моделювання M. Результати справного моделювання вхідних наборів формують матрицю GM (Good simulation Matrix), яка записується в Library. Блок Shift-Xor Register формує матрицю похідних в кубітній формі (Q-derivatives), застосовуючи реєстрові операції зсуву і хог. L-блок формування вихідного списку несправностей використовує формулу аналізу вхідного набору і рядки матриці похідних

$$L = x_i \wedge (X_{i1} \vee \dots \vee X_{in}).$$

Результати дедуктивного аналізу формують списки несправностей, відповідних вхідних наборів, які об'єднуються в DM - Fault Detected Matrix і заносяться до Library. Крім того, T-модуль формує Q-тест і оцінює його якість в матриці одиночних константних несправностей зовнішніх входів і виходів функціональних елементів, які заносяться в Library. Тести для функціональностей разом з матрицями несправностей формують бібліотеку $Library = \{Signature, Q\text{-coverage}, Q\text{-test}, Quality, DM, GM\}$, яка може багаторазово використовуватися як хмарний або вбудований в SoC сервіс для тестування і / або діагностування функціональностей UUT (Unit Under Test) на основі використання інтерфейсу I-face, що підтримує стандарти IEEE 1500 SECT, IP (Internet Protocol). Пошук тестових сервісів у бібліотеці здійснюється за кубітним вектором, попередньо

згорнутим в 16-розрядний двійковий код - сигнатуру (Sign), на основі регістра зсуву з лінійними зворотними зв'язками (LFSR. Це дає можливість структурувати бібліотеку для швидкого отримання інформації та проведення тестування в режимі online.

Структура взаємодіючих компонентів хмарного сервісу QuaSim представлена на рис. 7. Квантове або кубітне уявлення моделі цифрового пристрою разом з інтерпретативним симулятором складають ядро системи, інтегрованої у великі дані Інтернету.

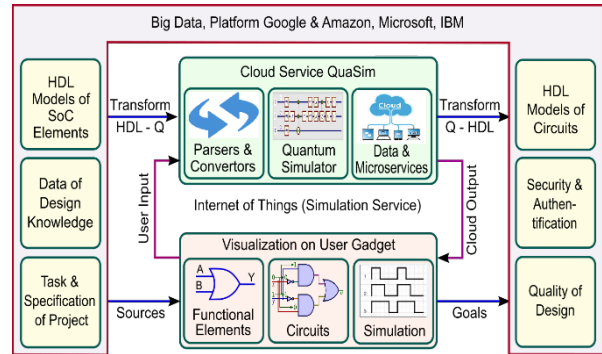


Рис. 7. Хмарний сервіс тестування і моделювання

Це дає можливість використовувати як вихідні дані відкриті специфікації і тестбенчі, описані в мовах VHDL, Verilog. Занурення QuaSim сервісу в інтернет-простір передбачає вивантаження результатів його роботи, пов'язаної з аналізом і синтезом навчальних або ринково орієнтованих проєктів у сервіси зберігання даних на платформах Google, Amazon, Microsoft, IBM, Facebook. Демонстрація хмарного інтерфейсу для тестування і верифікації цифрових схем представлена на рис. 8.

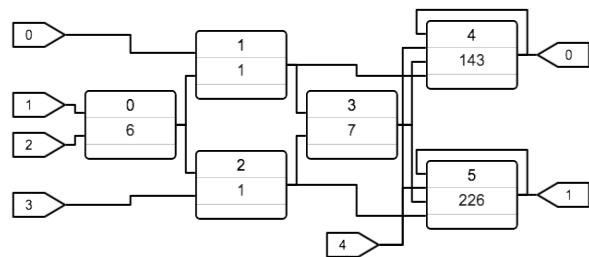


Рис. 8. Скріншот структури з тригерами

Природно, що інтеграція хмарного сервісу з кіберпростором передбачає наявність парсер-мікросервісів для перетворення специфікацій з мов опису апаратури у внутрішню мову QuaSim, а також має існувати і зворотне перетворення даних з кубітного уявлення в стандарти HDL-мов. Парсеризація забезпечує можливість використання відкритих в інтернеті проєктів для їх вивчення і порівняння в системі моделювання Q-sim, а також робить доступними внутрішні проєктні

рішення QuaSim для всіх бажаючих на ринку освітніх сервісів.

