

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ  
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

**АВТОМАТИЗИРОВАННЫЕ  
СИСТЕМЫ УПРАВЛЕНИЯ И  
ПРИБОРЫ АВТОМАТИКИ**

**Всеукраинский межведомственный  
научно-технический сборник**

**Основан в 1965 г.**

**Выпуск 172**

Харьков  
2015

## СОДЕРЖАНИЕ

<b>МОСКАЛЕНКО В.В., ПИМОНЕНКО С.В.</b> МЕТОД ПРОГНОЗУВАННЯ РІВНЯ СПРИЙНЯТТЯ ЯКОСТІ ОБСЛУГОВУВАННЯ В ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ.....	4
<b>БАРАННИК В.В., КРАСНОРУЦКИЙ А.А., СТАСЕВ С.Ю.</b> КОНЦЕПЦІЯ ФОРМИРОВАНИЯ СТРАТЕГИИ НОРМИРОВАНИЯ С УЧЕТОМ СТРУКТУРНОЙ НАСЫЩЕННОСТИ АЭРОФОТОСНИМКОВ.....	13
<b>ЛЕВЫКИН В.М., ЧАЛАЯ О.В.</b> МОДЕЛЬ РЕЛЯЦИОННОГО НЕЯВНОГО ЗНАНИЯ.....	18
<b>ШКИЛЬ А.С., КУЛАК Э.Н., СЕРОКУРОВА А.С.</b> ДИАГНОСТИРОВАНИЕ HDL-МОДЕЛЕЙ МИКРОПРОГРАММНЫХ АВТОМАТОВ.....	22
<b>ГУРИНА Д.В., ГАРЯЧЕВСКАЯ И.В.</b> РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ БЕСКОНТАКТНОГО ТЕПЛОВОГО КОНТРОЛЯ КАЧЕСТВА РАДИОЭЛЕКТРОННЫХ МОДУЛЕЙ НА ПЕЧАТНОЙ ПЛАТЕ.....	32
<b>ВИШНЯК М.Ю., КЛИМОВА И.Н.</b> УПРАВЛЕНИЕ ЗНАНИЯМИ НА АВТОМАТИЗИРОВАННОМ ПРЕДПРИЯТИИ.....	38
<b>ГВОЗДИНСКИЙ А.Н., ЯКИМОВА Н.А., ГУБИН В.А.</b> О ВОЗМОЖНЫХ ПРЕДСТАВЛЕНИЯХ МАТРИЦ ЛИНЕЙНЫХ ЛОГИЧЕСКИХ ОПЕРАТОРОВ.....	44
<b>ЧАЛАЯ Л.Э., ЧИЖЕВСКИЙ А.В., ВОЛОЩУК Е.Б.</b> МЕТОД ПОИСКА РЕЛЕВАНТНЫХ СВЯЗЕЙ МЕЖДУ КОНЦЕПТАМИ ПРОЕКТИРУЕМЫХ ОНТОЛОГИЙ.....	48
<b>ХАХАНОВ И.В., ЛИТВИНОВА Е.И.</b> СИНТЕЗ И АНАЛИЗ «КВАНТОВЫХ» МОДЕЛЕЙ ЦИФРОВЫХ СИСТЕМ.....	56
<b>БАРАННИК В.В., ЮДИН А.К., ФРОЛОВ О.В.</b> МЕТОД КОСВЕННОГО СТЕГАНОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ФУНКЦИОНАЛА ДЛЯ АДАПТИВНОГО ПОЗИЦИОННОГО КОДИРОВАНИЯ.....	71
<b>БАРАННИК В.В., ШУЛЬГИН С.С.</b> МОДЕЛЬ ОЦЕНКИ ИНФОРМАТИВНОСТИ СЛОТА Р-КАДРОВ НА ОСНОВЕ ВЫЯВЛЕНИЯ СТРУКТУРНО-ГРАДИЕНТНЫХ МЕЖТРАНСФОРМАНТНЫХ ОГРАНИЧЕНИЙ.....	76
<b>БАРАННИК В.В., КОМОЛОВ Д.І.</b> СЕЛЕКТИВНИЙ МЕТОД ШИФРУВАННЯ ВІДЕОПОТОКУ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ ПРИХОВУВАННЯ БАЗОВОГО І-КАДРУ.....	82
<b>РЕФЕРАТИ.....</b>	88

## **МЕТОД ПРОГНОЗУВАННЯ РІВНЯ СПРИЙНЯТТЯ ЯКОСТІ ОБСЛУГОВУВАННЯ В ІНФОРМАЦІЙНО- ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ**

Пропонується в рамках функціонально-вартісного аналізу синтез вирішальних правил для прогнозування рівня сприйняття якості обслуговування клієнтів інформаційно-телекомунікаційної системи. Розглядається застосування узагальненого критерію І.В. Кузьміна, інформаційна складова якого обчислюється в процесі інформаційно-екстремального навчання, а вартісна складова – в процесі підрахунку затрат на формування вхідного математичного опису системи та втрат внаслідок помилок прогнозування.

### **Вступ**

З розвитком інформаційно-телекомунікаційних систем потреби клієнтів і їх поведінка змінилися. Центр уваги зміщується від підвищення продуктивності мережі до покращення сприйняття якості обслуговування (Quality of Experience, QoE). Забезпечення заданого рівня QoE для послуг і додатків стає фундаментальною задачею при реалізації наскрізного керування ресурсами [1,2]. На рівень QoE впливають технологічні, соціальні та економічні фактори. При цьому технологічні фактори пов'язані з якістю обробки зростаючих обсягів трафіка, а соціальні характеризують вид контенту (аудіо, відео, інше), область застосування (освіта, мистецтво, медицина), спосіб надання послуг (streaming, broadcast, файл), напрямок потоків (однонаправлені, двонаправлені, мультинаправлені), місце знаходження (в транспорті, вдома, на вулиці) та пристрій доступу (смартфон, ноутбук, планшет). Економічні фактори пов'язані з бажанням постачальників збільшити свій прибуток, зберегти існуючих та залучити нових клієнтів шляхом розвитку позитивного чи негативного досвіду користувачів відносно тих чи інших послуг через рекламу, програми лояльності та інші методи впливу.

В працях [3,4] показано, що зв'язок QoE з традиційними ключовими метриками продуктивності (Key Performance Indicators) інформаційно-телекомунікаційних систем має нелінійний, нестаціонарний та часто неоднозначний характер. Як було показано в працях [5,6], перспективним підходом для оцінювання QoE є застосування ідей і методів машинного навчання та розпізнавання образів. При цьому врахування контекстних факторів через доступні метадані та результати опитувань клієнтів дозволяє побудувати більш точні моделі прогнозування QoE [7] і реалізувати цінову диференціацію надання послуг. В роботі [8] пропонується перехід від угод про рівень послуг (Service Level Agreements, SLA) до угод про рівень очікуваної якості послуг (Experience Level Agreements, ELA). При цьому перешкодою до широкого впровадження угод ELA є відсутність єдиного погляду на питання формування вхідного математичного опису систем оцінювання QoE для різноманітних інформаційно-телекомунікаційних сервісів та недослідженість проблеми оптимізації пов'язаних з цим витрат і відповідно прибутку. Основна причина такого стану обумовлена невирішеними проблемами науково-методологічного характеру, пов'язаними з незавершеністю формування основ функціонально-вартісного аналізу.

Одним з перспективних шляхів вирішення проблеми інформаційного синтезу систем керування слабоформалізованими процесами в рамках функціонального підходу є застосування ідей і методів інформаційно-екстремальної інтелектуальної технології аналізу даних, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її навчання [9]. Для врахування вартісної складової в [9] запропоновано використовувати узагальнений критерій І. В. Кузьміна.

У даній статті в рамках функціонально-вартісного аналізу розглядається задача синтезу інформаційно-екстремального класифікатора рівнів QoE, здатного адаптуватися до умов і потреб ринку інформаційно-телекомунікаційних послуг.

## 1. Постановка задачі

Нехай дано алфавіт класів розпізнавання  $\{X_m^0 \mid m = \overline{1, M}\}$ , які описують якість сприйняття сервісу інформаційно-телекомунікаційної системи. Як реалізації кожного класу розпізнавання розглядаються впорядковані вектори ознак, до складу яких входять метрики продуктивності та дані контекстних факторів впливу на QoE. Відомі матриця  $\{C_{m,k} \mid m = \overline{1, M}; k = \overline{1, M}\}$ , що містить втрати внаслідок помилкового розпізнавання реалізації класу  $X_m^0$ , як реалізації класу  $X_k^0$  та ціна  $W$  одного опитування щодо рівня якості сервісу. Відомий структурований вектор параметрів функціонування класифікатора:

$$g = \langle n_m, \Sigma, \delta_i, d_m \rangle, \quad (1)$$

де  $n_m$  – кількість векторів-реалізацій у вибірці  $m$ -го класу для потреб навчання та перенавчання;  $\Sigma$  – словник ознак розпізнавання;  $\delta_i$  – параметр, що визначає для  $i$ -ї ознаки ширину поля контрольних допусків;  $d_m$  – кодовий радіус контейнера класу  $X_m^0$ , що відновлюється в радіальному базисі бінарного простору ознак  $\Omega_B$ .

При цьому задано такі обмеження:  $n_m \geq n_{\min}$ , де  $n_{\min}$  – мінімальний репрезентативний обсяг вибірки для навчання (перенавчання);  $d_m < d(x_m \oplus x_c)$ , де  $d(x_m \oplus x_c)$  – кодова відстань між двійковим еталонним (усередненим) вектором класу  $X_m^0$  та двійковим еталонним (усередненим) вектором сусіднього класу  $X_c^0$ ;  $\delta_i \in [0; \delta_{\max}]$ , де  $\delta_{\max}$  – граничне значення параметра поля контрольних допусків.

Необхідно в процесі навчання системи сформуванати навчальну матрицю типу “об’єктивність”  $\{y_{m,i}^{(j)} \mid m = \overline{1, M}; j = \overline{1, n_m}; i = \overline{1, N}\}$ , де  $N$  – потужність повного словника ознак розпізнавання, та визначити оптимальні значення координат вектора параметрів функціонування (1), що забезпечують максимальне значення узагальненого критерію функціональної ефективності (КФЕ):

$$J = \frac{\bar{E}}{E_{\max}} \cdot \frac{C_{\min}}{C_{\text{training}} + C_{\text{error}}}, \quad (2)$$

де  $\bar{E}$  – усереднене за алфавітом класів  $\{X_m^0\}$  значення інформаційного критерію ефективності машинного навчання класифікатора рівнів QoE;  $E_{\max}$  – максимальне граничне значення інформаційного критерію;  $C_{\min}$  – мінімальне граничне значення витрат оператора/провайдера, пов’язаних з експлуатацією системи оцінювання рівня QoE;  $C_{\text{training}}$  – значення затрат на експлуатацію системи, що включає, наприклад, витрати на опитування респондентів і отримання контекстних ознак, вартість системних ресурсів, задіяних при навчанні (перенавчанні) та екзамені;  $C_{\text{error}}$  – втрати оператора/провайдера інформаційно-телекомунікаційної системи, пов’язані зі штрафами та неоптимальним керуванням ІТ-інфраструктурою внаслідок помилкового оцінювання рівня QoE.

Необхідно в режимі екзамену, тобто в робочому режимі системи оцінювання якості інформаційно-телекомунікаційних сервісів, прийняти рішення про належність вектора-реалізації до одного з класів алфавіту  $\{X_m^0\}$ , сформованого на етапі навчання, з метою прогнозування рівня QoE.

## 2. Алгоритм самонавчання функціонування здатного навчатися класифікатора

Кодування навчальної матриці в рамках ІЕІ-технології здійснюється з урахуванням ймовірнісних характеристик як кількісних, так і категоріальних ознак в базовому класі

$X_B^0 \in \{X_m^0\}$  за правилами:

$$x_{m,i}^{(j)} = \begin{cases} 1, & \text{якщо } A_{L,i} \leq y_{m,i}^{(j)} \leq A_{H,i}; \\ 0, & \text{інакше} \end{cases}$$

$$x_{m,C^*i-r}^{(j)} = \begin{cases} 1, & \text{якщо } y_{m,i}^{(j)} = v_r \text{ та } A_{L,i} \leq f_{m,i,r}^{(j)} \leq A_{H,i}; \\ 0, & \text{інакше} \end{cases} \quad r = \overline{1, C},$$

де  $y_{m,i}^{(j)}$  – числове значення  $i$ -ї ознаки в  $j$ -й реалізації  $m$ -го класу розпізнавання;  $A_{L,i}$ ,  $A_{H,i}$  – нижній та верхній пороги системи контрольних допусків;  $C$  – кількість номінальних значень, що може приймати  $i$ -та категоріальна ознака;  $v_r$  –  $r$ -е значення категоріальної ознаки;  $f_{m,i,r}^{(j)}$  – відносна частота появи  $r$ -го значення в  $i$ -й категоріальній ознаці.

Процедура двійкового кодування  $i$ -ї ознаки потребує обчислення верхньої  $A_{H,i}$  та нижньої  $A_{L,i}$  меж поля контрольних допусків, що здійснюється за правилами :

$$A_{B,i} = \bar{y}_{B,i} + \delta_i; \quad A_{H,i} = \bar{y}_{B,i} - \delta_i,$$

де  $\delta_i$  – заданий параметр поля контрольних допусків для  $i$ -ї ознаки розпізнавання;  $\bar{y}_{B,i}$  – середнє вибіркове значення ознаки / частоти появи категоріальної ознаки в базовому класі  $X_B^0 \in \{X_m^0\}$ .

Ітераційна процедура оптимізації впорядкованого вектора параметрів поля контрольних допусків  $\langle \delta_1, \delta_2, \dots, \delta_i, \dots, \delta_{N-1}, \delta_N \rangle$  на значення ознак та вектора ймовірностей  $\langle p_1, p_2, \dots, p_i, \dots, p_{N-1}, p_N \rangle$  включення ознак до робочого словника  $\Sigma^*$  полягає в максимізації узагальненого критерію (2)

$$\Sigma^* = \arg \max_{\Sigma \in \Omega} \{ \max_{G_\delta} \{ \max_{\{k\}} J_k \} \}, \quad (3)$$

де  $G_\delta$  – область допустимих значень поля контрольних допусків;  $\{k\}$  – множина кроків оптимізації геометричних параметрів розбиття.

Як критерій ефективності машинного навчання класифікатора розглядається модифікація інформаційної міри Кульбака [7]:

$$E_m = \left[ 1 - 2[p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}] \right] * \log_2 \left[ \frac{1 - [p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}]}{p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}} \right], \quad (4)$$

тут  $\alpha_m$  – помилка другого роду при розпізнаванні реалізацій класу  $X_m^0$ ;  $\beta_m$  – помилка другого роду;  $p_1, p_2$  – оцінка безумовної ймовірності появи реалізацій класу  $X_m^0$  та найближчого до нього сусіднього класу  $X_c^0$  відповідно.

Робоча (допустима) область визначення функції інформаційного критерію обмежена нерівностями  $D_{1,m} = 1 - \alpha_m \geq 0,5$ ,  $D_{2,m} = 1 - \beta_m \geq 0,5$ .

При обчисленні витрат, пов'язаних з використанням системи оцінювання рівнів QoE, в даному дослідженні запропоновано обмежитися затратами на формування вхідного математичного опису та втратами внаслідок неточної оцінки рівня QoE:

$$C = C_{\text{training}} + C_{\text{error}} = W * \sum_{m=1}^M n_m + \sum_{m=1}^M \sum_{k=1}^M C_{m,k} * n_{m,k},$$

де  $n_{m,k}$  – кількість реалізацій класу  $X_m^0$ , що потрапляють до контейнера класу  $X_k^0$ .

При оцінці втрат враховується, що за відсутності машинного навчання рішення приймається на основі контролю виконання умов контрактів на обслуговування (SLA), а після формування вирішальних правил штрафи за порушення умов SLA знімаються тільки при порушенні умов ELA, тобто при відсутності адекватної реакції на зниження QoE.

Оскільки в задачах багатопараметричної оптимізації окрім точності обчислень набуває важливого значення підвищення оперативності, то для оптимізації в рамках ІЕІ-технології параметрів функціонування класифікатора було обрано роевий алгоритм пошуку косяка риб (Fish School Search, FSS), який на відміну від інших алгоритмів еволюційного програмування, оснований на імітації механізмів природи, відрізняється простотою реалізації, інтерпретабельністю та високою швидкістю збіжності [9].

В алгоритмі FSS кожна риба зберігає одне з рішень задачі. При цьому косяк риб є агрегацією агентів рою, які рухаються приблизно з однією й тією ж швидкістю і орієнтацією, підтримуючи приблизно однакову відстань між собою. Індивідуальний успіх кожної риби в пошуці рішення характеризується її вагою, що відіграє роль пам'яті. Кожна ітерація пошуку виконує дві групи операторів – годування та плавання.

Оператор годування формалізує успішність дослідження агентами тих чи інших областей "акваріуму" і полягає в обчисленні ваги  $z$ -го агента, яка пропорційна нормалізованій різниці значень фітнес-функції на наступній та поточній ітераціях:

$$w_z[k+1] = w_z[k] + \frac{J(P_z[k+1]) - J(P_z[k])}{\max(J(P_z[k+1]), J(P_z[k]))}, \quad z = \overline{1, Z},$$

де  $P_z[k+1]$ ,  $P_z[k]$  – позиція  $z$ -го агента в багатовимірному просторі рішень на  $k$ -й та  $(k+1)$ -й ітерації алгоритму FSS.

Максимально можливе значення ваги агента  $w_z$  у алгоритмі FSS обмежується значенням  $w_{\max} > 0$ . При цьому під час ініціалізації популяції всім агентам присвоюється вага, що рівна  $w_{\max} \cdot 0,5$ .

В алгоритмі FSS розрізняють три види плавання – індивідуальне, інстинктивно-колективне та колективно-вольове. Ці види плавання здійснюються послідовно один за одним в окремі інтервали часу  $(t, \tau], (\tau, \theta], (\theta, t')$ ,  $t < \tau < \theta < t'$ ,  $t' = t + 1$ .

Під час індивідуального плавання агентів відбувається їх переміщення, що має рівномірний випадковий характер. При цьому за одну ітерацію алгоритму FSS крок індивідуального плавання виконується фіксовану кількість разів. Компоненти кроку переміщення  $v_z^{\text{ind}}$  рівномірно розподілені в заданому інтервалі  $v_{\max}^{\text{ind}}$ :

$$v_z^{\text{ind}} = U(0;1)v_{\max}^{\text{ind}}, \quad z = \overline{1, Z},$$

де  $U(0;1)$  – випадкове число із заданого діапазону  $(0;1)$ .

В процесі інстинктивно-колективного плавання на кожного з агентів чинять вплив всі інші агенти популяції і цей вплив пропорційний індивідуальним успіхам агентів. При цьому позиції агентів обчислюються за формулою

$$P_z^\theta = P_z^\tau + \frac{\sum_j v_j^{\text{ind}} (\tau)(J(P_j^\tau) - J(P_j^t))}{\sum_j (J(P_j^\tau) - J(P_j^t))}, \quad z = \overline{1, Z}. \quad (5)$$

Коллективно-вольове плавання полягає у зміщенні всіх агентів у напрямку поточного центру тяжіння популяції за умови збільшення сумарної ваги косяка риб в результаті індивідуального та інстинктивно-колективного плавання. Якщо сумарна вага змешилась, то зміщення відбувається в протилежному напрямку.

Коллективно-вольове плавання виконується за правилами

$$P'_z = P_z^\theta \pm v^{vol} (P_z^\theta - P_c^\theta), z = \overline{1, Z}, \quad (6)$$

де  $P_c^\theta$  – координати центра тяжіння косяка риб, що визначаються за формулою

$$P_c^\theta = \frac{\sum_z w_z^\theta P_z^\theta}{\sum_z w_z^\theta}.$$

У формулі (6) знак плюс використовується за умов  $\sum_z w_z^\theta > \sum_z w_z^{\theta-1}$ , а знак мінус – в протилежному випадку. При цьому розмір кроку переміщення агентів  $v^{vol}$  є випадковою величиною

$$v^{vol} = v_{max}^{vol} U(0;1),$$

де  $v_{max}^{vol}$  – закодоване значення максимально допустимої довжини кроку переміщення при колективно-вольовому плаванні.

Оптимізація затрат на машинне навчання здійснюється в процесі максимізації узагальненого критерію (2). При цьому умовою зупину є зменшення критерію (2) на значення, що перевищує заданий поріг  $\Delta$ . Розглянемо основні кроки алгоритму оптимізації затрат на навчання.

1. Обчислення мінімальних витрат на обслуговування системи оцінювання QoE.
2. Формування навчальних вибірок мінімального обсягу з максимальним словником ознак для заданого алфавіту класів.
3. Ініціалізація лічильник кроків збільшення затрат на машинне навчання:  $k := 0$ .
4. Додаткові опитування респондентів.
5. Запуск алгоритму інформаційно-екстремального навчання (3).
6.  $k := k + 1$ .
7. Якщо  $J[k] > J_{max}[k-1]$ , то  $J_{max}[k] := J[k]$ , інакше –  $J_{max}[k] := J_{max}[k-1]$ .
8. Якщо  $J_{max}[k] - J[k] > \Delta$ , де  $\Delta$  – заданий поріг, то перехід до кроку 9, інакше – до кроку 4.
9. ЗУПИН.

У режимі екзамену рішення про належність вектора-реалізації  $x^{(j)}$  до одного з класів алфавіту  $\{X_m^0\}$  приймається шляхом обчислення геометричної функції належності

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m},$$

де  $d(x_m^* \oplus x^{(j)})$  – кодова відстань від центра  $k$ -го контейнера класу  $X_m^0$  до вектора  $x^{(j)}$ .

Таким чином, алгоритм машинного навчання в рамках функціонально-вартісного аналізу полягає у реалізації ітераційної процедури максимізації узагальненого критерію ефективності системи (2) з метою формування оптимальних в інформаційному та варісному сенсі вхідного математичного опису та відповідних вирішальних правил.

### 3. Результати фізичного моделювання

За даними моніторингу процесу обслуговування мобільних клієнтів сервіс-провайдера мультимедійного контенту було сформовано вибірки спостережень для трьох класів сприйняття якості обслуговування QoE [5,6]. Клас  $X_1^0$  характеризував неприйнятний (незадовільний) рівень якості, а  $X_2^0$  та  $X_3^0$  – прийнятний та відмінний рівні якості обслуговування відповідно. Обсяг вибірки кожного класу становив  $n_{\max} = 400$ , а потужність повного словника ознак –  $N = 25$ . До словника ознак включено ключові та інтегральні показники продуктивності компонентів ІТ-інфраструктури, задіяних у наданні сервісу, та контекстні ознаки, що характеризують тип, популярність та ціну мультимедійного контенту, місце розташування користувача, тип пристрою перегляду, тип мережевого з'єднання, час доби та день тижня.

На рис.1 показано графік оптимізації затрат на машинне навчання класифікатора рівнів QoE за узагальненим критерієм (2). При цьому графік зміни вартісної складової  $K_C = C_{\min} / (C_{\text{training}} + C_{\text{error}})$  узагальненого критерію (2) помічено номером 1, а графік зміни інформаційної складової ефективності класифікатора  $K_I = \bar{E} / E_{\max}$  має номер 2. Навчання починається при обсязі вибірок  $n_{\min} = 100$ . З метою отримання оптимальних в інформаційному та вартісному сенсі вирішальних правил навчальні вибірки доповнюються новими класифікованими векторами, що збільшує затрати, але може підвищити інформаційну складову ефективності класифікатора.

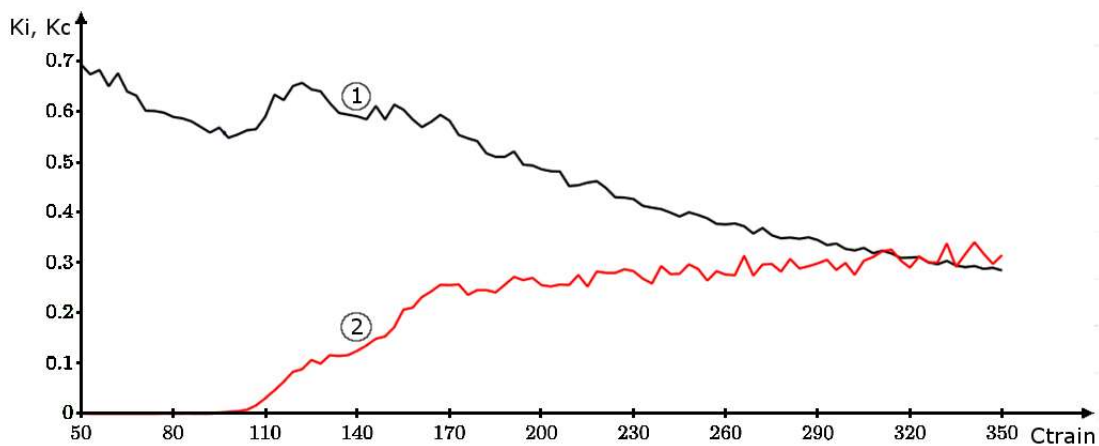


Рис.1. Залежність інформаційної та вартісної складової узагальненого критерію від затрат на машинне навчання: 1 – вартісна складова; 2 – інформаційна складова

Аналіз рис.1 показує, що підвищення затрат на машинне навчання дозволяє дещо підвищити достовірність вирішальних правил і відповідно зменшити втрати внаслідок невиправданого очікування клієнтів щодо якості сервісу чи простою невикористаних ресурсів. При цьому починаючи з деякого рівня затрат, вони стають невиправданими, оскільки істотно не підвищують достовірність вирішальних правил. На рис.2 показано графік зміни узагальненого критерію (2) від затрат на машинне навчання.

Графік, наведений на рис.2, побудований на повному наборі даних з метою ілюстрації закономірності, яка полягає в поступовому наростанні значення критерію та наступному його спаданні. В практичних (не лабораторних) умовах процес розширення навчаної матриці припиниться раніше. Момент зупину визначається заданим порогом  $\Delta$ , що визначає максимальне зменшення критерію (2) порівняно з глобальним максимумом.

Аналіз рис.2 показує, що починаючи з відмітки  $C_{\text{train}} = 165$  значення критерію (2) має тенденцію до спадання і в одному з локальних мінімумів буде прийнято рішення про зупин процедури навчання і перехід в робочий режим оцінювання рівня QoE.

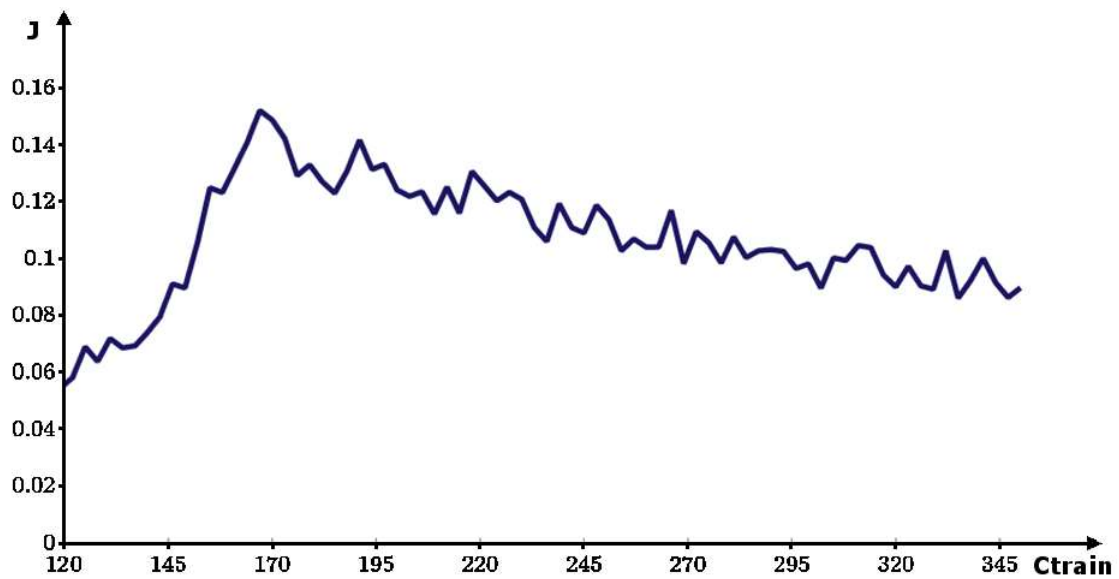


Рис. 2. Графік залежності узагальненого критерію (2) від затрат на навчання класифікатора

На рис. 3 показано графік зміни максимумів усередненого за алфавітом класів інформаційного критерію  $K_I$  в процесі роевої оптимізації косяком риб багатовимірного вектора параметрів, що включає в себе ймовірності входження ознак розпізнавання до робочого словника та значення параметра поля контрольних допусків для кожної ознаки розпізнавання, при оптимальному обсязі навчальної матриці.

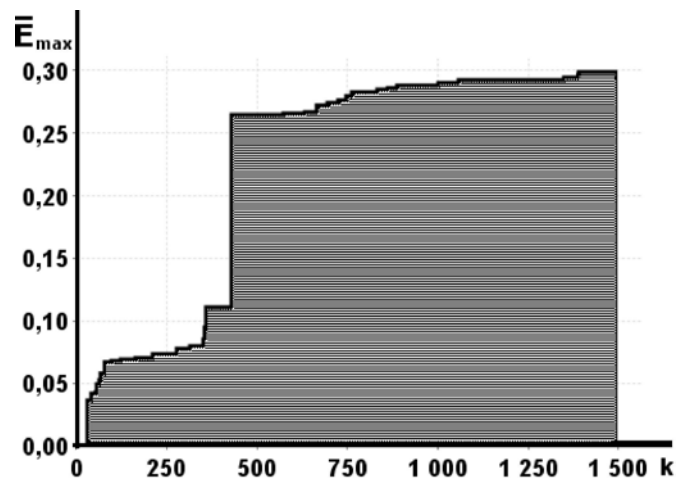


Рис. 3. Залежність КФЕ від кількості переміщень агентів роевого пошуку

Аналіз рис.3 показує, що в результаті 1500 ітерацій алгоритму пошуку було знайдено оптимальні в інформаційному сенсі словник ознак та параметри полів контрольних допусків на значення ознак розпізнавання. При цьому максимальна величина усередненого нормованого інформаційного критерію не досягає граничного значення. На рис.4 показано результати оптимізації геометричних параметрів розбиття на класи еквівалентності.

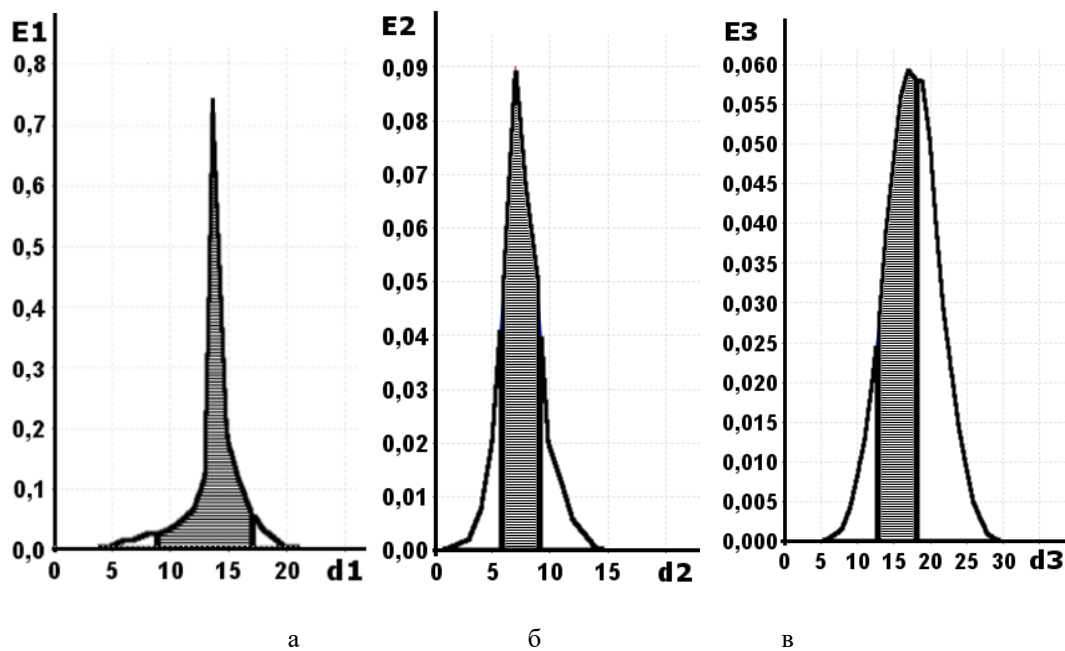


Рис. 4. Графіки залежності інформаційного критерію від радіуса контейнера класу: а – клас  $X_1^0$ ; б – клас  $X_2^0$ ; в – клас  $X_3^0$

Аналіз рис. 4 показує, що оптимальні значення радіусів гіперсферичних контейнерів класів  $X_1^0$ ,  $X_2^0$  та  $X_3^0$  відповідно рівні:  $d_1^* = 14$ ,  $d_2^* = 7$  та  $d_3^* = 17$  (в кодових одиницях). При цьому значення інформаційного критерію для класів  $X_1^0$ ,  $X_2^0$  та  $X_3^0$  становлять  $E_1^* = 0,9$ ,  $E_2^* = 0,06$  та  $E_3^* = 0,075$ , що відповідає таким відповідним значенням ймовірності правильного розпізнавання  $P_{\text{true},1} = 0,99$ ,  $P_{\text{true},2} = 0,8$ ,  $P_{\text{true},3} = 0,75$ .

Таким чином, отримані в процесі оптимізації параметрів функціонування (1) вирішальні правила для прогнозування рівня QoE є оптимальними в інформаційному та вартісному сенсі.

### Висновки

В рамках функціонально-вартісного аналізу здійснено синтез інформаційно-екстремального класифікатора рівнів QoE, що враховує як інтегральні показники продуктивності компонентів ІТ-інфраструктури, так і контекстні ознаки. При цьому не вдалося отримати безпомилкові за навчальною матрицею вирішальні правила, що пов'язано із суб'єктивною природою даних опитування та недостатньою інформативністю словника ознак. Тому дослідження буде продовжено в напрямку розширення словника інформативними ознаками.

**Список літератури:** 1. Skorin-Kapov L. A Multi-Dimensional View of QoE: the ARCU Model / L. Skorin-Kapov, M. Varela // Proceedings of the 35th International Convention MIPRO. 2012. Optija, Croatia : IEEE Press. P. 662 – 666. 2. Balachandran A. Developing a Predictive Model of Quality of Experience for Internet Video / A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, H. Zhang // Proceedings of the ACM SIGCOMM conference. 2013. Vol. 43, I. 4. New York, NY, USA : IEEE Press. P. 339-350. 3. Mateo P. A Context-aware Model for the Analysis of User Interaction and QoE in Mobile Environments / P. Mateo, D. S. Ruiz, G. M. Perez // International Journal of Human-Computer Interaction. 2014. Vol. 30, I. 12. Norwood, N.J, USA : Ablex Pub. P. 946-964. 4. Al-Shammari S. Defining a Metric for Measuring QoE of SaaS Cloud Computing / S. Al-Shammari, A. Al-Yasiri // Proceedings of the PGNET 2014 Conference. Liverpool, United Kingdom : IEEE Press. 2014. P. 251-256. 5. Bouten N. QoE Optimization Through In-Network Quality Adaptation for HTTP Adaptive Streaming / N. Bouten, J. Famaey, S. Latre, R. Huysegems, B. D. Vleeschauwer, W. Van Leekwijck, F. D. Turck // Proceedings of the 8th International Conference on Network and Service

Management. 2012. Las Vegas, USA : IEEE Press, Research Publishing Services. P. 336-342. **6. Battilotti S.** Approaches for Future Internet architecture design and Quality of Experience (QoE) Control / S. Battilotti, F. D. Priscoli, C. G. Giorgi, A. Pietrabissa, S. Monaco, M. Panfili, S. Canale, V. Surac // WSEAS transaction on Communications. 2015. Vol. 14. Wisconsin, USA : World Scientific and Engineering Academy and Society. P. 62-73. **7. Rojas-Mendizabala V. A.** Toward Total Quality of Experience: A QoE Model in a Communication Ecosystem / V. A. Rojas-Mendizabala, A. Serrano-Santoyo, R. Conte-Galvana, A. Gomez-Gonzalez // Conference on ENTERprise Information Systems. 2013. Vol. 50, I. 4. USA, NJ: IEEE Press. P. 58-65. **8. Varela M.** From Service Level Agreements (SLA) to Experience Level Agreements (ELA): The challenges of selling QoE to the user / M. Varela, P. Zwickl, M. Xie, H. Schulzrinne, P. Reichl // Communication Workshop (ICCW), 2015 IEEE International Conference on 8-12 June 2015. 2015. London, UK : IEEE Press. P. 1741-1746. **9. Moskalenko V.V.** Intelligent Decision Support System for Medical Radioisotope Diagnostics with Gamma-camera / A.S. Dovbysh, V.V. Moskalenko, A.S. Rizhova, O.V. Dyomin // Journal of Nano- and Electronic Physics. Sumy, Ukraine : Sumy State University. 2015. Vol.7, No 4. P. 04036-1–04036-7.

*Надійшла до редколегії 01.08.2015*

**Москаленко В'ячеслав Васильович**, старший викладач кафедри комп'ютерних наук Сумського державного університету. Наукові інтереси: інтелектуальні системи керування слабо-формалізованими процесами, машинне навчання. Адреса: Україна, 40035, Суми, вул. Сироватська, 66, кв. 84, м.т. +380664291318, e-mail: systemscoders@gmail.com.

**Пімоненко Сергій Володимирович**, директор телекомунікаційної компанії PSTS, аспірант кафедри комп'ютерних наук Сумського державного університету. Наукові інтереси: інформаційно-телекомунікаційні системи, машинне навчання. Адреса: Україна, 40035, Суми, вул. 40 Років Жовтня, 43а/92, м.т. +380990777791, e-mail: pstsnet@gmail.com.

## КОНЦЕПЦИЯ ФОРМИРОВАНИЯ СТРАТЕГИИ НОРМИРОВАНИЯ С УЧЕТОМ СТРУКТУРНОЙ НАСЫЩЕННОСТИ АЭРОФОТОСНИМКОВ

Рассматриваются существующие технологии обработки изображений в современных информационно-телекоммуникационных системах. Указывается на противоречие между временными затратами на доставку данных изображения и поддержание соответствия восстановленных фрагментов исходному аэрофотоснимку. Описываются характеристики компонент трансформанты для участков изображения с различной структурной сложностью. Обосновывается применение разделительной стратегии нормирования с учетом структурной сложности аэрофотоснимков. Создается технология адаптивной нормализации компонент сегмента изображения. Определяется порядок задания коэффициентов нормирования для сегментов изображения со средней степенью насыщенности.

### 1. Введение

Стремительное развитие информационных технологий привело к использованию однонаправленной передачи видеoinформационного ресурса в системах наблюдения. В результате эти системы приобрели широкое распространение в государственных ведомственных учреждениях, особенно в Министерстве обороны Украины. Это способствует улучшению качества системы управления при использовании видеoinформационного ресурса [1]. Результатом проведения аэромониторинга является получение видовых изображений. Под видовым будем понимать изображение местности или аэрофотоснимок (цифровое изображение), полученное в результате аэрофотографирования. В силу важности содержания полученного изображения возникает необходимость максимально снизить временные затраты на обработку и доведение данных аэрофотоснимка в условиях поддержки необходимого уровня соответствия восстановленного изображения исходному.

Для обоснования варианта решения поставленной задачи необходимо рассмотреть существующие технологии обработки аэрофотоснимков. Данные технологии (JPEG, JPEG2000) используют сжатие на базе статистического кодирования (кодов переменной длины) [2]. В процессе обработки аэрофотоснимков в разнообразных специализированных устройствах применяют математическое описание изображения как двумерного сигнала, который характеризуется объемом информации [3]. При кодировании изображения устраняется психовизуальная избыточность с применением предварительного трансформирования и последующего нормирования. Схема базовой технологии показана на рис. 1.

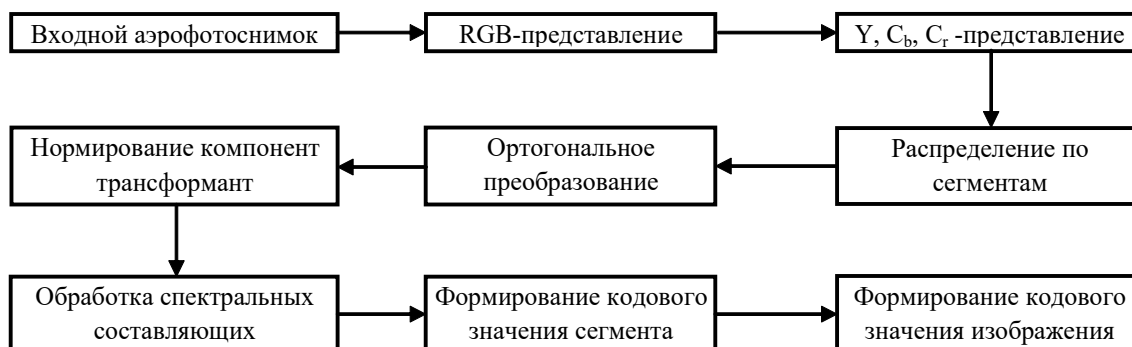


Рис. 1. Схема базовой технологии эффективного кодирования с применением предварительного трансформирования и последующего нормирования

При выполнении базовой технологии начальное значение информационной интенсивности потока выходных данных формируется после этапа нормирования. Эффективность базовой технологии выражается в снижении информационной интенсивности, что физически приводит к уменьшению временных затрат на доставку данных аэрофотоснимка. В то же время возникает проблема качества восстановленного изображения при больших коэффициентах нормирования.

Поэтому цель работы заключается в создании разделительной стратегии нормирования с учетом структурной насыщенности аэрофотоснимков.

## 2. Основной материал

Для создания данной стратегии необходимо более подробно рассмотреть процесс нормирования. При цифровой обработке аналогового сигнала происходит изменение представления широкого (и непрерывного) диапазона входных значений в ограниченный набор дискретных выходных значений. Термин “нормирование” в случае применения при обработке изображений подразумевает округление вещественных чисел до целых или преобразование целых чисел в меньшие целые [4].

Особенностью человеческого зрения является тот факт, что в результате слабой чувствительности к изменению высокочастотных составляющих изображения предоставляется возможность применять к коэффициентам, отвечающим за высокие частоты, более грубый шаг квантования. Нормирование происходит в результате деления матрицы коэффициентов  $x$  на так называемую матрицу нормирования  $W$ , т.е.  $X_{ij} = \lfloor x_{ij}/w_{ij} \rfloor$ , где  $x_{ij}$  – значение компоненты трансформанты;  $w_{ij}$  – значение компоненты матрицы нормирования;  $\lfloor x \rfloor$  – операция определения целой части  $x$ .

Как правило, матрицы нормирования отличаются для компонент цветности и яркости [5]. В большинстве алгоритмов обработки изображения нормирование заключается в обычном поэлементном делении значений компонент некой рабочей матрицы на матрицу нормирования. Для каждой компоненты цветоразностной модели ( $Y$ ,  $U$  и  $V$ ) задается своя матрица нормирования  $W[\alpha, \beta]$ . Нормирование задается следующей формулой:

$$x_{i,j}^{(Q)} = \left\lfloor \frac{x_{i,j}}{w_{i,j}} \right\rfloor. \quad (1)$$

На данном этапе путем изменения степени сжатия задаются амплитуды выходных значений компонент трансформанты и, следовательно, определяется уровень потерь. При использовании матрицы нормирования с большими значениями коэффициентов получается большее количество нулей и, следовательно, и большее значение степени сжатия. Из этого выходит, что при формировании матриц нормирования на этапе задания величины значений их элементов появляется возможность управлять степенью сжатия. В результате при более высоком значении элементов матрицы нормирования меньше будет диапазон значений элементов нормированной трансформанты  $X^{(Q)}[\alpha, \beta]$ , что позволит кодировать с применением меньшего количества информации. К примеру, в стандарте JPEG используются матрицы нормирования, полученные эмпирическим путем. Две рекомендуемые в стандарте JPEG данные по нормированию для яркостной составляющей представлены в таблице.

i \ j	1	2	3	4	5	6	7	8
1	16	11	10	16	24	40	51	61
2	12	12	14	19	26	58	60	55
3	14	13	16	24	40	57	69	56
4	14	17	22	29	51	87	80	62
5	18	22	37	56	58	109	103	77
6	24	35	55	64	81	104	113	92
7	49	64	78	87	103	121	120	101
8	72	92	95	98	112	100	103	99

Значения элементов данных матриц для стандарта JPEG получены опытным путем в результате оценок визуального восприятия [6].