5. Висновки

1) Розроблено метод і секвенсор синтезу тестів для функціональних логічних компонентів, який характеризується паралельним виконанням реєстрових логічних операцій (shift, or, not, pxor) над кубітним вектором і його похідними, що дає можливість істотно зменшити час генерування вхідних наборів і тестування пристрою в режимі embedded online.

2) Розроблено метод обчислення похідних для генерації тестів функціональних компонентів, який характеризується паралельним виконанням реєстрових логічних операцій (shift, or, not, pxor) над кубітним вектором, що дозволяє істотно зменшити час генерування вхідних наборів і тестування пристрою за рахунок апаратної надмірності.

3) Розроблено метод моделювання справної поведінки і несправностей для функціональних компонентів, який характеризується паралельним виконанням реєстрових логічних операцій (shift, or, not, pxor) над кубітним вектором і його похідними, що дає змогу істотно зменшити час верифікації і тестування цифрового пристрою в режимі embedded online.

4) Запропоновано процесор кубітного моделювання цифрових пристроїв імплементувати в SoC або Cloud Service для аналізу справної поведінки і несправностей на основі використання кубітних покриттів функціональних елементів, який відрізняється від відомих реалізацій застосуванням мінімального набору реєстрових логічних операцій і високою швидкодією.

5) Практична значущість досліджень полягає в можливості хмарної реалізації швидкодіючого методу синтезу тестів і моделювання несправностей для функціональних логічних компонентів на основі паралельного виконання реєстрових логічних операцій (shift, or, not, pxor) над кубітним вектором і його похідними, що дає можливість генерувати вхідні набори і оцінювати їх якість в режимі online. Крім того, хмарний мікросервіс синтезу тестів і моделювання дефектів для функціональних логічних компонентів може бути затребуваний для навчальних і наукових цілей в процесах синтезу та аналізу цифрових архітектур.

6) Запропонований метод синтезу тестів для функціональностей на основі кубітного покриття може бути використаний як вбудований BIST-компонент для сервісного обслуговування SoC на основі стандарту граничного сканування IEEE 1500 SECT або як хмарний online сервіс тестування апаратних модулів за допомогою IP-протоколу.

7) Подальші дослідження в даній області будуть спрямовані на створення програмно-апаратних генераторів тестів, симуляторів несправностей, справної поведінки, алгоритмів діагностування та бібліотечних рішень, вбудованих в інфраструктуру кристалів і / або хмарні сервіси, що використовують кубітний опис функціональності логічного компонента.

Література: 1. *Рябцев В.Г., Муамар Д.Н.* Метод и средства визуализации алгоритмов тестов диагностирования запоминающих устройств // Электронное моделирование. 2010. Том 32. № 3. С. 43-52. 2. *Zorian Y., Shoukourian S.* Test solutions for nanoscale Systems-on-Chip: Algorithms, methods and test infrastructure // Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers, Yerevan, 2013. P. 1-3. 3. *Tshagharyan G., Harutyunyan G., Shoukourian S. and Zorian Y.* Overview study on fault modeling and test methodology development for FinFET-based memories // 2015 IEEE East-West Design & Test Symposium (EWDTS), Batumi, 2015. P. 1-4. 4. *Проектирование и тестирование цифровых систем на кристаллах / В. И. Хаханов и др.* Харьков: ХНУРЭ, 2009. 484 с. 5. *Abramovici M., Breuer M.A. and Friedman A.D.* Digital systems testing and testable design // Computer Science Press. 1998. 652 p. 6. *Кубитные структуры данных вычислительных устройств / В. И. Хаханов, Ваджеб Гариби, Е. И. Литвинова, А. С. Шкиль* // Электронное моделирование. 2015. Т. 37, № 1. С. 76–99. 7. *Кубитные технологии анализа и диагностирования цифровых устройств / В. И. Хаханов, Тамер Бани Амер, С. В. Чумаченко, Е. И. Литвинова* // Электронное моделирование. 2015. Т. 37, № 3. С. 17–40. 8. *Автоматизированное проектирование цифровых устройств / С.С.Бадулин, Ю.М.Барнаулов и др. / Под ред. С.С. Бадулина.* М.: Радиоисвязь, 1981. 240 с. 9. *Michael A. Nielsen & Isaac L. Chuang.* Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676p. 10. *Mikio Nfifhara.* Quantum Computing. An Overview. Higashi-Osaka: Kinki University, 2010. 53p. 11. *Куроп А.Г.* Курс высшей алгебры. М.: Наука. 1968. 426 с. 12. *Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И.* Структура логического ассоциативного мультипроцессора // Автоматика и телемеханика. 2012. № 10. С. 71-92. 13. *Molnar L. and Gontean A.* Fault simulation methodes, 2016 // 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2016. P. 194-197. 14. *Hadjitheophanous S., Neophytou S. N., Michael M. K.* Scalable parallel fault simulation for shared-memory multiprocessor systems // 2016 IEEE 34th VLSI Test Symposium (VTS), Las Vegas, NV, 2016. P. 1-6. 15. *Pomeranz Irith, Reddy Sudhakar M.* Aliasing Computation Using Fault Simulation with Fault Dropping // IEEE Trans. on Computers. 1995. P. 139-144. 16. *Ubar R., Kõusaar J., Gorev M. and Devadze S.,* Combinational fault simulation in sequential circuits // 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, 2015. P. 2876-2879. 17. *Gorev M., Ubar R. and Devadze S.,* Fault simulation with parallel exact critical path tracing in multiple core environment // 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, 2015. P. 1180-1185. 18. *Pomeranz*

I. Fault simulation with test switching for static test compaction // 2014 IEEE 32nd VLSI Test Symposium (VTS), Napa, CA, 2014. P. 1-6. **19. Mirkhani S. and Abraham J. A.** EAGLE: A regression model for fault coverage estimation using a simulation based metric // 2014 International Test Conference, Seattle, WA, 2014. P. 1-10. **20. Hahanov I., Gharibi W., Iemelianov I., Tamer Bani Amer.** QuaSim – Cloud Service for Digital Circuits Simulation // IEEE 2016 East-West Design & Test Symposium. Yerevan, Armenia. P. 363-370. **21. Литвинова Е.И., Хаханов И.В.** Квантовый компьютер для проектирования цифровых систем // Радиоэлектроника и информатика. 2015. № 4. С. 42-45.

Transliterated bibliography:

1. Ryabcev V.G., Muamar D.N. Metod i sredstva vizualizacii algoritmov testov diagnostirovaniya zapominayushchih ustrojstv // Elektronnoe modelirovanie 2010. Tom 32. № 3. S. 43-52.
2. Zorian Y., Shoukourian S. Test solutions for nanoscale Systems-on-Chip: Algorithms, methods and test infrastructure // Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers, Yerevan, 2013, pp. 1-3. doi: 10.1109/CSITechnol.2013.6710371.
3. Tshagharyan G., Harutyunyan G., Shoukourian S. and Zorian Y. Overview study on fault modeling and test methodology development for FinFET-based memories // 2015 IEEE East-West Design & Test Symposium (EWDTS), Batumi, 2015. P. 1-4. doi: 10.1109/EWDTS.2015.7493149
4. Proektirovanie i testirovanie cifrovyyh sistem na kristallah / V. I. Hahanov i dr. Har'kov: HNURE, 2009. 484 s.
5. Abramovici M., Breuer M.A. and Friedman A.D. Digital systems testing and testable design // Computer Science Press. 1998. 652 p.
6. Kubitnye struktury dannyh vychislitel'nyh ustrojstv / V. I. Hahanov, Vadzheb Garibi, E. I. Lit-vinova, A. S. Shkil' // Elektronnoe modelirovanie. 2015. T. 37, № 1. S. 76–99.
7. Kubitnye tekhnologii analiza i diagnostirovaniya cifrovyyh ustrojstv / V.I. Hahanov, Tamer Bani Amer, S.V. Chumachenko, E.I. Litvinova // Elektronnoe modelirovanie. 2015. T. 37, № 3. S. 17–40.
8. Avtomatizirovannoe proektirovanie cifrovyyh ustrojstv / S.S.Badulin, Yu.M.Barnaulov i dr. / Pod red. S.S. Badulina.M.: Radio i svyaz', 1981. 240 s.
9. Michael A. Nielsen & Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676p.
10. Mikio Nfrfhara. Quantum Computing. An Overview. Higashi-Osaka: Kinki University, 2010. 53p.
11. Kurosh A.G. Kurs vysshej algebrы. Izd-vo: Moskva: Nauka. 1968. 426 s.
12. Bondarenko M.F., Hahanov V.I., Litvinova E.I. Struktura logicheskogo asociativnogo multipro-cessora // Avtomatika i telemekhanika. 2012. № 10. S. 71-92.
13. Molnar L. and Gontean A. Fault simulation methods," 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2016.P. 194-197.
14. Hadjitheophanous S., Neophytou S. N., Michael M. K. Scalable parallel fault simulation for shared-memory multiprocessor systems // 2016 IEEE 34th VLSI Test Symposium (VTS), Las Vegas, NV, 2016.P. 1-6.