Для изменения значения коэффициента сжатия производится умножение исходных матриц на некоторое число  $Q$ .

Стандарт JPEG2000 подразумевает применение собственных таблиц нормирования, однако при этом их требуется передавать декодеру вместе со сжатыми данными, что приводит к увеличению общего размер закодированных данных.

В качестве альтернативы возможно создание собственных матриц нормирования (поточковой обработки) с применением одного параметра  $R$ , который определяется пользователем. Значение элементов матрицы нормирования определяется следующей формулой:

$$w_{i,j} = 1 + (i + j) \times R. \quad (2)$$

Для заданного выражения характерно уменьшение значений компоненты трансформанты при условном перемещении из левого верхнего угла в правый нижний. При правильном выполнении нормирования полученная выходная трансформанта будет содержать всего несколько ненулевых коэффициентов с характерным расположением в левом верхнем углу матрицы. В дальнейшем согласно алгоритму JPEG к этим полученным числам будет применяться кодирование.

В результате применения нормирования алгоритму будут присущи некоторые специфические эффекты при больших значениях коэффициента сжатия. Потери в области низкочастотных компонент приводят к появлению в полученном изображении квадратов  $8 \times 8$ . Потери высокочастотных компонент вызывают так называемый “эффект Гиббса”, когда вокруг элементов аэрофотоснимка с резким перепадом структурной насыщенности появляется некий ореол.

Алгоритмы JPEG и JPEG2000 применяются для сжатия изображения с последующим его визуальным восприятием, и полученное сжатое изображение не пригодно к дальнейшей цифровой обработке.

В результате приходим к выводу, что нормирование применяется для: экономии памяти; изменения характеристик последовательностей применительно к последующей компрессии; регулировки степени сжатия и соответственно уровня потерь; подготовки к последующей обработке; добавления эффектов. Проанализируем изложенное выше касательно выполнения сжатия аэрофотоснимка. Изменение характеристик последовательностей применительно к последующей компрессии возникает в результате уменьшения количества возможных вариантов значений  $i$ , как следствие, увеличения частоты повторений. На данный момент нормирование – основополагающая часть многих алгоритмов кодирования с потерей информации со значением коэффициента сжатия от 3:1 до 100:1.

Сущность современных методов сжатия заключается в разбиении изображений на области определенной величины. Обычно это квадраты с размерами  $8 \times 8$ ,  $16 \times 16$ . При этом в одну область попадают неоднородные участки изображения, в результате чего не учитывается структурная сложность участков аэрофотоснимка. С одной стороны, это приводит к уменьшению временных затрат на обработку на этапе сжатия, а с другой – вызывает частичную (необратимую) потерю информации. Поэтому возникает необходимость в разработке методики обработки аэрофотоснимка с учетом его структурной сложности.

Для этого предлагается применить стратегию определения параметров обработки с учетом структурной сложности фрагмента аэрофотоснимка.

Структурно-функциональная схема определения параметров нормирования при обработке аэрофотоснимка в зависимости от структурной сложности фрагментов изображений представлена на рис. 2.

В изложенной схеме требуется определить характеристику обработки в соответствии со структурной сложностью фрагмента аэрофотоснимка.

Для этого при обработке аэрофотоснимка (его фрагментов) предлагается управлять величиной матрицы нормирования при осуществлении ортогонального преобразования с последующим кодированием. В результате этого предоставляется возможность производить быструю обработку аэрофотоснимка с учетом его особенностей и локальных связей между фрагментами.

На основании физических свойств фрагмента аэрофотоснимка, который соответствует классу “слабонасыщенный”, вся энергия сконцентрирована в низкочастотной области, значения для высокочастотной части близки к нулю. Для фрагмента же изображения с классом “сильнонасыщенный” вся энергия, соответственно, расположена в высокочастотной области, значения для низкочастотной части стремятся к нулю, что приводит к уменьшению комбинаторной избыточности в трансформантах.



Рис. 2. Структурно-функциональная схема алгоритма определения параметров нормирования

Для фрагментов изображений с классом “средненасыщенные” характерны промежуточные значения элементов трансформанты. В результате этого предлагается осуществлять следующую стратегию определения коэффициентов матрицы нормирования  $W$ : для фрагментов аэрофотоснимков, которые соответствуют классу “слабонасыщенный”, задавать большие значения коэффициентов  $w_{i,j}$ ; для фрагментов аэрофотоснимков, которые соответствуют классу “сильнонасыщенный”, задавать малые значения коэффициентов  $w_{i,j}$ ; для фрагментов аэрофотоснимков, которые соответствуют классу “средненасыщенный”, значения коэффициентов  $w_{i,j}$  определять в зависимости от значений динамических диапазонов для каждой трансформанты.

Значения коэффициентов  $w_{i,j}$  матриц нормирования  $W$  определяются опытным путем, опираясь на полученные в ходе эксперимента данные для типов изображений с различными значениями структурной сложности (как и для стандарта JPEG2000 необходимо создать матрицы). В результате созданной технологии предоставляется возможность управлять параметрами нормирования значений компонент трансформанты на основании параметров структурной сложности фрагмента изображения. Это приводит к уменьшению временных затрат на доведение слабонасыщенных фрагментов изображения, а для сильнонасыщенных увеличивается степень соответствия восстановленного изображения исходному.

### 3. Выводы

Разработана стратегия управления временными затратами на доведение данных аэрофотоснимка путем применения адаптивной нормализации компонент сегмента изображения. Адаптивная нормализация заключается в определении значений коэффициентов нормирования в зависимости от значений динамических диапазонов для каждой трансформанты с учетом насыщенности сегмента изображения. Построенная технология позволяет снизить временные затраты на доставку данных слабонасыщенных участков аэрофотоснимка путем увеличения значений матрицы нормирования. Созданная технология учитывает структурную сложность участков изображения, что увеличивает соответствие восстановленного аэрофотоснимка исходному.

**Список литературы:** 1. Баранник В.В., Підлісний С.А. Аналіз дії кібератак на відео інформаційний ресурс в інформаційно-телекомунікаційних мережах // АСУ та прилади автоматіки. 2014. Вип. 164. С. 16-22. 2. Gonzales R.C. and Woods R.E. "Digital image processing" in Prentice Hall, edition. II, 2002. 1072p. 3. Сэломон. Д. Сжатие данных, изображений и звука. М.: Техносфера, 2004. 368 с. 4. Красильников Н.Н. Цифровая обработка изображений. М.: Вузовская книга, 2011. 320 с. 5. Прэмм У. Цифровая обработка изображений: в 2 т., пер. с англ. М.: Мир, 1985. 736 с. 6. William B. Pennebaker, Joan L. Mitchell. JPEG: Still Image Data Compression Digital Multimedia Standards Series. Springer Science & Business Media, 1992. 638 p.

*Поступила в редколлегию 11.09.2015*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры Харьковского университета Воздушных Сил им. И. Кожедуба. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

**Красноруцкий Андрей Александрович**, канд. техн. наук, старший научный сотрудник, докторант Харьковского университета Воздушных Сил им. И. Кожедуба. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

**Стасев Сергей Юрьевич**, аспирант ХНУРЭ. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

---

## МОДЕЛЬ РЕЛЯЦИОННОГО НЕЯВНОГО ЗНАНИЯ

---

Рассматривается проблема моделирования неявного знания. Описываются особенности реляционного неявного знания и выполняется его структуризация на скрытые и очевидные неформализованные зависимости между сущностями; скрытые зависимости между различными моделями предметной области; скрытые правила использования реляционных знаний. Предлагается модель реляционных неявных знаний, которая включает допустимый набор состояний для сущностей заданной предметной области, неявные отношения переходов между этими состояниями, интерпретацию взаимосвязей между состояниями, а также ограничения на использование неявных знаний.

### Введение

В настоящее время в области менеджмента знаний большое внимание уделяется выявлению и использованию скрытых знаний. Как показано в работах [1-3], знания людей подразделяются на общедоступные знания в документированной форме и знания, принадлежащие отдельным индивидуумам. Последние отражают опыт, практические навыки человека, его понимание взаимоотношений между людьми, а также зависимостей между предметами и явлениями окружающего мира. Иными словами, человеческие знания подразделяются на формализованные явные и неформализованные неявные. Неявное знание (tacit knowledge) обычно передается людьми при непосредственном контакте. Это означает, что такое знание имеет иную форму, отличную от символьной формы явного знания.

В работах [3-8] выделены общие свойства неявных знаний, а также их влияние на управление предприятием. Основной акцент в указанных работах уделен выявлению отношений между людьми при обучении, сокрытии, использовании неявных знаний. В работе [9] скрытые взаимосвязи между людьми предлагается рассматривать как реляционные неявные знания. Однако вопросы структуризации и формализации таких знаний применительно к произвольным сущностям предметной области не рассматриваются.

Следовательно, проблематика моделирования неявных знаний в целом и реляционных неявных знаний в частности является актуальной.

### Постановка задачи

Для использования неявного знания в системах искусственного интеллекта необходимо выделить виды такого знания, формализовать их свойства и выполнить преобразование знания в традиционную символьную форму с учетом особенностей каждого вида неявных знаний.

*Целью* данного исследования является построение модели реляционного неявного знания, отражающего неформализованные взаимосвязи между сущностями предметной области, а также правила интерпретации таких взаимосвязей и ограничения на их использование.

### Структуризация и моделирование реляционного неявного знания.

Опираясь на проведенное в рассмотренных ранее работах описание неявных знаний, определим реляционное неявное знание как такое, которое содержит неформализованные причинно-следственные зависимости между сущностями предметной области. Отметим, что при формализации данного знания сущности в предметной области описываются через набор их элементов, а также взаимосвязей между этими элементами. При таком информационном описании под сущностью можно понимать любой объект предметной области, в том числе объект, изменяющий свое состояние. Например, исполнители в данной области также могут рассматриваться в качестве сущностей. Рассматриваемое реляционное знание является неявным в силу сложности с его формализацией. Последние определяются либо зависимостями, либо затрудненным доступом к носителю знания [9].

Предлагаемая структуризация реляционных неявных знаний опирается на выделение базовых неявных связей между сущностями предметной области и последующее обобщение этих зависимостей:

- скрытые зависимости между сущностями;
- очевидные неформализованные зависимости между сущностями;
- скрытые зависимости между различными моделями предметной области;
- скрытые правила использования реляционных знаний.

Основным признаком неявного знания является невозможность его документирования традиционными методами, поскольку это знание представляется в форме, отличной от традиционной символической.

Скрытые зависимости между сущностями характеризуются следующими особенностями. Процессы передачи знания реализованы напрямую между носителем и потребителем. Тогда передаваемые и используемые зависимости будут скрыты. Поэтому отпадает необходимость в его промежуточном символическом оформлении. Поскольку при передаче скрытых зависимостей нет необходимости в документировании, то такое знание должно передаваться максимально эффективно, в сжатой (возможно, многомерной) форме и, следовательно, должно иметь небольшой объем. Это означает, что и у источника и у приемника существуют дополнительные априорные знания, которые позволяют «расшифровать» скрытые зависимости. Следовательно, для экстернализации (преобразования в явные знания) [3] данного вида неявного знания необходимо выполнить детализацию передаваемых зависимостей на основе использования документированных априорных знаний.

Второй вид неявного реляционного знания – очевидные неформализованные зависимости. Такие зависимости могут быть получены из свойств объектов предметной области, однако они не представлены в символической форме. Трудности их формализации могут быть вызваны различными причинами, например сложностью исходных связей, частичным использованием зависимостей на практике и т.п. В целом данные зависимости определяются известными свойствами сущностей предметной области, поэтому они могут быть экстернализованы с помощью известных методов data, process, web – mining.

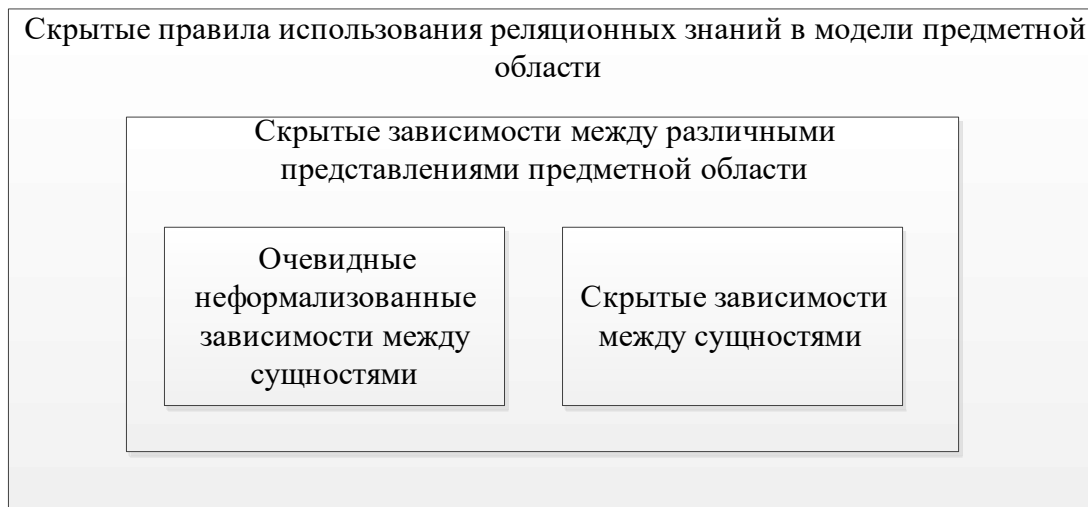
Неявные зависимости между различными моделями предметной области не позволяют полностью документировать знание о подмножестве взаимодействующих сущностей. Действительно, модель предметной области должна содержать явные темпоральные знания о свойствах сущностей, текущих значениях этих свойств, а также причинно-следственных связях между сущностями. Отметим, что такие связи изменяются в зависимости от значений свойств сущностей.

Для обеспечения полноты и целостности знаний при их передаче между различными моделями предметной области необходимо иметь полностью явное представление знаний об источнике у получателя. Поскольку в общем случае модели предметной области могут содержать и явные, и неявные зависимости между одними и теми же сущностями предметной области, то транзитивно получаем, что между моделями существуют скрытые зависимости.

Из изложенного видно, что необходимым условием экстернализации отношений между моделями предметной области является формализация скрытых и очевидных неформализованных зависимостей между сущностями.

Последняя разновидность рассматриваемых знаний – скрытые правила использования неявных реляционных знаний. Эти правила представляют собой метазнания, которые определяют приоритеты при выборе используемой модели предметной области. Пример использования таких знаний – два аналогичных интерактивных процесса сервисного обслуживания транснациональной фирмы, выполняющиеся в различных странах. Процессы отличаются временными затратами на выполнение. Различная эффективность исполнения таких процессов зависит от имеющихся приоритетов по выбору наиболее важных в данный момент последовательностей действий. Эти приоритеты определяются скрытыми правилами использования знаний в рамках модели предметной области. Иными словами, выбор модели предметной области на основе неявных правил определяет выбор последовательности действий процесса.

Выполненная структуризация неявных реляционных знаний позволяет построить много-  
 слойную обобщенную модель таких знаний. Иерархия реляционных неявных знаний приве-  
 дена на рисунке.



Используя представленную структуризацию неявных реляционных знаний  $K^R$  и учиты-  
 вая приведенные особенности их составляющих, определим логическую модель таких  
 знаний. Обозначим через  $M_{EC}^R$  модель предметной области, содержащую только неявную  
 реляционную компоненту знаний нижнего уровня:

$$M_{EC}^R = (K_{EC}^R, O), O = \{o_i\}, o_i = \{a_j\}, a_j = \{(v_{jk}, t_{jk})\}, \quad (1)$$

где  $K_{EC}^R = K_E^R \wedge K_C^R$  – дизъюнкция очевидных  $K_E^R$  и скрытых  $K_C^R$  реляционных неявных  
 знаний;  $O$  – набор сущностей предметной области;  $\{a_j\}$  – набор атрибутов, позволяющих  
 полностью описать сущность в модели;  $\{(v_{jk}, t_{jk})\}$  – набор значений  $v_{jk}$   $k$  – значения  $j$  –  
 атрибута сущности  $o_i$ , которое зафиксировано в момент времени  $t$ .

Базовые элементы модели – очевидные  $K_C^R$  (2) и скрытые  $K_C^O$  (3) неформализованные  
 зависимости:

$$K_E^R : O_l \rightarrow O_m, O_l, O_m \subseteq O \mid \forall o_i \in O_l, O_m \quad \{a_j\} \neq \emptyset \wedge a_j \neq \emptyset; \quad (2)$$

$$K_C^R : O_l \rightarrow O_m, O_l, O_m \subseteq O \mid \exists o_i \in O_l, O_m \wedge (\{a_j\} = \emptyset \vee a_j = \emptyset). \quad (3)$$

Для очевидных зависимостей известны атрибуты сущностей, которые они используют.  
 Однако, как было показано ранее, зависимости не формализованы по различным причинам  
 и потому представлены в форме, которая не может быть использована в системах ИИ.

Для скрытых зависимостей неизвестны либо атрибуты сущностей ( $\{a_j\} = \emptyset$ ), либо хотя  
 бы отдельные значения этих атрибутов.

Зависимости между моделями предметной области  $M_l^R$  и  $M_m^R$  определяются в соответ-  
 ствии с представленной на рисунке структурой только при наличии очевидных и скрытых  
 реляционных знаний:

$$K_M^R : M_{EC,l}^R \rightarrow M_{EC,m}^R \mid \exists K_C^R, K_E^R \in M_{EC,l}^R, M_{EC,m}^R. \quad (4)$$

Отметим, что зависимость (4) показывает, что неявные знания  $K_M^R$  могут быть получе-  
 ны путем сравнения, например, моделей двух аналогичных дискретных процессов, выпол-  
 няющихся в предметной области.

Аналогично, правила использования реляционных знаний оперируют остальными видами реляционных знаний, позволяя определить выбор той или иной модели.

$$K_R^R : K_{EC}^R, K_M^R \rightarrow M_{EC}^R . \quad (5)$$

Таким образом, выражение (1) определяет набор состояний для заданной предметной области, выражения (2) и (3) – базовые неявные отношения между этими состояниями, выражение (4) – интерпретацию взаимосвязей между состояниями, а (5) – правила выбора интерпретации. В такой трактовке мы получаем модель реляционных неявных знаний на основе структур Крипке [10]:

$$M^R = (\{(v_{jk}, t_{jk})\}, K_{EC}^R, K_M^R | K_R^R), \quad (6)$$

где  $K_{EC}^R$  будем рассматривать как отношение переходов между состояниями из множества  $\{(v_{jk}, t_{jk})\}$ :  $K_{EC}^R((v_{jk}, t_{jk}), (v'_{jk}, t'_{jk}))$ ;  $K_M^R$  – функция интерпретации, задающая набор атомарных высказываний, истинных в модели  $M^R$  для конкретного состояния  $(v_{jk}, t_{jk})$ ;  $K_R^R$  – ограничения на использование неявных знаний для заданного набора состояний, связей между ними и правил интерпретации.

### Выводы

Предложена модель реляционных неявных знаний, которая определяет допустимый набор состояний для сущностей заданной предметной области, неявные отношения переходов между этими состояниями, интерпретацию взаимосвязей между ними, а также ограничения на использование неявных знаний. Модель обеспечивает возможность выделения неявных знаний на основе реализации состояний и переходов между этими состояниями, а также правил интерпретации как средства проверки допустимых действий в заданном состоянии.

**Список литературы:** 1. *Polanyi M. Personal Knowledge: Towards a Post-Critical Philosophy*/ M. Polanyi. – University of Chicago Press, 1958. 493 p. 2. *Polanyi M. The Tacit Dimension* / M. Polanyi. University of Chicago Press, 1966. 104 p. 3. *Нонака И. Компания - создатель знания. Зарождение и развитие инноваций в японских фирмах* / И. Нонака, Х. Такеучи. М., 2003. 384 с. 4. *McInerney C. "Knowledge Management and the Dynamic Nature of Knowledge"* / Claire McInerney // *Journal of the American Society for Information Science and Technology*. 2002. № 53 (12). 1009–1018. 5. *Smith E. A. The role of tacit and explicit knowledge in the workplace* / Elizabeth A. Smith // *Journal of Knowledge Management*. 2001. №5(4). P. 311–321. 6. *Cleeremans A. Principles for implicit learning.* / A. Cleeremans // In D. Berry (Ed./ *How implicit is implicit learning?*). Oxford: Oxford University Press. 1997. P. 195-234. 7. *Мильнер, Б. З. Управление знаниями в корпорациях* / Б. З. Мильнер, З. П. Румянцева, В. Г. Смирнова, А. В. Блишкова. М. : Дело. 2006. 304 с. 8. *Gascoigne N. and Thornton T. Tacit knowledge* / Neil Gascoigne and Tim Thornton. Acumen. 2013. 207 p. 9. *Collins H. Tacit and Explicit Knowledge* / Harry Collins. The University of Chicago Press. 2010. 186 p. 10. *Kripke S.A. Semantical Considerations on Modal Logic* / S.A. Kripke // *ActaPhilosophicaFennica*. 1963. №16. P. 83-94.

Поступила в редколлегию 21.09.2015

**Левыкин Виктор Макарович**, заведующий кафедрой информационно-управляющих систем ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 70-21-451.

**Чалая Оксана Викторовна**, канд.эконом.наук, доцент кафедры информационно-управляющих систем ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 70-21-451.

## ДИАГНОСТИРОВАНИЕ HDL-МОДЕЛЕЙ МИКРОПРОГРАММНЫХ АВТОМАТОВ

Рассматриваются методы поиска ошибок проектирования в HDL-моделях микропрограммных автоматов. Исходное описание автомата представляется композицией операционного и управляющего автоматов, которая описывается содержательной граф-схемой алгоритма. HDL-модель управляющего автомата дана в форме двухпроцессного автоматного шаблона автомата Мура. Диагностический эксперимент проводится путем обхода всех дуг графа переходов управляющего автомата, начиная от начальной вершины, путем эмуляции функций операционного автомата в системе верификации HDL-моделей (TestBench) в среде проектирования Active-HDL.

### 1. Введение

Цифровое операционное устройство (ОУ) можно представить комбинацией операционного автомата (ОА) и управляющего автомата (УА). Операционный автомат выполняет преобразование данных, а именно, выполняет микрооперации (МО), инициируемые управляющими сигналами  $Y = \{y_i\}$ , порядок следования которых определяется УА. Управляющий автомат определяет выполнение последовательности микроопераций на основе граф-схемы алгоритмов (ГСА) и множества оповестительных сигналов  $X = \{x_j\}$ , вырабатываемых ОА. Такой автомат принято называть микропрограммным управляющим автоматом [1]. В общем случае операционный автомат по способу описания не отличается от управляющего, а распределение ролей между двумя взаимосвязанными автоматами определяется наличием входных сигналов. Если входные сигналы присутствуют в обоих структурах, установить роль ведущего и ведомого достаточно сложно, поскольку любой автомат может быть описан графом переходов. Но классификация схемных стандартов операционного автомата по их функциям (триггеры, счетчики, регистры, память, коммутаторы, декодеры, АЛУ) позволяет оперировать ими как примитивами. В этом случае управляющий автомат рассматривается как способ описания взаимодействия примитивов операционного автомата во времени. Разделение конкретного проекта на УА и ОА субъективно и является следствием практического опыта и квалификации разработчика.

При существующем многообразии исходных форм описания проектов цифровых устройств (ЦУ) можно выделить наиболее популярные в мире: аналитические – языки описания аппаратуры (HDL), графические или визуальные – иерархические цифровые структуры и схемы, граф-схемы алгоритмов операционных или управляющих устройств (flow chart). Одним из распространенных способов исходного описания конечного автомата (УА) на языке описания аппаратуры является автоматный шаблон, т.е. специальная структура HDL-кода, которая строится на основе графа переходов автомата (state diagram) или прямой структурной таблицы. Построение графа переходов конечного автомата на основе других способов описания его функционирования является искусством проектировщика и особенностями инструментальных средств систем автоматизированного проектирования радиоэлектронной аппаратуры (САПР РЭА) [2].

Наиболее сложным и затратным этапом в современном цикле проектирования ЦУ является функциональная верификация, т.е. процесс обнаружения, локализации и устранения ошибок в системной модели относительно спецификации, на что затрачивается более половины общего времени проектирования. Основной формой описания проектов ЦУ в САПР РЭА являются языки описания аппаратуры, поэтому объектом верификации есть модель ЦУ, написанная на языке описания аппаратуры, т. е. HDL-модель.

Возможные ошибки проектирования в HDL-моделях определяются стилем описания HDL-кода. Под ошибкой проектирования понимается ошибка в HDL-операторе, которая не относится к классу синтаксических и нарушает алгоритм функционирования модели устройства, заданный спецификацией. Выделение фрагментов HDL-кода, описывающих пове-

дение конечных автоматов стилем «автоматный шаблон», позволяет определить ошибку проектирования типа «неправильный переход в графе переходов автомата», что соответствует ошибке в выборе текущего состояния в операторе when, ошибке выбора следующего состояния в функции переходов ( $a_i$  вместо  $a_j$ ), ошибке в операторе if( ) при анализе входного сигнала, ошибке в назначении выходного сигнала. Для проведения диагностического эксперимента (ДЭ) по поиску ошибок проектирования реализуется стратегия обхода всех дуг графа переходов конечного автомата, начиная с начальной вершины. При этом проверяются все одиночные неисправности переходов, а также исправности функций автомата, обеспечивающих эти переходы [3].

ДЭ над HDL-моделью конечного автомата состоит в подаче на нее входных воздействий в соответствии с выбранной стратегией обхода содержательного графа переходов, получении выходных реакций на Waveform и сравнения полученных реакций с эталоном. На основании этого делается вывод о соответствии HDL-модели спецификации. ДЭ проводится с использованием системы верификации HDL-моделей (TestBench) в среде проектирования Active-HDL. При проведении ДЭ в простых HDL-моделях УА подача входных воздействий и сравнение полученных реакций с эталонами не представляет особых трудностей даже в режиме визуального сравнения по Waveform, так как тестовые данные подаются непосредственно на входы автомата, а реакции снимаются с его выходов.

При проведении ДЭ для HDL-моделей микропрограммных автоматов задача подачи входных воздействий и сравнения выходных реакций с эталонами усложняется. С одной стороны, при проведении ДЭ по обходу всех дуг графа автомата фактически проверяется УА, а входные данные (операнды микропрограммы) подаются на ОА и с него же снимаются выходные реакции (результат). С другой стороны, входные данные УА (оповестительные сигналы  $x_i$ ) и выходные (сигналы инициализации микроопераций  $y_i$ ) не определены в спецификации и непосредственно сравниваться с эталоном при проведении ДЭ не могут. Таким образом, задача разработки методики проведения ДЭ над HDL-моделью микропрограммного автомата, заданного содержательной граф-схемой алгоритма микропрограммы, является актуальной. Эталонные реакции (функции выходов ОА) при этом определяются исключительно спецификацией на операции, выполняемые операционным устройством.

## **2. Подготовка диагностического эксперимента для микропрограммного автомата**

Особенностью проведения ДЭ по диагностированию HDL-модели (как и любого программного кода) является отсутствие эталонного программного кода, поэтому в качестве эталона можно использовать только результаты выполнения алгоритма микропрограммного ОА, определенного спецификацией. Таким образом, стратегия обхода всех дуг графа переходов УА реализуется путем построения последовательностей микроопераций обработки данных операционным автоматом.

Подготовку и проведение диагностического эксперимента по локализации ошибок проектирования в HDL-моделях микропрограммных автоматов будем рассматривать на примере микропрограммы (МКП) сложения четырехразрядных двоичных знаковых чисел в дополнительном модифицированном коде. Для данной микропрограммы в качестве УА рассматривается автомат Мура. На рис. 1,а показан фрагмент содержательной граф-схемы алгоритма указанной микропрограммы с отметками состояний УА для автомата Мура. Каждой операторной вершине ГСА (состоянию автомата Мура) соответствует набор управляющих сигналов  $u_i$ , каждый из которых инициирует выполнение определенной микрооперации ОА. Для упрощения дальнейшего изложения в данном фрагменте микропрограммы не анализируется результат сложения на переполнение разрядной сетки, что в целом не меняет структуры микропрограммы. На рис. 1, б приведен содержательный граф переходов УА Мура для данной микропрограммы, в котором латинскими буквами поименованы дуги графа.

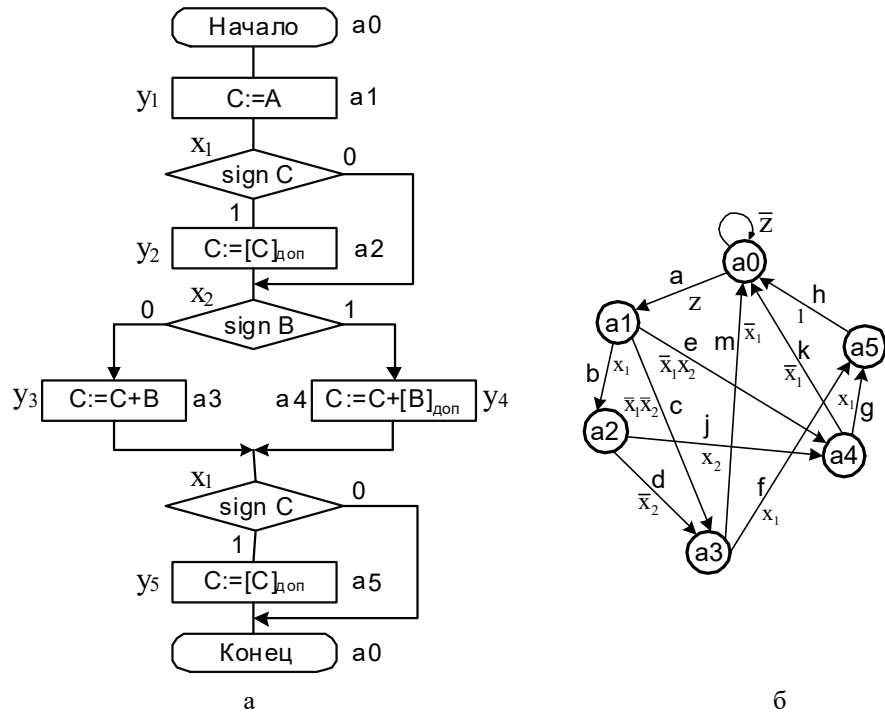


Рис. 1. Фрагмент содержательной ГСА микропрограммы сложения (а) и граф переходов управляющего автомата (б)

Для реализации стратегии обхода всех дуг графа по методике, предложенной в [3], на основании модифицированной матрицы смежности (рис.2,а) строится дерево решений для обхода путей (маршрутов) графа. В линейной ГСА, где логические условия следуют друг за другом, максимальное количество путей обхода графа автомата Мура будет  $2^n$ , где  $n$  – количество логических условий  $x_i$  с двумя альтернативами, т.е. для данного графа переходов будет 8 маршрутов обхода. При этом условие однократного покрытия всех дуг графа при его обходе на данном этапе не анализируется. Особенность данного дерева (рис.2,б) состоит в том, что терминальной вершиной во всех маршрутах обхода графа УА является состояние  $a_0$ , т.е. каждый путь в графе реализует полный цикл работы ОА.

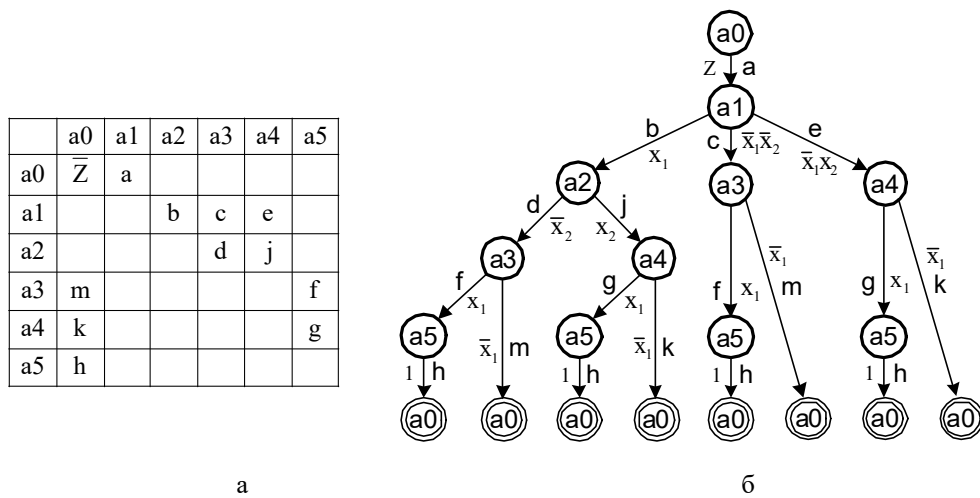


Рис. 2. Матрица смежности (а) и дерево решений для маршрутов обхода графа переходов управляющего автомата (б)

По дереву решений строится таблица проверок (маршруты обхода графа), по которым непосредственно и проводится диагностический эксперимент [4]. В таблице представлен полный перечень проверок  $P_i$  для маршрутов обхода графа. Каждая проверка характеризуется перечнем активизируемых дуг графа  $УА$ , перечнем управляющих сигналов для инициализации микроопераций  $u_i$  (вершин графа  $УА$ ) и диапазоном значений операндов (входных данных), которые данный маршрут обхода реализуют.

Полный перечень проверок при проведении ДЭ и перечень МО

М	a	b	c	d	e	f	g	h	j	k	m	Сигналы	Операнды и результат	Совместимая последовательность МО
												инициализации МО		
$P_1$	1	1		1		1		1				$u_1, u_2, u_3, u_5$	$C=(-A)+B<0$	да
$P_2$	1	1		1							1	$u_1, u_2, u_3$	$C=(-A)+B>0$	да
$P_3$	1	1					1	1	1			$u_1, u_2, u_4, u_5$	$C=(-A)+(-B)<0$	да
$P_4$	1	1							1	1		$u_1, u_2, u_4$	$C=(-A)+(-B)>0$	нет
$P_5$	1		1			1		1				$u_1, u_3, u_5$	$C=A+B<0$	нет
$P_6$	1		1								1	$u_1, u_3$	$C=A+B>0$	да
$P_7$	1				1		1	1				$u_1, u_4, u_5$	$C=A+(-B)<0$	да
$P_8$	1				1					1		$u_1, u_4$	$C=A+(-B)>0$	да

Любой маршрут обхода графа микропрограммного автомата порождает выполнение последовательности микроопераций, которые реализуют определенную часть ГСА микропрограммы (алгоритма обработки данных). Поэтому перед проведением ДЭ возникает необходимость рассмотрения возможности совместного выполнения определенных микроопераций. В теории проектирования микропрограммных автоматов используется понятие функциональной и структурной совместимости микроопераций [1]. Микрооперации называются функционально совместимыми, если в один момент времени они присваивают результаты разным операндам. Микрооперации называются структурно-совместимыми, если они одновременно не используют одни и те же аппаратные ресурсы ОА. В дальнейшем изложении управляющий сигнал инициализации микрооперации  $u_i$  и соответствующая ему микрооперация будут использоваться как синонимы.

При организации ДЭ по обходу графа микропрограммы возникает необходимость введения понятия совместимости последовательности микроопераций (во времени выполнения). Последовательность микроопераций (управляющих сигналов  $u_i$ ) называется совместимой, если существуют значения операндов ОА, которые позволяют реализовать данную последовательность. Если таких операндов не существует, то последовательность микроопераций называется несовместимой. Например, в операции сложения знаковых чисел с переполнением разнознаковые операнды никогда не дадут переполнения разрядной сетки. Таким образом, если в заданной последовательности микроопераций для сложения разнознаковых чисел присутствует  $u_i$ , реализующая выдачу сигнала переполнения разрядной сетки, данная последовательность является несовместимой. Несовместимость последовательности микроопераций бывает позитивной (в последовательности  $u_i$  присутствуют лишние микрооперации) или негативной (в последовательности  $u_i$  отсутствуют необходимые микрооперации). Это следует учитывать при формировании путей обхода графа переходов  $УА$ .

Если проанализировать таблицу, то можно сделать вывод, что последовательность микроопераций, реализующая проверку  $P_4 = \{u_1, u_2, u_4\}$ , определяет наличие положительной суммы (отсутствует МО  $u_5$ ) при отрицательных операндах (МО  $\{u_2, u_4\}$ ), что невозможно, и данная последовательность МО несовместима. Аналогично, последовательность МО, реализующая проверку  $P_5 = \{u_1, u_3, u_5\}$ , определяет наличие отрицательной суммы (присутствует МО  $u_5$ ) при положительных операндах (МО  $u_3$  при отсутствии МО  $u_2$ ) и также является несовместимой. Таким образом, проверки  $P_4$  и  $P_5$  нереализуемы и вектор экспериментальных проверок (ВЭП) будет  $V = (P_1, P_2, P_3, P_6, P_7, P_8)$ . При этом, если проанализи-

ровать оставшиеся строки табл.1, то отсутствие проверок  $P_4$  и  $P_5$  не нарушает полноту ДЭ по обходу всех дуг графа (во всех столбцах, соответствующих дугам графа  $\{a, b, c, d, e, f, g, h, j, k, m\}$ , присутствуют единицы, и условие однократного покрытия всех дуг графа соблюдается). Такая ситуация может быть не всегда, но рассмотрение условий получения однократного покрытия при обходе всех дуг графа не входит в задачи данного исследования.

В данной работе результат проведения ДЭ определяется путем визуального сравнения результатов моделирования HDL-модели на временной диаграмме (Waveform) с эталонными значениями (результатами выполнения операции сложения для разных операндов). При подборе операндов для реализации указанных проверок необходимо учитывать следующее:

- значения операндов должны быть такими, чтобы не было переполнения разрядной сетки, так как в рассматриваемом фрагменте микропрограммы отсутствует анализ переполнения;
- значения операндов для всех проверок должны быть такими, чтобы результат выполнения операции сложения для них различался.

На рис.3 приведен один из возможных вариантов значений входных данных и результатов для проверок  $\{P_1, P_2, P_3, P_6, P_7, P_8\}$ , что является эталонными значениями для проведения ДЭ. Для упрощения формирования эталонов на данном рисунке приведены обратные и дополнительные коды рассматриваемых операндов и результатов.

Десятичные числа	-2	-3	-4	-5	-8	-9
Прямой код	11 0010	11 0011	11 0100	11 0101	11 1000	11 1001
Обратный код	11 1101	11 1100	11 1011	11 1010	11 0111	11 0110
Дополнит. код (+1)	11 1110	11 1101	11 1100	11 1011	11 1000	11 0111

M	Операнды и знак результата	Десятичные данные и результат	Двоичные данные и результат выполнения сложения в дополнительном коде	Результат в прямом коде (+1)
$P_1$	$C=(-A)+B<0$	$(-9)+5 = (-4)$	$11\ 0111 + 00\ 0101 = 11\ 1100$	$11\ 0100$
$P_2$	$C=(-A)+B>0$	$(-3)+5 = 2$	$11\ 1101 + 00\ 0101 = 1\ 00\ 0010$	$00\ 0010$
$P_3$	$C=(-A)+(-B)<0$	$(-3)+(-5) = (-8)$	$11\ 1101 + 11\ 1011 = 1\ 11\ 1000$	$11\ 1000$
$P_6$	$C=A+B>0$	$9+5 = 14$	$00\ 1001 + 00\ 0101 = 00\ 1110$	$00\ 1110$
$P_7$	$C=A+(-B)<0$	$3+(-5) = (-2)$	$00\ 0011 + 11\ 1011 = 11\ 1110$	$11\ 0010$
$P_8$	$C=A+(-B)>0$	$9+(-3) = 6$	$00\ 1001 + 11\ 1101 = 1\ 00\ 0110$	$00\ 0110$

Рис. 3. Подготовка эталонов для проведения диагностического эксперимента

В заключение подготовки ДЭ приведем эталонную HDL-модель УА Мура (рис.4), построенную по графу переходов УА (см. рис.1,б). Как уже упоминалось выше, при проведении реального ДЭ эталонного кода не существует (есть только HDL-код «неизвестного происхождения» с возможными ошибками проектирования), но для наглядности изложения и иллюстрации процедур локализации ошибок в HDL-коде эталонный код и результаты его моделирования привести целесообразно.

```

library IEEE;
use IEEE.std_logic_1164.all;
entity FSM is
    port (Reset, Z, Clk, x1, x2: in STD_LOGIC;
          y1, y2, y3, y4, y5: out STD_LOGIC);
end;

architecture Moore of FSM is
    type State_type is (a0, a1, a2, a3, a4, a5);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <= NextState;
        end if;
    end process;

    Sreg0_NextState: process (State, x1, x2, Z)
    begin
        case State is
            when a0=> if Z='1' then NextState <= a1;
                       else NextState <= a0;
                       end if;
            when a1=> if x1='1' then NextState <= a2;
                       elsif x2='1' then NextState <= a4;
                       else NextState <= a3;
                       end if;
            when a2=> if x2='1' then NextState <= a4;
                       else NextState <= a3;
                       end if;
            when a3=> if x1='1' then NextState <= a5;
                       else NextState <= a0;
                       end if;
            when a4=> if x1='1' then NextState <= a5;
                       else NextState <= a0;
                       end if;
            when a5=> NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;

    y1 <= '1' when State=a1 else '0';
    y2 <= '1' when State=a2 else '0';
    y3 <= '1' when State=a3 else '0';
    y4 <= '1' when State=a4 else '0';
    y5 <= '1' when State=a5 else '0';
end;

```

Рис. 4. Эталонная HDL-модель УА, построенная по графу переходов

### 3. Построение TestBench

Любая САПР на основе HDL имеет в своем составе встроенную систему верификации HDL-моделей (TestBench). Данная система позволяет подавать на верифицируемую HDL-модель (UUT) входные воздействия, моделировать их, снимать выходные реакции и выводить результат на систему визуального отображения временных диаграмм (Waveform). Также существует возможность проводить верификацию с использованием специальной тестовой программы (HDL-кода) в рамках TestBench (рис.5).

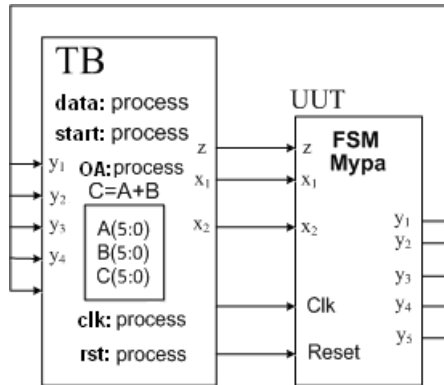


Рис. 5. Структура системы верификации (TestBench) при проведении ДЭ над УА

Тестовая программа TestBench (рис.6) состоит из ряда процессов:

**clk: process**, который формирует синхросигнал со скважностью  $2T$ , где  $T$  – полупериод синхросигнала;

**rst: process**, который формирует сигнал сброса исходя из числа проверок ДЭ;

**start: process**, который формирует сигналы запуска автомата и определяет полный цикл проведения ДЭ;

**data: process**, который обеспечивает подачу входных воздействий на HDL-модель УА и формирует цикл работы микропрограммы;

**OA: process**, который запускается сигналами инициализации микроопераций  $y_i$ , эмулирует работу ОА и вырабатывает оповестительные сигналы  $x_i$  с использованием операторов условного назначения сигналов **when**.

В тестовой программе также используются:

переменная **endsim** – признак конца моделирования;

константа  $T$  -  $1/2$  периода синхросигнала **Clk**;

константа  $N$  - максимальная длина пути проверки с учетом сброса и запуска;

сигналы **A**, **B**, **C** – операнды и результат в двоичном виде;

сигналы **Adec**, **Bdec**, **Cdec** – операнды и результат в десятичном виде.

Сигнал сброса **Reset** активизируется в начале каждого цикла проверки ДЭ, для реализации неразрушаемого эксперимента, позволяющего в случае невозврата автомата в начальное состояние (по каким-либо причинам) обеспечить его гарантированную установку в начальное состояние.

ОА срабатывает по переднему фронту, а УА – по заднему. Это необходимо в случае использования УА Мура. Если ОА и УА срабатывают по одному фронту, они не укладываются в один такт, выполняя полный цикл взаимодействия – генерацию управляющих сигналов и выполнения МО, инициируемых этими управляющими сигналами. Следствием этого является неверное формирование признаков состояния ОА и неправильная генерация следующих управляющих сигналов УА.

В автомате Мили таких проблем не возникает, ОА и УА могут срабатывать по одному фронту. Эти особенности природы разных типов УА не зависят от системы моделирования HDL-моделей. Моделирование выполнялось в системах VCS 2014.12, Riviera-PRO EDU 2015.06, Active-HDL и дало идентичные результаты.