15. Pomeranz Irith, Reddy Sudhakar M. Aliasing Computation Using Fault Simulation with Fault Dropping // IEEE Trans. on Computers. 1995. P. 139-144.
16. Ubar R., Kõusaar J., Gorev M. and Devadze S., Combinational fault simulation in sequential circuits // 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, 2015. P. 2876-2879.
17. Gorev M., Ubar R. and Devadze S., Fault simulation with parallel exact critical path tracing in multiple core environment // 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, 2015. P. 1180-1185.
18. Pomeranz I. Fault simulation with test switching for static test compaction // 2014 IEEE 32nd VLSI Test Symposium (VTS), Napa, CA, 2014. P. 1-6.
19. Mirkhani S. and Abraham J. A. EAGLE: A regression model for fault coverage estimation using a simulation based metric // 2014 International Test Conference, Seattle, WA, 2014. P. 1-10.
20. Hahanov I., Gharibi W., Iemelianov I., Tamer Bani Amer. QuaSim – Cloud Service for Digital Circuits Simulation // IEEE 2016 East-West Design & Test Symposium. Yerevan, Armenia. P. 363-370.
21. Litvinova E.I., Hahanov I.V. Kvantovyy komp'yuting dlya proektirovaniya cifrovyyh sistem // Radioelektronika i informatika. 2015. № 4. S. 42-45.

Надійшла до редколегії 11.07.2017

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Литвинова Євгенія Іванівна, д-р техн. наук, проф. кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування та тестування цифрових систем. Хобі: музика. Адреса: Україна, 61166, Харків, пр. Науки, 14, e-mail: litvinova_eugenia@icloud.com.

Смельянов Ігор Валерійович, науковий співробітник кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування та тестування цифрових систем. Хобі: мандри. Адреса: Україна, 61166, Харків, пр. Науки, 14, e-mail: iyemelyanov@itdelight.com.

Хаханов Іван Володимирович, студент кафедри АПОТ факультету КІУ ХНУРЕ. Наукові інтереси: технічна діагностика цифрових систем, програмування. Хобі: горні лижі, англійська мова. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. +3805770-21-326, e-mail: ivanhahanov@icloud.com.

Litvinova Evgenia Ivanovna, Dr. of Tech. Sciences, prof., Design Automation Department, NURE. Scientific interests: design and testing of digital systems. Hobbies: music. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14, e-mail: litvinova_eugenia@icloud.com.

Yemelyanov Igor Valerievich, researcher, Design Automation Department, NURE. Scientific interests: design and testing of digital systems. Hobbies: traveling. Address: Ukraine, 61166, Kharkov, Nauki Ave, 14, e-mail: iyemelyanov@itdelight.com.

Hahanov Ivan Vladimirovich, student, Design Automation Department, NURE. Scientific interests: technical diagnostics of digital systems, programming. Hobby: mountain skiing, English. Address: Ukraine, 61166, Kharkov, Nauki Ave., 14, ph. + 3805770-21-326, e-mail: ivanhahanov@icloud.com.