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity fsm_tb1 is
end fsm_tb1;
architecture TB_ARCHITECTURE of fsm_tb1 is
  component fsm
  port(Reset, Z, Clk, x1, x2: in std_logic;
        y1, y2, y3, y4, y5: out std_logic);
  end component;
  signal Reset, Z, Clk, x1, x2, y1, y2, y3, y4, y5: std_logic;
  signal A, B, C: STD_LOGIC_VECTOR(5 downto 0);
  signal Adec, Bdec, Cdec: integer;
  shared variable endsim: BOOLEAN:=false; — конец моделирования
  constant T: time:= 5 ns; — 1/2 периода Clk
  constant N: integer:= 7; — максимальная длина пути проверки
begin
  — Unit Under Test port map
  UUT : fsm port map (Reset => Reset, Z => Z, Clk => Clk, x1 => x1, x2 => x2, y1 => y1, y2 => y2, y3 => y3,
y4 => y4, y5 => y5);
  — Add your stimulus here ...
  clock: process — синхронизация
  begin
    if not endsim then Clk<='0'; wait for T;
    Clk<='1'; wait for T;

    else wait;
    end if;

  end process;
  rst: process — сброс
  begin
    if not endsim then Reset <='1'; wait for 2*T;
    Reset <='0'; wait for(N-1)*2*T;
    else wait;
    end if;

  end process;
  start: process — запуск
  begin
    if not endsim then Z <='0'; wait for 2*T;
    Z <='1'; wait for 2*T;
    Z <='0'; wait for(N-2)*2*T;

    else wait;
    end if;

  end process;
  data: process — входные данные
  begin
    A <="111001"; B <="000101"; wait for N*2*T; — проверка P1
    A <="110011"; B <="000101"; wait for N*2*T; — проверка P2
    A <="111001"; B <="110101"; wait for N*2*T; — проверка P3
    A <="001001"; B <="000101"; wait for N*2*T; — проверка P4
    A <="000011"; B <="110101"; wait for N*2*T; — проверка P5
    A <="001001"; B <="110011"; wait for N*2*T; — проверка P6
    endsim := true;
    wait;
  end process;
  OA: process (Clk, y1, y2, y3, y4, y5) — Эмуляция ОА
  begin
    if Clk'event and Clk = '1' then
      if (y1 = '1') then C <= A;
      elsif (y2 = '1') then C <= C(5 downto 4)&((not C(3 downto 0))+1);
      elsif (y3 = '1') then C <= C + B;
      elsif (y4 = '1') then C <= C+(B(5 downto 4)&((not B(3 downto 0))+1));
      elsif (y5 = '1') then C <= C(5 downto 4)&((not C(3 downto 0))+1);
      end if;
    end if;

    x1 <= '1' when C(5 downto 4) = "11" else '0';
    x2 <= '1' when B(5 downto 4) = "11" else '0';
    — Отображение результата в виде десятичных чисел
    Adec <= -(ieee.std_logic_signed.conv_integer('0' & A(5 downto 0))) when A(5 downto 4) = "11" else
(ieee.std_logic_signed.conv_integer('0' & A(3 downto 0)));
    Bdec <= -(ieee.std_logic_signed.conv_integer('0' & B(3 downto 0))) when B(5 downto 4) = "11" else
(ieee.std_logic_signed.conv_integer('0' & B(3 downto 0)));
    Cdec <= -(ieee.std_logic_signed.conv_integer('0' & C(3 downto 0))) when C(5 downto 4) = "11" else
(ieee.std_logic_signed.conv_integer('0' & C(3 downto 0)));
  end TB_ARCHITECTURE;

```

Рис. 6. Фрагмент тестовой программы TestBench для проведения ДЭ

Данная структура TestBench является универсальной и может использоваться в качестве шаблона для проведения верификации любых микропрограммных УА, при этом необходимо менять только ОА: process (для соответствующей МКП) и подавать соответствующие данные в data: process. Для удобства визуального наблюдения на Waveform входные данные и результат преобразуются в десятичные числа.

На рис.7 изображена временная диаграмма работы устройства сложения (ОА+УА) на основе эталонного кода УА, приведенного на рис.4.

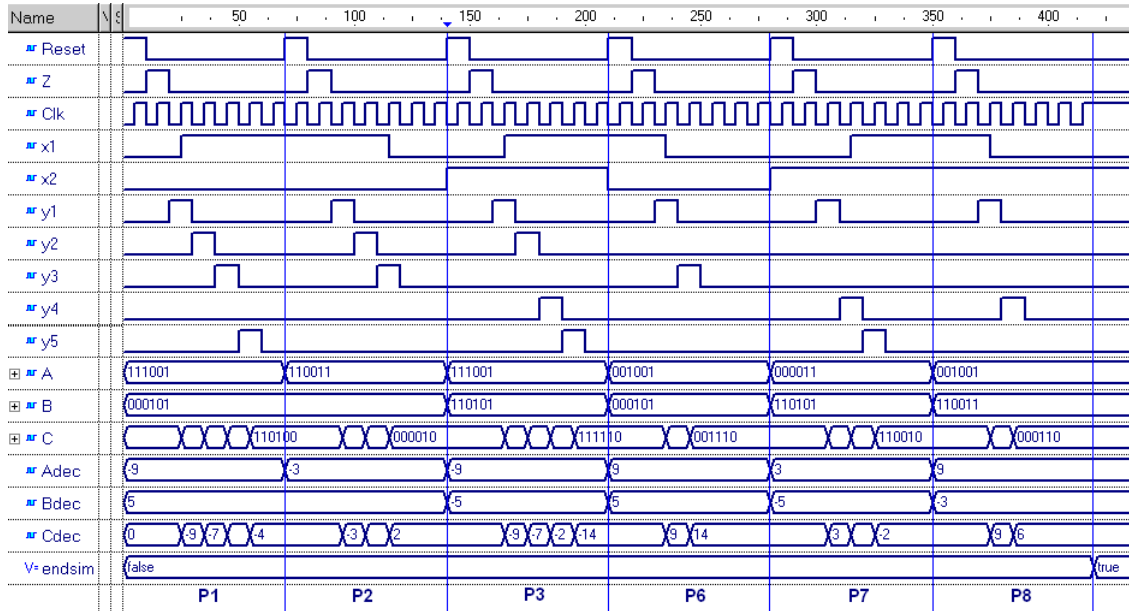


Рис. 7. Временная диаграмма моделирования работы эталонной HDL-модели устройства сложения с УА Мура

#### 4. Поиск ошибок проектирования в HDL-модели управляющего автомата

Для демонстрации методики поиска ошибок проектирования внесем ошибку в эталонный HDL-код. Предположим, что вместо перехода a2 – a3 реализуется переход a2 – a5. При этом не выполняется микрооперация u3 при условии, что была выполнена u2. Фрагмент ошибочного кода приведен на рис.8.

```

when a2=> if x2='1' then NextState <= a4;
           else NextState <= a5;
           end if;

```

Рис. 8. Фрагмент ошибочного HDL-кода, реализующего переход {a2 – a5} вместо {a2 – a3}

Диагностический эксперимент проводится путем реализации системы проверок {P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>6</sub>, P<sub>7</sub>, P<sub>8</sub>}. На рис.9 приведена временная диаграмма моделирования ошибочного кода. При анализе Waveform следует учитывать, что Cdec=Adec+Bdec в десятичном виде.

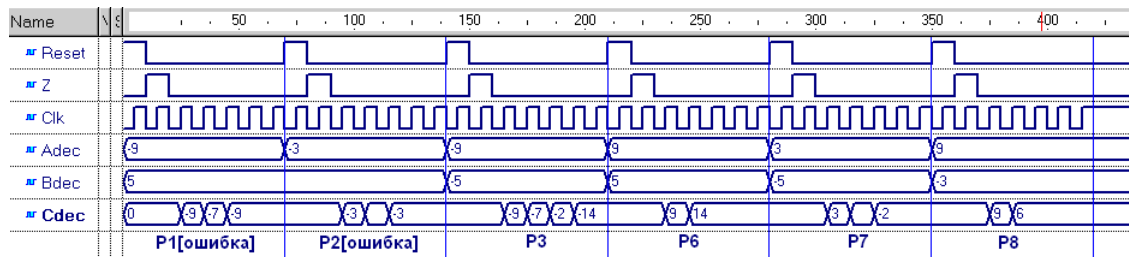


Рис. 9. Временные диаграммы проведения ДЭ для ошибочного HDL-кода УА

В результате анализа проведения ДЭ проверки  $P_1$  и  $P_2$  положительны (результат не совпал с эталоном), а остальные отрицательны, т.е. ВЭП будет  $V=(1, 1, 0, 0, 0, 0)$ .

Место возникновения ошибки в HDL-коде (перечень подозреваемых дуг графа,  $D$ ) определяется путем анализа маршрутов обхода графа переходов по формуле, где  $M_j$  –  $j$ -я строка таблицы проверок (таблица):

$$D = \bigcap_{v_j=1} M_j - \bigcup_{v_j=0} M_j, \quad (1)$$

$$D = \{a,b,d,f,h\} \cap \{a,b,d,m\} - \{a, b, g, h, j\} \cup \{a, c, m\} \cup \{a, e, g, h\} \cup \{a, e, k\} = \\ = \{a, b, d\} - \{a, b, c, e, g, h, j, k\} = \{d\}.$$

Таким образом, ошибочный оператор находится в группе операторов, связанных с состоянием  $a_2$  по ветви  $x_2=0$  (см. рис.1,б), что соответствует присутствующей ошибке проектирования.

## 5. Выводы

Предложен метод диагностирования HDL-модели микропрограммного автомата путем проведения диагностического эксперимента по обходу всех дуг графа управляющего автомата, начиная с начальной вершины. При этом осуществляется инициализация всех микроопераций операционного автомата, а операнды и результаты проверяются по спецификации реализуемой микропрограммы. Диагностический эксперимент проводится на примере микропрограмм сложения знаковых чисел в дополнительном модифицированном коде для управляющего автомата Мура. Введено понятие совместимых последовательностей микроопераций, которые обеспечивают проведение диагностического эксперимента для непротиворечивых числовых значений операндов. Локализация ошибок проектирования в HDL-модели управляющего автомата осуществляется до группы операторов HDL-кода, связанных с реализацией ошибочного перехода в графе автомата. Данные (оповестительные сигналы) на управляющий автомат подаются путем эмуляции функций операционного автомата в системе верификации HDL-моделей (Test Bench) САПР Active-HDL. Полученные результаты также проверялись в САПР VCS 2014.12 и Riviera-PRO EDU 2015.06.

Вопросы автоматизации выбора конкретных числовых значений операндов для квазиоптимального обхода всех дуг графа переходов управляющего автомата являются предметом дальнейших исследований.

**Список литературы:** 1. *Майоров С.А.* Структура электронных вычислительных машин / *С.А. Майоров, Г.И. Новиков.* Л.: Машиностроение, 1979. 384 с. 2. *Шкиль А.С.* Модель процесса перехода от содержательного графа микропрограммы к графу автомата / *А.С. Шкиль, В.И. Хаханов, Е.В. Ковалев* // АСУ и приборы автоматики. 2000. Вып. 112. С. 112-120. 3. *Шкиль А.С.* Поиск ошибок проектирования в HDL-моделях цифровых автоматов / *С. Альмадхоун, Е.Е. Сыревич, А.С. Шкиль* // Вестник Херсонского государственного технического университета. 2013. №2 (46). С. 377-383. 4. *Шкиль А.С.* Автоматизация поиска ошибок проектирования в HDL-моделях конечных автоматов / *А.С. Шкиль, Г.П. Фастовец, А.С. Серокурова* // АСУ и приборы автоматики. 2014. Вып.168. С. 43-52.

*Поступила в редколлегию 22.09.2015*

**Шкиль Александр Сергеевич**, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: диагностика цифровых систем, дистанционное образование. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26.

**Кулак Эльвира Николаевна**, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: автоматизированное проектирование цифровых автоматов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26.

**Серокурова Анна Сергеевна**, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых автоматов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26.

## РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ БЕСКОНТАКТНОГО ТЕПЛООВОГО КОНТРОЛЯ КАЧЕСТВА РАДИОЭЛЕКТРОННЫХ МОДУЛЕЙ НА ПЕЧАТНОЙ ПЛАТЕ

Разрабатывается программный модуль бесконтактного теплового контроля качества радиоэлектронных модулей на печатной плате при производстве РЭС. Рассматриваются и совершенствуются методы, необходимые для подготовки видимого изображения четырех печатных плат, выделения контуров всех элементов на них, а также нахождения неисправных элементов на печатных платах путем наложения полученных контуров на термограмму.

### Введение

Существует задача контроля качества радиоэлектронных модулей на печатной плате (ПП), а именно проверка правильности монтажа элементов и проверка на дефекты платы в целом.

Для решения этой задачи целесообразно использование теплового неразрушающего контроля качества печатных плат. Данный метод контроля имеет ряд преимуществ, таких как бесконтактность процесса и относительно низкая стоимость наряду с другими современными методами контроля качества ПП.

### Постановка задачи исследования

*Целью* данного исследования является разработка программного модуля для осуществления бесконтактного теплового контроля качества печатных плат.

Для проектирования данного программного модуля необходимо выбрать метод автоматизированного контроля качества ПП, осуществив при этом анализ существующих методов контроля, обосновать выбор метода теплового неразрушающего контроля, разработать алгоритм подготовки видимого изображения печатных плат (выбрать и совершенствовать методы для фильтрации изображения, повышения контраста и оконтуривания элементов). После подбора и доработки выбранных методов необходимо провести анализ полученных результатов контроля качества.

В процессе производства печатных плат используют различные методы контроля качества, которые предназначены для обеспечения выпуска продукции, отвечающей требованиям нормативных документов.

Существуют следующие методы контроля качества печатных плат:

- а) электрическое тестирование;
- б) оптический контроль;
- в) рентгеновский контроль;
- г) тепловой неразрушающий контроль.

Данные методы используются для контроля качества печатных плат на различных этапах изготовления. Для дальнейшего проектирования программного модуля был выбран тепловой неразрушающий контроль активного типа (с применением внешних нагревателей), так как его использование целесообразно при проверке готовой продукции.

В тепловых методах неразрушающего контроля используется тепловая энергия, распространяющаяся в объекте контроля. Температурное поле поверхности объекта является источником информации об особенностях процесса теплопередачи, которые, в свою очередь, зависят от наличия внутренних или наружных дефектов.

Различают пассивный и активный тепловой контроль. При пассивном тепловом контроле анализ тепловых полей изделий производят регистрацией их собственного теплового излучения. Активный тепловой контроль предполагает нагрев объекта внешним источником энергии.

Основной характеристикой температурного поля, являющейся индикатором дефектности, служит величина локального температурного перепада. Координаты места перепада, его рельеф или, иными словами, топология температурного поля и его величина в градусах являются функцией большого количества факторов. Эти факторы можно подразделить на внутренние и внешние. Внутренние факторы определяются теплофизическими свойствами контролируемого объекта и дефекта, а также их геометрическими параметрами. Эти же факторы определяют временные параметры процесса теплопередачи, в основном, процесса развития температурного перепада. Внешними факторами являются характеристики процесса теплообмена на поверхности объекта контроля (чаще всего величина коэффициента конвективной теплоотдачи), мощность источника нагрева и скорость его перемещения вдоль объекта контроля.

Существуют следующие способы активного теплового контроля изделий:

а) кратковременный локальный нагрев изделия с последующей регистрацией температуры той же (при одностороннем контроле) или противоположной области (при двустороннем контроле);

б) с использованием сканирующей системы, состоящей из жестко закрепленных относительно друг друга источников нагрева и регистрирующего прибора (например, радиометра), перемещающихся с постоянной скоростью вдоль поверхности образца;

в) одновременный нагрев поверхности образца вдоль некоторой линии с последующей регистрацией температуры вдоль той же линии (при одностороннем контроле) или вдоль аналогичной линии с противоположной поверхности образца (при двустороннем контроле). Подобная регистрация может быть осуществлена, например, прибором “Термопрофиль”;

г) одновременный нагрев всей поверхности образца и последующая одновременная регистрация температурного распределения на этой же или на противоположной поверхности. Подобный способ контроля может быть осуществлен при помощи тепловизора.

Эффективность выявления дефектов каждым из описанных способов теплового контроля уменьшается от первого к четвертому, а производительность - возрастает.

Бесконтактные методы теплового контроля основаны на использовании инфракрасного излучения, испускаемого всеми нагретыми телами. Инфракрасное излучение занимает широкий диапазон длин волн от 0,76 до 1000 мкм. На практике преимущественно используются два спектральных диапазона 3.5 и 8.14 мкм, совпадающие с окнами максимальной прозрачности атмосферы и являющиеся наиболее информативными. Спектр, мощность и пространственные характеристики этого излучения зависят от температуры тела и его излучательной способности, обусловленной, в основном, его материалом и микроструктурными характеристиками излучающей поверхности. Например, шероховатые поверхности излучают сильнее, чем зеркальные. При повышении температуры мощность излучения быстро растет, а ее максимум сдвигается в область более коротких длин волн. Эта закономерность характеризуется законом смещения Вина.

Характер спектра зависит, в основном, от агрегатного состояния вещества. Для твердых характерны непрерывные спектры излучения.

Для характеристики теплового излучения удобным оказалось понятие абсолютного черного тела (АЧТ), т.е. тела, поглощающего все падающее на него излучение. Излучение АЧТ описывается аналитически, оно является функцией только его температуры. Физической моделью АЧТ может служить замкнутая полость с отверстием, значительно меньшим ее габаритов.

Законы изучения АЧТ могут применяться с известной поправкой для большинства реальных тел, что определяет их значение.

Суммарную плотность потока излучения АЧТ в зависимости от его температуры определяет закон Стефана-Больцмана (получаемый интегрированием закона Планка).

В основу принципа действия тепловизионных приборов положено двумерное преобразование собственного теплового излучения от объектов и местности, или фона, в видимое изображение, что является одной из высших форм преобразования и хранения информации. Наличие в поле зрения регистрируемого теплового контраста позволяет визуализировать на мониторе полутоновые черно-белые или адекватные им “псевдоцветные” тепловизионные изображения.

Тепловизионная техника обладает рядом достоинств и присущих только ей возможностей: обнаружение удаленных теплоизлучающих объектов (или целей) независимо от уровня естественной освещенности, а также до определенной степени - тепловых и других помех (дождя, тумана, снегопада, пыли, дыма и др.).

Тепловизор позволяет выделять на тепловом изображении объекта области одинаковых температур с помощью изотерм, высвечивающихся на мониторе. В нижней части кадра формируется серая шкала, которая используется для измерения температуры. При этом яркость отдельных участков изображения объекта сравнивают с яркостью элементов шкалы, для которой при калибровке прибора определяют температурный перепад, соответствующий переходу от белого к черному.

Разрабатываемый программный модуль бесконтактного теплового контроля качества печатных плат должен производить:

- получение видимого изображения 4 печатных плат, находящихся на конвейере;
- фильтрации полученного изображения;
- повышение контраста изображения;
- оконтуривание всех элементов на печатной плате;
- анализ и подготовка термограммы, полученной в ходе нагревания элементов платы;
- выделение на термограмме зоны интереса в кадре, а именно области, содержащей дефект.

Для выделения на термограмме интересующей нас области, а именно области с превышением граничного значения интервала рабочих температур, производится нормализация видимого изображения с помощью восьми реперных точек, выделенных на термограмме и на видимом изображении.

Использование восьми реперных точек позволяет одновременно работать с четырьмя платами.

#### **Методы решения и полученные результаты**

Первым этапом разработки программного модуля контроля качества печатной платы является повышение контраста изображения. В отличие от существующих сглаживающих фильтров, которые снижают локальную контрастность изображения, размывая его, контрастоповышающие фильтры производят обратный эффект.

Ядро контрастоповышающего фильтра имеет значение, большее 1, в точке (0, 0), при общей сумме всех значений, равной 1. Например, контрастоповышающим фильтром является фильтр с ядром, задаваемым матрицей:

$$M_1^{\text{contr}} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix},$$

или матрицей:

$$M_2^{\text{contr}} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

Эффект повышения контраста достигается за счет того, что фильтр подчеркивает разницу между интенсивностями соседних пикселей, удаляя эти интенсивности друг от друга.

Для дельнейшей фильтрации изображения после повышения контрастности был выбран метод SUSAN, разработанный С.М. Смитом в 1995 году.

Использование метода SUSAN необходимо для очистки реального изображения от различных видов шумов. Этот метод включает в себя два этапа. Сначала определяется «шумовой» пиксель (как правило, в выявлении шума и состоит основная сложность). Затем значение шумового пикселя заменяется на новое, как правило рассчитанное из окружающих пикселей значение.

Обычно при использовании метода SUSAN рассматривается группа пикселей 5x5 элементов, центральной пиксель этой матрицы является тестируемым.

При разработке автоматизированного метода контроля качества группа обрабатываемых пикселей была уменьшена до 3x3, так как при работе с маленькими элементами ПП необходимо максимально точно фильтровать шумы. Использование матрицы пикселей 3x3 несколько замедляет работу программы, но показывает более точные результаты, необходимые для следующего этапа оконтуривания элементов.

В ходе проверки рассчитывается отклонение яркости пикселя от среднего значения яркости. Если фильтр «решает», что такого пикселя быть не должно, его «шумовое» значение заменяется новым, вычисленным на основе окружающих пикселей.

Критерий определения шума в данном методе состоит в рассмотрении  $n$  пикселей, попавших в матрицу пикселей.

Находим сумму отклонений яркости пикселей от среднего значения:

$$S = \sum_{i=0}^{n-1} \Delta_i,$$

где  $\Delta_i = |\bar{b} - b_i|$ ;  $b_i$  – значение пикселя  $i$ ;  $\bar{b} = \sum_{i=0}^{n-1} (b_i / n)$  – среднее значение яркости.

Определяется относительный вклад отклонения  $\Delta_k$  тестируемого пикселя в значение  $S$ :

$$P_k = \frac{\Delta_k}{S},$$

здесь  $k$  – номер тестируемого пикселя.

Очевидно, что  $\sum_{i=0}^{n-1} p_i = 1$ .

Если в рассматриваемом фрагменте изображения наблюдается более-менее равномерное распределения яркостей пикселей, то и значение  $p_i$  будет не сильно отличаться от  $1/n$ .

Яркость шумового пикселя значительно отличается от средней яркости окружающих его пикселей.

Величина  $\Delta_k$  такого пикселя больше, чем у остальных пикселей, а значит величина  $P_k$  будет превышать  $1/n$ . Это и является критерием шумового пикселя.

Если  $P_k > \frac{1}{n}$ , значит, пиксель  $k$  является шумом.

После выявления шумового пикселя необходимо решить, что с ним делать. Здесь возможны следующие варианты:

- заменить шумовой пиксель средним значением  $\bar{b}$ ;
- заменить шумовой пиксель средним значением, вычисленным с учетом значений всех пикселей, за исключением шумового.

В усовершенствованном методе автоматизированного контроля использован другой вариант решения задачи выявления шумового пикселя, так как он позволяет наиболее точно найти шумовые пиксели.

Необходимо заменить шумовой пиксель средним значением, посчитанным с учетом значений всех пикселей, не удовлетворяющих критерию отбора шума. Данное решение предполагает, что в рассматриваемый фрагмент может попасть более одного пикселя, удовлетворяющего критерию «шум», и при расчете нового значения их не следует учитывать.

Представлен результат обработки элемента смонтированным на ПП усовершенствованным методом SUSAN (рис.1).

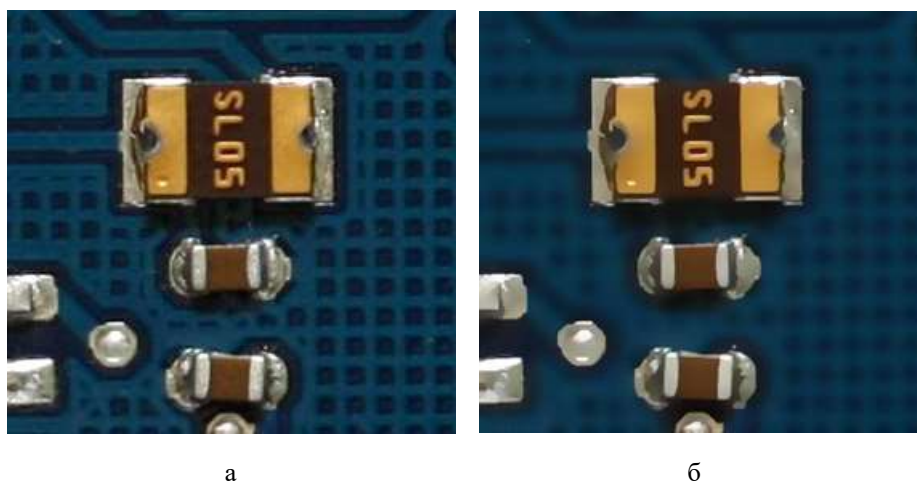


Рис. 1. Результат обработки видимого изображения методом SUSAN: а – исходное изображение; б – обработанное

Следующим шагом при подготовке видимого изображения является выделение контуров.

В разрабатываемом методе автоматизированного контроля качества для усовершенствования выбран метод Робертса и разностный метод.

При усовершенствовании первого метода матрица преобразований уменьшена до размера 3x3 пикселя и в ходе запуска формировалась лишь после предварительной фильтрации изображения печатных плат.

Так же определяются весовые коэффициенты пикселей в операциях сложения.

Яркость пикселей, попавших в область действия матрицы, примерно одинакова. Это значит, что после сложения получится сумма, близкая к нулю. Если же преобразуемый пиксель (ему соответствует центральный элемент матрицы 3x3) имеет яркость, превышающую окружающие пиксели, то результат сложения будет больше нуля.

Сумма элементов матрицы равна нулю. Поэтому изображение превратится в черно-белое. Если же центральный элемент матрицы сделать, например, равным 9, то тогда цвета изображения в основном не изменятся и будут выделены лишь границы.

Как было отмечено ранее, перед началом работы данного метода необходимо произвести предварительную фильтрацию усовершенствованным методом SUSAN.

На рис. 2 представлен результат использования стандартного и усовершенствованного метода Робертса после предварительной фильтрации изображения.

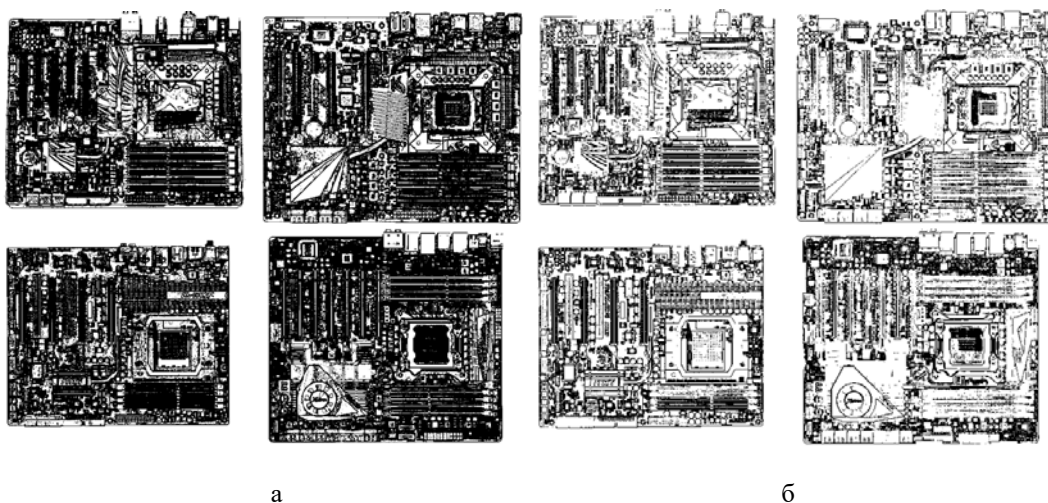


Рис. 2. Результат выделения контуров элементов на ПП с помощью метода Робертса после фильтрации: а – стандартный метод; б – усовершенствованный метод

Следующим этапом является наложение выделенных контуров элементов ПП на термограмму. На рис. 3 представлен результат наложения полученного контура на термограмму.

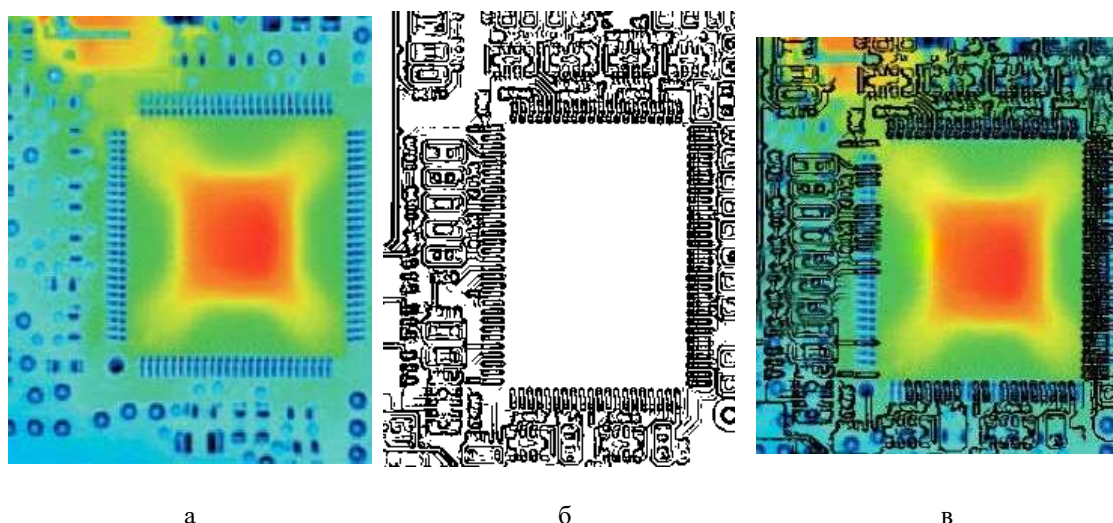


Рис. 3. Результат наложения контура на термограмму: а – часть термограммы; б – часть полученного контура; в – результат наложения полученного контура на термограмму

Разработанный программный модуль контроля качества также сопоставляет полученные результаты с базой данных, а именно обнаруживает пиксели («шумовые пиксели»), которые будут отображать перегрев элементов.

«Шумовые» пиксели автоматически сверяются с ранее разработанной базой данных. Данная БД содержит в себе наименования элементов и интервалы рабочих температур для каждого из элементов. Также она содержит положение каждого из элемента на плате. Если показание температуры выходит за границы интервала рабочих температур, то разработанная программа выводит сообщение о том, что элемент бракован.

#### **Выводы**

В ходе разработки программного модуля был выбран и обоснован метод теплового неразрушающего контроля. Также были выбраны и усовершенствованы методы подготовки видимого изображения, наложения полученных контуров на термограмму. Использование разработанного модуля позволит ускорить процесс одновременного контроля качества 4 печатных плат.

**Список литературы:** 1. Программно-технические средства диагностики цифровых моделей систем управления технологическим оборудованием [Текст] : [моногр.] / И. Ш. Невлюдов, М. А. Омаров, Б. А. Шостак ; МОН Украины. Х. : НТМТ, 2008. 216 с. 2. Пуятин Е. П. Обработка изображений в робототехнике [Текст] / Е. П. Пуятин, С. И. Аверин. М. : Машиностроение, 1990. 320 с. 3. Поляков А.Ю., Брусенцов В.А. Методы и алгоритмы компьютерной графики в примерах на Visual C++. СПб.: БХВ-Петербург, 2003. 560 с.

*Поступила в редколлегию 14.09.2015*

**Гурина Дарья Владимировна**, аспирантка кафедры ТАПР ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

**Гарячевская Ирина Васильевна**, канд. техн. наук, доцент кафедры информационных технологий в физико-энергетических системах, заместитель декана по учебной работе физико-энергетического факультета Харьковского национального университета имени В.Н. Каразина. Адрес: Украина, 61022, Харьков, пл. Свободы, 4.

## УПРАВЛЕНИЕ ЗНАНИЯМИ НА АВТОМАТИЗИРОВАННОМ ПРЕДПРИЯТИИ

---

Внедрение управления знаниями на автоматизированном предприятии далеко не всегда обеспечивает ожидаемый результат. Основная причина – различные толкования базовых понятий, следствием чего является подмена управления знаниями другими видами управлений. Излагается точка зрения, выявляющая концептуальные ограничения современных систем управления знаниями.

### 1. Введение

Для современного предприятия характерно наличие компьютерной системы, поддерживающей его деятельность. И когда на таком предприятии начинают создавать систему управления знаниями (УЗ), то часть существующей системы и элементы ее развития называют по-новому. Управление знаниями официально, как направление, недавно отметило свое двадцатилетие. Накопилось уже много наработок, существуют разные точки зрения, количество публикаций экспоненциально возрастает, все большее количество людей касаются данной тематики, как представители научной сферы, так и практики, на предприятиях которых занимаются управлением знаниями.

Из-за различных трактовок основополагающих понятий на практике мало кто занимается действительно управлением знаниями, его подменяют управлением информацией или интеллектуальным капиталом. Отсюда и низкая отдача. В [1] как раз и указывается на убыточность систем управления знаниями, хотя предлагаемый там подход к решению проблемы оставляет много вопросов.

Путаница возникает из-за того, что управление знаниями сильно связано с такими подсистемами как управление персоналом, управление информацией и данными, управление интеллектуальным капиталом. И поскольку результатом управления знаниями является обретение сотрудниками знаний, то оно сильно влияет и на остальные подсистемы управления и на управление предприятием в целом.

Основная причина неудач в том, что все еще не уточнены основные понятия этой дисциплины, а именно: «знание», «управление знаниями», «система управления знаниями», а также не определены границы области управления знаниями. М.Желены [2] обращал внимание на важность уточнения понятий, причем для быстроразвивающейся области это следует делать не только один раз где-то в начале, но и периодически возвращаться к уточнению их содержания. Для осмысливания понятия «знание», обозначающего виртуальную сущность и имеющего различные трактовки, необходимо также привлекать знания из других областей, чему многие авторы не уделяют должного внимания.

*Цель исследования:* поиск путей повышения эффективности систем управления знаниями.

*Задача исследования:* определить «скрытые» ограничения современных систем управления знаниями.

### 2. Система формирования знаний сотрудника

Основное назначение управления знаниями можно сформулировать как постоянное увеличение человеческого капитала предприятия, который, главным образом, определяется знаниями сотрудников. Поэтому уместно было бы говорить об управлении знаниями сотрудников, а не какими-то знаниями вообще. При обучении человек воспринимает не знания (даже когда обучается с учителем), а только информацию, которую он «автоматически» переводит в знания. В [3, 4] показаны некоторые причины искажения поступающей информации и ошибки перевода ее в знания. Преобразование человеком информации в знания упрощенно показано на рис.1.

Поступающая через органы чувств информация подвергается «фильтрации» [3], искажениям «призмой» [4], затем она дополняется до целостного образа (в какой-то мере

ошибочного) [4], с которым и работает функция обучения. На все эти преобразования сильно влияет текущее содержимое «базы знаний» человека. Формально это можно показать расширением модели, рассмотренной в [5]:

$$\begin{aligned}
 S &\subset X \times Y, \\
 \bar{\omega} &= \{\omega_t : Z_t \times I_t \rightarrow I'_t \ \& \ t \in T\}, \\
 \bar{\psi} &= \{\psi_t : Z_t \times I'_t \rightarrow I''_t \ \& \ t \in T\}, \\
 I &= \{X, Y\}, \\
 \bar{\rho} &= \{\rho_t : Z_t \times X''_t \rightarrow Y_t \ \& \ t \in T\}, \\
 \bar{o} &= \{o_{tt'} : Z_t \times X''_t \times Y''_t \rightarrow Z_{t'} \ \& \ t, t' \in T \ \& \ t' > t\}, \\
 \bar{\varphi} &= \{\varphi_{tt'} : Z_t \rightarrow Z_{t'} \ \& \ t, t' \in T \ \& \ t' > t\}, \\
 T &= \{t : t \geq 0\},
 \end{aligned}$$

где  $S$  – система;  $X$  – входной объект (информация, которую человек воспринимает);  $Y$  – выходной объект (принятые решения, выполненные действия и т.п.);  $Z$  – состояние (знание, которым обладает человек);  $\bar{\omega}$  – семейство функций, описывающее прохождение информации через «фильтр» и «призму»;  $\bar{\psi}$  – семейство функций, описывающее формирование образа на основе поступившей частичной и искаженной информации;  $\bar{\rho}$  – семейство реакций системы  $S$ ;  $\bar{o}$  – семейство функций перехода состояний (изменение знаний в процессе деятельности);  $\bar{\varphi}$  – семейство функций перехода состояний (изменение знаний при размышлениях);  $T$  – множество моментов времени.

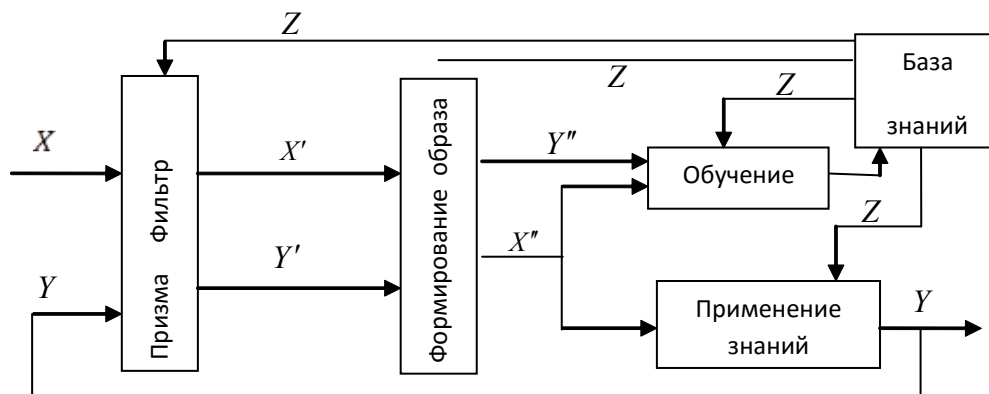


Рис. 1. Упрощенная схема системы преобразования человеком информации в знания

Функции  $\bar{\omega}$  и  $\bar{\psi}$  представляют естественные ограничения на овладение человеком знаниями. Обычно их эффект не учитывается, и конечный результат УЗ оказывается существенно скромнее возможного.

### 3. Знания и проблемы его передачи

Определений понятия «знание» очень много. Для УЗ достаточно конструктивным является определение, предложенное в [5]. Там знания определяются через их основные свойства, а именно: знания не материальны, находятся внутри человека; определяют представления человека о внешнем мире; отражают и аккумулируют человеческий опыт; определяют возможности человека получать конкретные результаты. Такой взгляд на знания позволяет построить более четкую границу между УЗ и управлением информацией.

Если знания не материальны и находятся внутри человека, то для передачи их от человека к человеку требуется «посредник». В [6] приводится схема передачи знаний, как иллюстрация проблемы извлечения знаний из текстов. Можно сказать, что в качестве «посредника» был использован текст. В общем случае «посредником» могут быть любые информационные сообщения, воспринимаемые органами чувств. Для предотвращения путаницы при использовании термина «знание» в [5, 7] применялось понятие «описание знаний». К описаниям знаний относим учебники, инструкции и т.п., также сюда отнесем и сообщения, которые формирует обучающий обучаемому. С учетом сказанного схему из [6] представим в несколько измененном виде на рис.2.

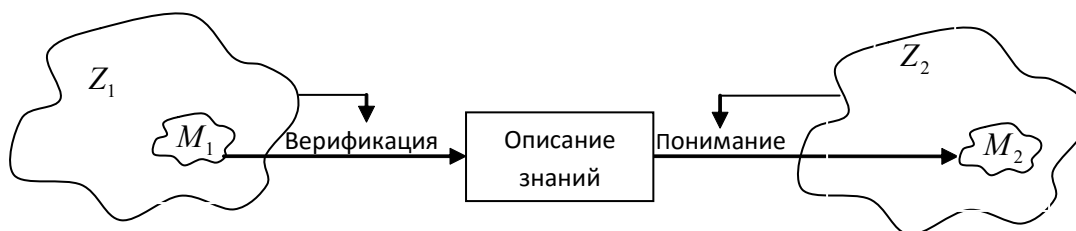


Рис. 2. Схема передачи знаний

В передачу знаний через «посредника» заложены две погрешности, получающиеся при составлении описания и при освоении описания. Описание знаний всегда является неполным и зависит от состояния знаний автора в момент создания описания. Освоение знаний осуществляется системой S (рис.1) и порождает соответствующие неточности. Поэтому задача освоения знания по описанию состоит в достижении наиболее полного соответствия  $\mu_1$   $\mu_2$ , т.е.

$$\|\mu_1 - \mu_2\| \rightarrow \min, \mu_1 \subset Z_1, \mu_2 \subset Z_2,$$

где  $\mu_1$  – знания (смысл), закладываемые автором описания;  $\mu_2$  – знания (смысл), получаемые обучаемым. Воспринимая описания знаний, каждый обучаемый будет строить свою ментальную модель (получать свои знания, отличные от его коллег, работающих с теми же самыми описаниями). Это проявление действий функции  $\bar{\omega}$  и  $\bar{\psi}$ . Одно и то же сообщение будет формировать различные кусочки знаний у разных людей, поэтому не следует отождествлять знания с информацией, хотя многие авторы определяют знание как информацию. Например, в [8]: «Любая организация является как получателем знаний из внешней среды, так и источником новых знаний». Здесь требуется уточнить. Организация не получает знания из внешней среды. Она получает информацию и/или описания знаний (информационные сообщения). Даже ученик, слушающий учителя, не получает знания. Ученик и организация по полученным сообщениям сами вырабатывают знания. Далее, там же: «Информация — это «сырье» для производства знаний, в процессе ее переработки возникают новые знания». Это утверждение уже почти верно, если иметь в виду конкретного человека или конкретную организацию. Но в общем случае это не всегда так: полученными знаниями уже кто-то другой мог обладать.

Еще замечание. Опять же в [8]: «Перспективность предприятия определяется ... способностью персонала преобразовывать информацию в знания и рационально их использовать». Способность преобразовывать информацию в знания заложена в человеке с рождения, и мы ею постоянно пользуемся. Существующая система знаний конкретного человека на момент поступления новой информации определяет, будет ли эта информация воспринята и как. А степень рациональности использования зависит от того, какими именно знаниями обладает человек.

Как видно из сказанного, учет особенностей человека работать с информацией позволяет строить более полезные системы управления знаниями.

#### 4. Управление знаниями и другие виды управления

Результатом управления знаниями являются знания персональные, групповые, корпоративные, которыми владеет персонал предприятия.

Разграничение близких (а у некоторых авторов и пересекающихся) управлений можно осуществить по объекту управления. Это известно, но при описании системы управления знаниями (СУЗ) возникает путаница. В таблице представлены факторы, разграничивающие близкие управлению знаниями виды управления.

Через свой объект управления (знания) УЗ оказывает влияние на все подразделения предприятия и менеджмент, даже на управление предприятия в целом (рис.3.). Некоторые авторы ранних публикаций УЗ отождествляли с управлением предприятием и утверждали, что УЗ – это следующий уровень развития менеджмента предприятия.

Наименование управления	Объект управления	Основное назначение
Управление информацией и данными (информационный менеджмент)	Информация и данные, необходимые для успешной деятельности предприятия	Информационное обеспечение всех подразделений и менеджмента
Управление персоналом	Весь персонал (исполнители и менеджеры)	Обеспечение предприятия квалифицированным персоналом. Организация обучения и развития.
Управление интеллектуальным капиталом	Интеллектуальные активы предприятия	Развитие и эффективное использование интеллектуальных активов предприятия
Управление знаниями	Знания персонала, знания, помещенные в компьютеры (СППР, и др. системы)	Обеспечение персонала знаниями, необходимыми для постоянного роста конкурентоспособности предприятия
Управление ИТ-инфраструктурой предприятия	ИТ-инфраструктура	Обеспечение предприятия необходимым уровнем ИТ-инфраструктуры

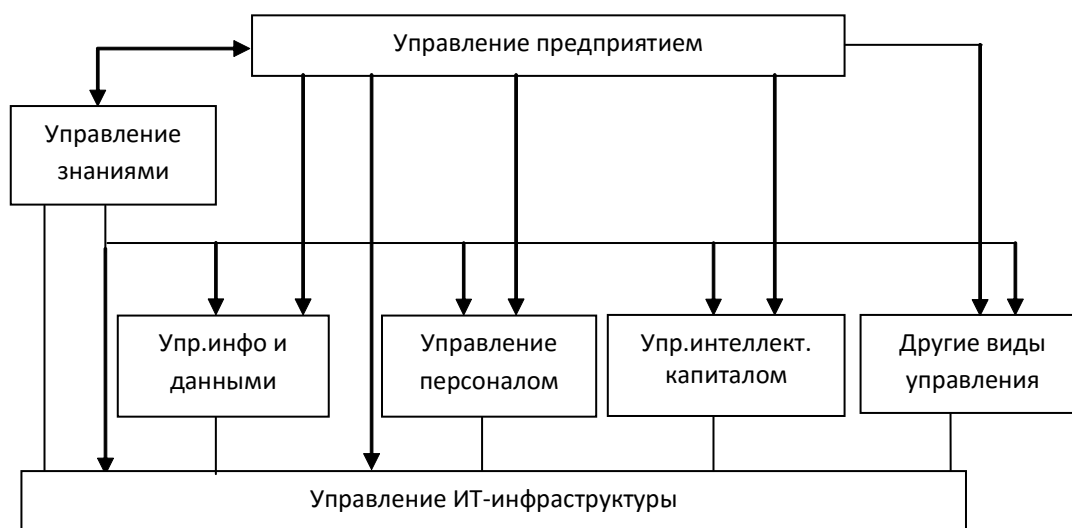


Рис. 3. Влияние УЗ на другие подсистемы управления

Перечень уже ставших традиционными технологических процессов УЗ следует также подкорректировать.

Процесс извлечения знаний из источников представляет собой преобразование информации в описание знаний, которое выполняет подготовленный специалист (по знаниям). Могут использоваться вспомогательные средства: *datamining*, *textmining*. Это прямое извлечение. Извлечение через посредника: специалист предметной области изучает источник знаний и применяет полученные знания. Затем выполняется прямое извлечение знаний у этого специалиста.

Структурирование описаний знаний – разбиение их на смысловые части или единицы с установлением возможных связей, построение схем и моделей, как, например, при структурировании процедурного знания [7].

Хранение знаний – не совсем корректный термин: знания хранятся в головах их владельцев. Лучше назвать хранением описаний знаний, в том числе и структурированных, причем такое хранилище не совсем корректно называть базой знаний. Заполнив базу текстовой информацией (отчеты, инструкции, положения и т.п.), не получим базу знаний. Это база текстовых данных. А оперировать удобнее со структурированными описаниями знаний.

Распространение знаний – процесс специального обучения персонала. Он может осуществляться с использованием структурированных описаний знаний, а также при организации передачи непосредственно от человека человеку, например, консультации эксперта, сообщества практиков, сообщества обучающихся, наставничество.

Процесс «Использование/применение знаний» можно исключить из УЗ, так как это фактически есть выполнение персоналом своих обязанностей. Как известно, выполнять работу человек может только применяя знания, а в процессе их применения знания совершенствуются, также практика добавляет и новый опыт. Позже такой сотрудник сможет участвовать в процессе извлечения знаний.

Близкая ситуация и с процессом генерации новых знаний. Целенаправленная генерация новых знаний может осуществляться при выполнении научно-исследовательских и опытно-конструкторских работ, решении сложных проблем, внедрении новой техники, технологий и т.п. Результаты рассматриваются как источники знаний, к которым может применяться процесс извлечения знаний и затем структурирование описаний знаний. Процесс генерации новых знаний должен быть организован, но организация процесса является прямой обязанностью управления предприятием, а не управления знаниями.

К управлению знаниями следует отнести процесс поиска и анализа источников знаний, возможности использования их в процедуре извлечения.

Важным и достаточно трудным вопросом является определение того, какие именно знания необходимы. Здесь существуют психологические проблемы (см. рис.1). Другая разновидность проблем возникает из-за того, что, формулируя, какие знания необходимы для выполнения работы, обычно рассматривают только видимую часть «айсберга», а все, что скрыто и отличает мастера от любителя, не учитывается.

## **5. Система управления знаниями**

Система управления знаниями представляет собой систему, реализующую технологию управления знаниями. В публикациях нередко встречаются схемы систем, которые нельзя отнести к СУЗ. Например, в [9] приведена схема, которая, скорее, относится к системе поддержки принятия решений (СППР), хотя в качестве инструмента СУЗ может содержать экспертную систему (ЭС). Но ЭС тоже не СУЗ.

Другой пример. В [8] отмечается, что «традиционно в систему управления знаниями (*Knowledge management*) включались следующие подсистемы:

- делопроизводство и информационное обеспечение подразделений;
- обучение и развитие – обеспечивает передачу знаний и опыта новым поколениям работников;
- организация НИОКР, инновационной и рационализаторской деятельности работников;
- внешние и внутрикорпоративные коммуникации».

Здесь тоже функции других подсистем перенаправлены в СУЗ. Подсистема делопроизводства и информационного обеспечения подразделений относится к управлению информацией и данными. Подсистема обучения и развития персонала относится к подсистеме управления персоналом, хотя частично может быть отнесена к СУЗ, т.е. только сам

процесс формирования необходимых знаний работников. А вопросы планирования, мотивации, аттестации, учета остаются в управлении персоналом. Организация НИОКР, инновационной и рационализаторской деятельности работников относится к управлению предприятием, а результаты являются входом для СУЗ. Внешние и внутренние коммуникации может быть не совсем подсистема, но могут использоваться как инструменты УЗ.

СУЗ может (и должна) применять ресурсы и инструменты других систем, например, обращаться к хранилищам данных через собственные средства data mining для выявления скрытых закономерностей, которые затем будут преобразованы в описания знаний. Аналогично, text mining оперирует с базами текстовых документов. Текстовый документ не является знанием и база не относится к СУЗ, но text mining может извлекать информацию, необходимую для управления знаниями. Также из СУЗ может осуществляться доступ к каталогу базы текстовых и графических документов, который обслуживается в рамках информационной системы предприятия. Что касается «золотых страниц» экспертов (носителей знаний), должно быть принято решение, относить ли их к СУЗ или к информационной системе. СУЗ может иметь и свои специфические базы, например, такие как база структурированных описаний знаний.

В состав СУЗ может входить ЭС как электронный наставник-консультант, реализующая концепцию обучения на рабочем месте. Это же и способ распространения знаний.

СУЗ естественно использует ИТ-технологии и базируется на ИТ-инфраструктуре предприятия. Это касается и других видов управления. Но когда говорим о СУЗ, то следует рассматривать ИТ-инструменты, которые помогают или обеспечивают выполнение конкретных функций управления знаниями.

## **6. Заключение**

Учет особенностей человека по восприятию информации и формированию знаний снимает «скрытые препятствия» на пути совершенствования СУЗ. Понимание этих особенностей позволяет более четко установить содержание понятий «знание», «управление знаниями», «система управления знаниями». Подмена СУЗ информационными системами снижает эффект УЗ, так как для получения необходимых знаний сотруднику придется проработать огромное количество документов и нет никакой гарантии, что требуемые знания будут получены.

*Научная новизна* – предложена модель, описывающая особенности человека по восприятию информации и формированию знаний. Более четко очерчена область УЗ.

*Практическая значимость* – руководствуясь изложенными идеями, становится возможным повысить полезность и эффективность УЗ на автоматизированных предприятиях.

**Список литературы:** 1. *Мирошниченко А., Фадеев П., Терпиловский Г.* 9 законов убыточности систем управления знаниями. 2012. Электронный ресурс. [http://www.know-man.com/2012/08/9\\_8.html](http://www.know-man.com/2012/08/9_8.html). 2. *Желены М.* Знание против информации / Информационные технологии в бизнесе / Под ред. М.Желены. СПб.: Питер, 2002. С. 211 – 218. 3. *Лопатников Л.И.* Экономико-математический словарь: Словарь современной экономической науки. - М.: Дело, 2003. 520 с. 4. *О'Коннор Дж., Макдермот И.* Искусство системного мышления: Необходимые знания о системах и творческом подходе к решению проблем / Пер. с англ. М.: Альпина Бизнес Букс, 2006. 256 с. 5. *Вишняк М.Ю., Довгань С.С.* Расширение функций АСУ: управление знаниями/АСУ и приборы автоматизации. 2008. Вып. 145. С. 109-116. 6. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных систем. М.: Радио и связь, 1992. 200 с. 7. *Вишняк М.Ю., Довгань С.С.* Модель структурированного представления знаний о процедурах / Бионика интеллекта. 2009. №2 (71). С.93-98. 8. *Турик И., Плескач В.* Управление знаниями на промышленном предприятии. Менеджер по персоналу. 2009. №4. С.8-20. 9. Менеджмент знаний: подход к внедрению. 2015. Электронный ресурс: <http://md-hr.ru/articles/html/article32613.html>

*Поступила в редколлегию 12.09.2015*

**Вишняк Михаил Юрьевич**, канд. техн. наук, доцент кафедры системотехники ХНУРЭ. Научные интересы: современные технологии управления организационными сетями. Применение информационных технологий в реализации концепции знаний. Адрес: Украина, 61166, Харьков, пр. Науки, 14. тел. (057) 702-10-06, (057) 345-12-57, (095) 887-44-09, E-mail: mvishnyak95@gmail.com.

**Климова Ирина Николаевна**, аспирантка кафедры системотехники ХНУРЭ. Научные интересы: управление знаниями в организационных системах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14. тел. (057) 702-10-06, (098) 955-04-06, E-mail: kler.86@bk.ru

## О ВОЗМОЖНЫХ ПРЕДСТАВЛЕНИЯХ МАТРИЦ ЛИНЕЙНЫХ ЛОГИЧЕСКИХ ОПЕРАТОРОВ

Показывается возможность представления матриц линейных логических операторов по аналогии с другими формами, такими, как графическая интерпретация логических пространств и описание с помощью бинарных предикатов. По результатам проведенных исследований предлагаются различные формы представления матриц линейных логических операторов. Рассматриваются примеры представления матриц линейных логических операторов.

### 1. Введение

Устоявшиеся представления о математической логике как о науке, изучающей законы мышления с применением аппарата математики, главным образом, для нужд самой математики, в современных условиях становится слишком узким [1]. Объектами математической логики являются любые дискретные конечные системы, а ее главная задача – структурное моделирование таких систем. Человеческий язык, как явление дискретное, естественно должен описываться средствами дискретной математики.

Для описания естественного человеческого языка лучше всего подошел бы аппарат уравнений, подобный аппарату, используемому в математическом анализе, но отличающийся от последнего тем, что он предназначен для формализации не непрерывных, а дискретных процессов. Такой язык дают логические исчисления, а именно: исчисление высказываний и исчисление предикатов. Однако чтобы иметь возможность эффективно решать указанные уравнения, необходимо довести эти исчисления до уровня алгебраической системы [2, с.54]. В классической линейной алгебре широко используется аппарат матриц. Но мы имеем возможность записывать только двухмерные матрицы, не говоря уже о громоздкости этой записи, возрастающей по мере увеличения размерности матриц.

*Актуальность исследования.* С учетом особенностей логической алгебры становится возможным разработать методы представления булевых матриц, сокращающие их запись, а также допускающие отсутствие ограничений на арность матриц. Эти методы изложены в [3] и в данной статье.

### 2. Методы решения

Возьмем в качестве поля логических скаляров множество  $G = \{0, 1\}$ . В качестве логических векторов рассмотрим множество конституент единицы по  $m$  переменным. Такое логическое пространство будем называть *булевым*. В то же время систему всех конечных предикатов арности  $m$ , заданных на декартовом произведении  $K = K_1 \times \dots \times K_m$ ,  $|K_i| = k_i$ , можно рассматривать как логическое пространство. Множество  $K_i$  является алфавитом, на котором задан аргумент  $x_i$  рассматриваемых предикатов. Полем логических скаляров для такого пространства может служить любая система всех конечных предикатов арности  $n < m$ . Однако алфавитами и для аргументов предикатов поля скаляров, и для аргументов предикатов множества векторов такого логического пространства должны служить одни и те же множества. Построенное таким образом пространство назовем предикатным логическим пространством или пространством  $t$ -местных предикатов над скалярным полем  $n$ -местных предикатов [3].

Рассмотрим пример логического пространства. В качестве поля логических скаляров возьмем множество  $P$  всех одноместных предикатов  $P_i(x)$ ,  $(i = 0, \dots, 3)$ , определенных на множестве  $K = \{0, 1\}$ . Множество таких предикатов задано табл. 1.

Таблица 1

x	$P_0$	$P_1$	$P_2$	$P_3$
0	0	0	1	1
1	0	1	0	1

Далее одноместные предикаты представляются строками  $P = (P(0), P(1))$ . Тогда  $P_0 = (0, 0)$ ,  $P_1 = (0, 1)$ ,  $P_2 = (1, 0)$ ,  $P_3 = (1, 1)$ . Обозначим это скалярное поле через  $p$ . Тогда, например, логическая матрица [4] будет иметь вид:

$$P_1 \begin{pmatrix} P_1 & P_2 & P_1 \\ P_0 & P_3 & P_1 \\ P_2 & P_0 & P_3 \end{pmatrix} = (0, 1) \begin{pmatrix} (0,1) & (1,0) & (0,1) \\ (0,0) & (1,1) & (0,1) \\ (1,0) & (0,0) & (1,1) \end{pmatrix} =$$

$$= \begin{pmatrix} (0,1) & (0,0) & (0,1) \\ (0,0) & (0,1) & (0,1) \\ (0,0) & (0,0) & (0,1) \end{pmatrix} = \begin{pmatrix} P_1 & P_0 & P_1 \\ P_0 & P_1 & P_1 \\ P_0 & P_0 & P_1 \end{pmatrix}.$$

В качестве пространства логических векторов [5] возьмем множество двухместных предикатов  $Q_j(x, y)$ ,  $j = 0, \dots, 15$ , определенных на декартовом произведении  $K^2 = \{0, 1\}^2$ . Множество таких векторов, заданное табл. 2, будем записывать в виде матриц

$$Q_j = \begin{pmatrix} Q_j(0,0) & Q_j(1,0) \\ Q_j(0,1) & Q_j(1,1) \end{pmatrix}.$$

Таблица 2

x	y	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>	Q <sub>8</sub>	Q <sub>9</sub>	Q <sub>10</sub>	Q <sub>11</sub>	Q <sub>12</sub>	Q <sub>13</sub>	Q <sub>14</sub>	Q <sub>15</sub>
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Согласно определению операции дизъюнкции векторов и операции умножения вектора на скаляр, для векторов такого пространства

$$Q_i \vee Q_j = \begin{pmatrix} Q_i(0,0) \vee Q_j(0,0) & Q_i(1,0) \vee Q_j(1,0) \\ Q_i(0,1) \vee Q_j(0,1) & Q_i(1,1) \vee Q_j(1,1) \end{pmatrix},$$

$$P_i Q_j = \begin{pmatrix} P_i(0)Q_j(0,0) & P_i(1)Q_j(1,0) \\ P_i(0)Q_j(0,1) & P_i(1)Q_j(1,1) \end{pmatrix}.$$

Нулевые и единичные элементы в этом случае имеют вид:  $0 = P_0$ ,  $0 = Q_0$ ,  $1 = P_3$ ,  $1 = Q_{15}$ . Обозначим это пространство через  $Q$ . Это пространство является полным. Приведем пример комбинации векторов  $Q_1$  и  $Q_5$ :

$$P_0 Q_1 \vee P_2 Q_5 = (0, 0) \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \vee (1, 0) \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = Q_4.$$

Вычислим размерность  $p$  описанного выше совершенного предикатного пространства. Множество логических скаляров для описанного предикатного пространства будет иметь мощность  $2^{k_1 k_2 \dots k_n}$ , а количество векторов, составляющих рассматриваемое предикатное пространство, будет равно  $2^{k_1 k_2 \dots k_m}$  [6]. Согласно определению, любой вектор совершенного логического пространства можно единственным образом представить в виде линейной комбинации базисных векторов. Число таких комбинаций будет равно  $(2^{k_1 k_2 \dots k_n})^p$ . С другой стороны, как было сказано выше, это число равно  $2^{k_1 k_2 \dots k_m}$ . Следовательно, имеет место равенство  $2^{k_1 k_2 \dots k_m} = (2^{k_1 k_2 \dots k_n})^p$ , откуда, логарифмируя обе части этого равен-

ства по основанию 2, находим число базисных векторов описанного выше совершенного предикатного пространства:

$$p = k_{n+1}k_{n+2} \cdots k_m. \quad (1)$$

Если же  $|K_1| = \dots = |K_m| = k$ , то

$$p = k^{m-n}. \quad (2)$$

Рассмотрим предикатное пространство  $Q$ . Алфавитом в указанном пространстве служит множество  $K^3 = \{0, 1\}^3$ . Очевидно, что  $k = 2$ ,  $n = 1$ ,  $m = 2$ . Количество логических скаляров в этом примере равно четырем, а векторов - шестнадцати. Размерность этого пространства равна  $2^{2-1} = 2$ .

В силу того, что векторы булева пространства можно понимать как множество конститuent единицы по  $m$  переменным, число векторов булева пространства равно  $2^m$ . Поле логических скаляров для булева пространства содержит два элемента: ноль и единицу. Следовательно, справедливо соотношение  $2^p = 2^m$ , где  $p$  – размерность булева пространства, т.е. количество его базисных векторов. Очевидно, что

$$p = m. \quad (3)$$

Следует отметить, что в булевом пространстве существует единственный с точностью до перестановки набор векторов, который может служить базисом выбранного пространства, тогда как в предикатном логическом пространстве это не так.

В силу равенства (3), для булевых пространств любой матрице над полем  $G = \{0, 1\}$  отвечает некоторый линейный логический оператор. Для предикатных пространств это не так. В случае таких пространств на размерность матриц, соответствующих линейным логическим операторам, накладывается следующее ограничение. Допустим, что некоторая логическая матрица  $A_{q \times p}$  над полем логических скаляров, элементами которого являются конечные предикаты арности  $n$ , представляет собой матрицу некоторого логического оператора  $A$ , переводящего векторы предикатного пространства  $W$  размерности  $p$  в векторы предикатного пространства  $V$  размерности  $q$ . В силу равенства (1), для  $p$  и  $q$  должны выполняться следующие соотношения:

$$p = k_{m_p-n} \cdots k_{m_p}, \quad (4)$$

$$q = k_{m_q-n} \cdots k_{m_q}, \quad (5)$$

где  $m_p$  и  $m_q$  – арности предикатов, представляющих собой векторы пространств  $W$  и  $V$  соответственно. Следовательно, множество матриц линейных логических операторов в предикатном пространстве является подмножеством множества всех логических матриц над предикатным скалярным полем. Мощность этого подмножества для каждого конкретных пространств прообразов  $W$  и пространства образов  $V$  равна  $(2^{k_1 k_2 \cdots k_n})^{pq}$ , где  $p$  и  $q$  – размерности пространств  $W$  и  $V$  соответственно. В случае  $k_1 = \dots = k_m = k$ , как следует из (2), соотношения (4) и (5) принимают вид

$$p = k^{m_p-n}, \quad (6)$$

$$q = k^{m_q-n}, \quad (7)$$

т.е. когда мощности алфавитов, над которыми заданы аргументы предикатов, являющихся элементами рассматриваемого логического пространства, равны, указанная выше логическая матрица  $A_{q \times p}$  является матрицей описанного линейного логического оператора в том и только в том случае, когда количество ее строк и количество ее столбцов представляют собой некоторую целую степень числа  $k$ . Мощность множества матриц линейных логических операторов в этом случае равна  $2^{k^n pq}$ .

Например, матрица

$$A = \begin{pmatrix} (0,1) & (1,0) \\ (1,1) & (0,1) \\ (1,0) & (0,0) \end{pmatrix}$$

над полем одноместных предикатов над алфавитом  $K = \{0, 1\}$  не соответствует ни одному линейному логическому оператору, так как число ее строк равно трем, а  $k = 2$ . Матрица же

$$B = \begin{pmatrix} (0,1) & (1,0) & (1,1) & (1,0) \\ (1,0) & (1,0) & (0,0) & (0,1) \end{pmatrix}$$

соответствует линейному логическому оператору  $B$ , переводящему векторы предикатного пространства  $W$  размерности 4 в векторы предикатного пространства  $V$  размерности 2. Учитывая соотношения (6) и (7), а также то, что  $k = 2$ , а  $n = 1$ , находим, что арность предикатов, представляющих собой векторы пространства образов  $V$ , равна  $m_q = 2$ , а арность предикатов, представляющих собой векторы пространства прообразов  $W$ , равна  $m_p = 3$ .

Таким образом, в случае рассмотрения матриц, представленных в бинарнопредикатном виде [7], можно утверждать, что любая такая матрица является матрицей некоторого логического оператора в силу того, что представление с помощью бинарных предикатов является формой записи булевых матриц.

### 3. Выводы

В ходе исследований, изложенных в статье, была показана возможность представления матриц линейных логических операторов по аналогии с другими формами, такими, как графическая интерпретация логических пространств и описание с помощью бинарных предикатов.

*Научная новизна:* по результатам проведенных исследований предложены различные формы представления матриц линейных логических операторов.

*Практическая значимость:* рассмотренные в статье возможные методы описания матриц линейных логических операторов позволяют упростить их компьютерное моделирование.

**Список литературы:** 1. Сигорский В.П. Математический аппарат инженера. Киев: Техника. 1975. 768с. 2. Шабанов-Кушнарченко Ю.П. Теория интеллекта. Математические средства. Харьков: Вища школа. 144с. 3. Гвоздинская Н.А., Шабанов-Кушнарченко Ю.П. Предикатные логические пространства // Сб. науч. тр. по материалам 4-й международной конференции "Теория и техника передачи, приема и обработки информации. (Новые информационные технологии)". Туапсе: ХТУРЕ. 1998. С.239. 4. О логических матрицах./ Гвоздинская Н.А., Дударь З.В., Пославский С.А., Шабанов-Кушнарченко Ю.П. // Проблемы бионики. 1998. Вып. 48. С. 12 - 22. 5. О логических пространствах./ Гвоздинская Н.А., Дударь З.В., Пославский С.А., Шабанов-Кушнарченко Ю.П. // АСУ и ПА. 1997. Вып. 106. С. 21 - 30. 6. Шабанов-Кушнарченко Ю.П. Неполные и полные логические пространства. // Проблемы бионики. 1991. Вып.46. С.10-17. 7. Гвоздинский А.Н., Якимова Н.А., Губин В.А. Представление булевых логических матриц в виде бинарных предикатов // Радиоэлектроника и информатика. 2007. Вып 2. С. 108 – 110. 8. Гвоздинский А.Н., Якимова Н.А., Губин В.А. Бинарные предикаты при описании булевых логических пространств // АСУ и ПА. 2012. Вып. 161. С. 108 - 118.

Поступила в редколлегию 12.09.2015

**Гвоздинский Анатолий Николаевич**, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЕ. Научные интересы: методы оптимизации в организационном управлении. Адрес: Украина, 61166, Харьков, пр. Ленина 14, тел. 70-21-337.

**Якимова Наталья Анатольевна**, канд. техн. наук, доцент кафедры компьютерной алгебры и дискретной математики Одесского национального университета им.И.И. Мечникова. Научные интересы: логическая алгебра, искусственный интеллект. Адрес: Украина, 65001, Одесса-001, ул.Дворянская, тел.: 8(048)7238405.

**Губин Вадим Александрович**, старший преподаватель кафедры искусственного интеллекта ХНУРЕ. Научные интересы: интеллектуальный анализ текстовых данных. Адрес: Украина, 61166, Харьков, пр. Ленина 14, тел. 70-21-337.

## **МЕТОД ПОИСКА РЕЛЕВАНТНЫХ СВЯЗЕЙ МЕЖДУ КОНЦЕПТАМИ ПРОЕКТИРУЕМЫХ ОНТОЛОГИЙ**

---

Предлагается метод определения наиболее релевантных связей между концептами формируемых онтологических моделей. Вычислительная схема метода, основанная на модификации алгоритма Гинзбурга, позволяет повысить качество автоматически создаваемых онтологий. Метод может эффективно использоваться для задач семантического поиска в системах интеллектуального анализа электронных текстов и формирования онтологических моделей предметной области.

### **1. Постановка проблемы**

Основной задачей современных систем поиска и предварительной обработки web-документов является оперативное предоставление пользователям сети Интернет необходимой информации. При этом результаты поиска не всегда оказываются удовлетворительными, так как поисковые Интернет-сервисы могут выдавать по запросам пользователей большое количество условнорелевантных web-данных, которые далеко не всегда удовлетворяют истинным интересам пользователей. Кроме того, такие результаты могут быть существенно зашумлены нерелевантными ссылками. Все это снижает эффективность получения пользователями необходимой значимой информации из сети Интернет, ресурсы которой постоянно растут. В связи с этим особенно актуальными становятся автоматические методы работы с большими объемами информации. В последнее время получили широкое распространение исследования в области автоматического синтеза онтологических моделей, позволяющих повысить эффективность систем семантического поиска по запросам пользователей (в корпусе текстов, электронных библиотеках, в сети Интернет) [1]. Актуальными также являются задачи использования онтологии как основы для спецификации и разработки программного обеспечения, поддержки общего доступа к информации, поиска информации, взаимодействия при объединении информации, создании порталов знаний, разработке пользовательского интерфейса программных систем, редакторов информации и интеллектуальных систем [2]. Качество формируемых онтологий, используемых для создания поисковых систем, во многом определяется полнотой учета в онтологической модели наиболее значимых концептов для корпуса анализируемых текстов с учетом их тематической специфики (под концептами будем в дальнейшем понимать наиболее значимые слова и словосочетания в анализируемом тексте, которые могут быть учтены в онтологической модели). В связи с этим целесообразно решить задачу формирования множества концептов будущей онтологии с учетом связей между ними. В работах [3, 4] уже были рассмотрены решения автоматического построения онтологий, в частности, методы нахождения концептов для онтологии и связей между ними. Данное исследование ставит перед собой целью усовершенствование и дополнение алгоритмов и методов автоматического синтеза онтологических моделей.

Методы нахождения концептов при автоматическом синтезе онтологий и нахождения шаблонных связей между ними (типа «часть-целое» и «отношение») рассматриваются в работе [5]. Однако результаты экспериментального исследования этих методов показали, что при поиске слов и словосочетаний, которые могут использоваться в качестве концептов, сформированное множество концептов-претендентов не всегда соответствует такому же множеству, составленному экспертом предметной области. Это приводит к тому, что некоторые важные понятия предметной области могут не попасть в автоматически создаваемую онтологию. Кроме того, в этих методах отсутствует процедура общего ранжирования по значимости списка всех концептов-претендентов, а осуществляется лишь раздельное ранжирование слов и словосочетаний, входящих в этот список.

В частности, возникают следующие проблемы:

- не всегда удается правильно найти связи между концептами;

– не всегда удастся выделить концепты, имеющие связь с наибольшим количеством других концептов;

– найденные связи между концептами будущей онтологии не всегда актуальны для конкретной предметной области. При этом не только повышается используемый объем памяти и увеличивается время на создание онтологии и обработку запросов к ней, но и избыточным становится объем онтологии, что снижает оперативность дальнейшего ее применения.

В данной статье рассматривается возможность частичного устранения перечисленных трудностей на основе комбинированного применения и модификации существующих методов определения релевантных связей между концептами формируемых онтологических моделей.

*Целью данного исследования* является модификация и программная реализация методов автоматического поиска актуальных связей между концептами проектируемой онтологии для заданной предметной области.

## **2. Установление связей между концептами проектируемой онтологии**

Выделим три основных подхода для решения задачи установления связей между концептами проектируемой онтологии:

– поиск слова-претендента на связь в онтологии и последующий подбор концептов, для которых актуальна эта связь (метод 1);

– определение для рассматриваемого концепта списка вероятных слов-претендентов на использование в качестве связи для этого концепта и последующий подбор концепта для установления связи (метод 2);

– нахождение в онтологии двух концептов, которые необходимо связать, и последующий подбор связи для данных концептов (метод 3).

Достоинства первого подхода (метод 1):

– поиск в тексте слов-связей и концептов осуществляется отдельно. Это означает, что концепт и связь не обязательно должны составлять в тексте словосочетание при поиске данной связки в тексте программы автоматического синтеза онтологии;

– возможность варьировать количество учитываемых связок «концепт-связь-концепт» с помощью настраиваемых коэффициентов (уменьшать в случае нахождения большого количества ненужной информации и увеличивать в случае недостаточного количества связей в онтологии).

Недостатки первого подхода:

– в общем множестве найденных связей между концептами присутствуют несущественные или несуществующие связи;

– некоторые важные концепты предметной области не имеют связей сформированного множества с другими концептами проектируемой онтологии.

Устранению отмеченных недостатков способствует комбинированное применение второго и третьего подходов (метод 2 и 3).

Предлагаемый ниже метод поиска связей для онтологии, основанный на таком комбинированном подходе, назовем методом главного концепта.

## **3. Метод главного концепта.**

Данный метод предполагает необходимость вычисления вероятности применения слова в качестве релевантной связки для рассматриваемого концепта.

Рассмотрим вначале свойства слов, которые в формируемой онтологии будут применяться в качестве слов-отношений, связывающих концепты в онтологии.

Задача автоматического определения таких связок является далеко не тривиальной. Рассмотрим пример определения слов связок для концептов следующего текстового фрагмента:

1) «на основе алгоритма Гинзбурга был разработан метод выделения ключевых слов»;

2) «разработанный алгоритм синтезирует функциональную модель»;

3) «осуществляется определение для концепта онтологии необходимого списка связей».

В этом примере словами-связками между понятиями являются соответственно слова «разработан», «синтезирует», «определение». Здесь в качестве слов-связок могут приме-

няться как слова специфичные для рассматриваемой предметной области, так и достаточно общие, которые могут присутствовать в любом тексте. Можно отметить, что слово-связка вероятнее всего будет находиться в тексте между понятиями, которые оно связывает. Вследствие этого целесообразно определить степень специфичности претендента на слово-связку в контексте понятия, которое будет связывать данное слово-связка. Для решения этой задачи предлагается использовать алгоритм Гинзбурга [6]. В соответствии с этим алгоритмом, если слово-связка входит в контекст леммы-понятия в рамках рассматриваемого текста, считают, что оно специфично в контексте данного понятия. Введем понятие тройки элементов, используемых для реализации процедуры предварительного отбора наиболее релевантных связей для проектируемой онтологии. К элементам такой тройки отнесем: слово, обозначающее связь между двумя концептами ( $L_1$ ), и собственно два концепта ( $W_1$  и  $W_2$ ), каждый из которых может быть представлен одним словом либо словосочетанием. Таким образом, тройку можно представить в виде: «слово №1, связь, слово №2»:

$$W_1 \leftrightarrow L_1 \leftrightarrow W_2. \quad (1)$$

Отметим, что если концепт представлен словосочетанием, то в тройку вносится главное слово словосочетания.

Выделим четыре возможных варианта представления любой тройки в зависимости от уровня специфичности слова-связки по отношению к понятиям:

$$W_1 \xleftarrow{F_{(W_1,L_1)} \uparrow} L_1 \xleftarrow{F_{(W_2,L_1)} \uparrow} W_2; \quad (2)$$

$$W_1 \xleftarrow{F_{(W_1,L_1)} \uparrow} L_1 \xleftarrow{F_{(W_2,L_1)} \downarrow} W_2; \quad (3)$$

$$W_1 \xleftarrow{F_{(W_1,L_1)} \downarrow} L_1 \xleftarrow{F_{(W_2,L_1)} \uparrow} W_2; \quad (4)$$

$$W_1 \xleftarrow{F_{(W_1,L_1)} \downarrow} L_1 \xleftarrow{F_{(W_2,L_1)} \downarrow} W_2, \quad (5)$$

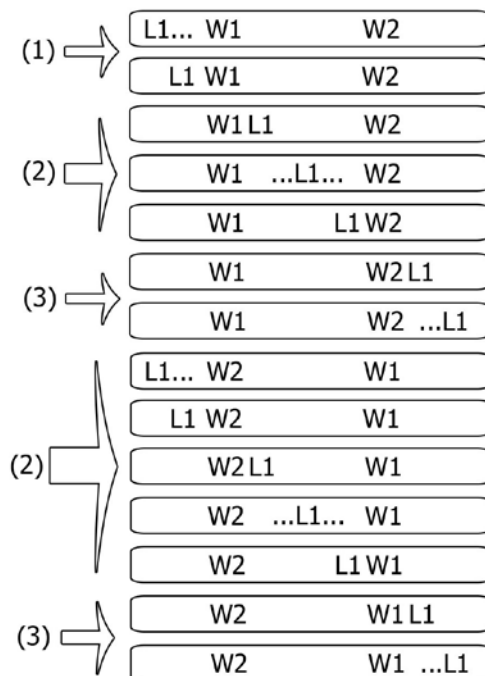
где символ  $F_{(W_i,L_1)} \uparrow$  означает, что слово-связка  $L_1$  специфична для слова  $W_i$ , а  $F_{(W_i,L_1)} \downarrow$  означает, что слово-связка  $L_1$  не специфична для слова  $W_i$ .

На основе статистического анализа текстов рассматриваемой предметной области могут быть определены коэффициенты вероятности принадлежности определенной тройки к одному из вариантов ее представления: (2)-(4) или (5). Например, для корпуса текстов из электронной библиотеки методических указаний Харьковского национального университета радиоэлектроники по технической тематике было определено количество троек, принадлежащих к одному из четырех типов, и рассчитаны (как среднее арифметическое по всей выборке текстов) соответствующие вероятности:  $p_1 = 0,25$ ;  $p_2 = 0,15$ ;  $p_3 = 0,1$ ;  $p_4 = 0,5$  [3].

На основании полученных значений  $p_1, p_2, p_3, p_4$  определим вероятности выбора слова в качестве связки в зависимости от его положения в предложении по отношению к концептам. Рассмотрим варианты положения слова-связки в предложении относительно концептов, которые оно связывает. Назовем «нормальным порядком» расположения концептов в тексте, если первый концепт располагается в тексте раньше, чем второй, и «обратным порядком», если второй концепт располагается в тексте раньше, чем первый. В зависимости от положения слова-связки в предложении относительно концептов рассматриваемую тройку можно отнести к одной из трех возможных групп (рисунок). Первая группа содержит тройки, в которых слово-связка находится в предложении между первым и вторым концептами, а концепты расположены в нормальном порядке. Вторая группа содержит тройки, в которых слово-связка находится в предложении между первым и вторым концептами, а концепты расположены в обратном порядке. К этой группе отнесем также тройки, в которых слово-связка расположено в тексте раньше, чем концепты. Третья группа содержит тройки, в которых слово-связка расположено в тексте после концептов.

На основе значений  $p_1, p_2, p_3, p_4$  можно определить коэффициенты  $k$ , соответствующие вероятности автоматического выбора определенной тройки в качестве актуальной для проектируемой онтологии. Коэффициенты  $k$  представляют собой отношение числа троек

определенного типа к общему числу троек, актуальных для онтологии заданной предметной области. Для рассмотренного выше примера получены следующие значения этого коэффициента в зависимости от варианта (группы) положения слова-связки в предложении относительно концептов: группа (1) –  $k = 0,7$ ; группа (2) –  $k = 0,2$ ; группа (3) –  $k = 0,1$ .



Варианты положения слова-связки  $L_1$  в тексте относительно концептов  $W_1$  и  $W_2$

При принятии решения о занесении той или иной тройки в проектируемую онтологию, кроме расположения элементов тройки, необходимо учитывать наличие слов между ними и их количество. Очевидно, что целесообразнее вносить в онтологию тройку, элементы которой следуют непосредственно друг за другом, чем тройку, между концептами и связкой которой находятся фрагменты предложения.

Назовем расстоянием между элементами тройки количество слов, которые находятся в предложении между двумя любыми элементами тройки. Обозначим через  $N$  расстояние в предложении между двумя концептами  $W_1$  и  $W_2$  рассматриваемой тройки.

Тогда вероятность актуальности рассматриваемой тройки в зависимости от положения ее элементов в предложении можно определить следующим образом:

$$P_{\text{place}} = \frac{k * ( (|m - n| / \min(n, m) + 2) + 1 )}{n + m + 1}, \quad (6)$$

где  $n$  – расстояние от  $L_1$  до  $W_1$ ,  $n = N$ , если между  $L_1$  и  $W_1$  находится  $W_2$ ;  $m$  – расстояние от  $L_1$  до  $W_2$ ,  $m = N$ , если между  $L_1$  и  $W_2$  находится  $W_1$ .

В соответствии с (6), чем больше расстояние между словом-связкой и концептами в тройке, тем меньше вероятность ее актуальности для проектируемой онтологии. Также необходимо отметить, что приведенная формула учитывает приоритет троек, у которых расстояние слова-связки хотя бы с одним из концептов является намного меньше среднего значения такого расстояния для всей совокупности рассматриваемых концептов.

Предлагаемый метод главного концепта имеет ряд преимуществ. В частности, его можно применять для поиска связей в онтологии, в которой до этого не была определена ни одна связь. Дополнительным преимуществом метода является возможность задавать здесь список концептов, для которых необходимо найти связь (например, всех концептов, имеющих в онтологии в данный момент). Это позволяет определить максимально возможное количество актуальных (релевантных) связей для проектируемой онтологии. Сле-

дует отметить, что целесообразно искать слово-связку для выбранного концепта/понятия только в тех предложениях, где встречается собственно сам этот концепт.

Алгоритм установления связей в онтологии по методу главного концепта можно представить набором следующих действий:

- выбор концепта/понятия и нормализация его до одного слова ( $W_1$ ), для которого следует сформировать тройку в проектируемой онтологии;

- определение множества слов  $M \uparrow (W_1)$ , входящих в контекстное множество данного концепта  $W_1$  (из множества всех слов в предложениях, где присутствует данный концепт с понятием  $M(W_1)$ ), а также множества слов  $M \downarrow (W_1)$ , не входящих в контекстное множество данного концепта (по алгоритму Гинзбурга [6]);

- определение наиболее вероятного типа связи ( $F_{(W_1, L_1)} \uparrow$  или  $F_{(W_1, L_1)} \downarrow$ ) между концептом  $W_1$  и предполагаемым словом-связкой  $L_1$ ;

- определение множества  $M(L_1)$ , состоящего из претендентов на слова-связки, удовлетворяющих установленному типу связи ( $M(L_1)$  принимается как  $M \uparrow (W_1)$  или как  $M \downarrow (W_1)$ );

- определение для каждого  $L_i$  (из множества  $M(L_1)$ ) множества слов  $M \uparrow (W_2)$ , входящих в контекстное множество данного слова  $L_i$  (из множества всех слов, входящих в одно предложение с данным словом  $L_i$  и данным словом  $W_1$ , во всех предложениях, где присутствуют  $L_i$  и  $W_1$ ), а также множества слов,  $M \downarrow (W_2)$  не входящих в контекстное множество данного концепта (по алгоритму Гинзбурга [6]);

- определение множеств  $M_i(W_2)$ , состоящих из претендентов на концепт, связываемый с концептом  $W_1$  при помощи слова-связки  $L_i$ , удовлетворяющих установленному типу связи (для каждого  $L_i$  из множества  $M(L_1)$ );

- определение наиболее вероятного типа связи ( $F_{(W_2, L_1)} \uparrow$  или  $F_{(W_2, L_1)} \downarrow$ ) между будущим словом-связкой  $L_1$  и концептом  $W_2$  (для каждого  $L_i$  из множества  $M(L_1)$ );

- включение в онтологию наиболее вероятной связки из множества  $M(T_i)$  возможных вариантов троек.

Наиболее вероятный тип связи  $F_{(W_1, L_1)}$  определяется по следующим зависимостям:

$$F_{(W_1, L_1)} = \begin{cases} F_{(W_1, L_1)} \uparrow, & \text{if } P_{L_1} > P_{L_2} \\ F_{(W_1, L_1)} \downarrow, & \text{if } P_{L_1} < P_{L_2} \end{cases}, \quad (7)$$

$$P_{L_1} = \frac{|M \uparrow (W_1)|}{N_{\text{all}}} * (p_1 + p_2), \quad (8)$$

$$P_{L_2} = \frac{|M \downarrow (W_1)|}{N_{\text{all}}} * (p_3 + p_4), \quad (9)$$

где  $N_{\text{all}}$  – мощность объединения множеств  $M \uparrow (W_1)$  и  $M \downarrow (W_1)$ , определяемая количеством всех слов в предложениях с  $W_1$ .  $p_1, p_2, p_3, p_4$ .

В (8), (9) вероятности  $p_i$  суммируются, так как типы троек, соответствующие вероятностям  $p_1$  и  $p_2$ , удовлетворяют условию типа связи  $F_{(W_1, L_1)} \uparrow$ , а  $p_3$  и  $p_4$  – условию типа связи  $F_{(W_2, L_1)} \downarrow$ .

Наиболее вероятный тип связи  $F_{(W_2, L_1)}$  определяется по следующим зависимостям:

$$F_{(W_2, L_1)} = \begin{cases} F_{(W_2, L_1)} \uparrow, & \text{if } P_{L_3} > P_{L_4} \\ F_{(W_2, L_1)} \downarrow, & \text{if } P_{L_3} < P_{L_4} \end{cases}, \quad (10)$$

$$P_{L_3} = \begin{cases} \frac{|M \uparrow (W_2)|}{N_{all2}} * p_1 * (N_{all2} - N_{L_i}), & \text{if } F_{(W_1, L_1)} = F_{(W_1, L_1)} \uparrow \\ \frac{|M \uparrow (W_2)|}{N_{all2}} * p_2 * (N_{all2} - N_{L_i}), & \text{if } F_{(W_1, L_1)} = F_{(W_1, L_1)} \downarrow, \end{cases} \quad (11)$$

$$P_{L_4} = \begin{cases} \frac{|M \downarrow (W_2)|}{N_{all2}} * p_3 * (N_{all2} - N_{L_i}), & \text{if } F_{(W_1, L_1)} = F_{(W_1, L_1)} \uparrow \\ \frac{|M \downarrow (W_2)|}{N_{all2}} * p_4 * (N_{all2} - N_{L_i}), & \text{if } F_{(W_1, L_1)} = F_{(W_1, L_1)} \downarrow, \end{cases} \quad (12)$$

где  $N_{all2}$  – мощность объединения множеств  $M \uparrow (W_2)$  и  $M \downarrow (W_2)$ ;  $N_{L_i}$  – количество повторений  $L_i$ .

В формулах (11) и (12) вероятности  $P_{L_3}$  и  $P_{L_4}$  умножаются на разность общего количества слов в предложениях, где присутствуют элементы рассматриваемой тройки и вхождения в эти предложения слова-связки  $L_i$ . Таким образом, чем чаще в предложении встречается определенное слово, тем меньше вероятность его выбора в качестве слова-связки, так как маловероятно, что одно и то же слово будет выполнять в предложении и роль слова-связки для двух понятий, и просто встречаться в предложении в каких-либо других контекстах.

На заключительном этапе алгоритма определяется множество  $M(T_i)$  – множество троек, для которых определены типы связей  $F_{(W_1, L_1)}$  и  $F_{(W_2, L_1)}$  ((2), (3), (4) или (5) соответственно). При этом предлагается ранжировать элементы из данного множества в соответствии со значениями вероятностей их выбора в качестве троек, актуальных для проектируемой онтологии. Вероятность выбора тройки  $T_i$  в качестве актуальной для онтологии  $P_{T_i}$  рассчитывается по следующей зависимости:

$$P_{T_i} = \begin{cases} \frac{p_{place} * f_{(W_1, L_1)} * f_{(W_2, L_1)} * f_{(W_1, W_2)}, & \text{if } T_i \subset (2), \\ \frac{p_{place} * f_{(W_1, L_1)} * f_{(W_1, W_2)}}{f_{(W_2, L_1)}}, & \text{if } T_i \subset (3), \\ \frac{p_{place} * f_{(W_2, L_1)} * f_{(W_1, W_2)}}{f_{(W_1, L_1)}}, & \text{if } T_i \subset (4), \\ \frac{p_{place} * f_{(W_1, W_2)}}{f_{(W_1, L_1)} * f_{(W_2, L_1)}}, & \text{if } T_i \subset (5) \end{cases} \quad (13)$$

где  $p_{place}$  – вероятность, рассчитываемая по формуле (6);  $f_{(W_1, L_1)}$  – сила связи, рассчитанная для  $W_1$  и  $L_1$  из тройки  $T_i$  по алгоритму, описанному в [3];  $f_{(W_2, L_1)}$  – сила связи, рассчитанная для  $W_2$  и  $L_1$  из тройки  $T_i$  по алгоритму, описанному в [3];  $f_{(W_1, W_2)}$  – сила связи, рассчитанная для  $W_1$  и  $W_2$  из тройки  $T_i$  по алгоритму, описанному в [3]; (2, 3, 4, 5) – один из четырех определенных типов тройки  $T_i$ .

Эта формула позволяет учесть (в зависимости от типа рассматриваемой тройки), насколько сила семантической связи между элементами тройки влияет на вероятность ее выбора в качестве актуальной для проектирующейся онтологии. Если тип связи между двумя элементами определен как  $F_{(W_1, L_1)} \uparrow$ , то с возрастанием силы связи между элементами тройки  $W_i$  и  $L_1$  возрастает вероятность актуальности данной тройки для онтологии. Если же тип связи между двумя элементами определен как

$F_{(W_i, L_1)} \downarrow$ , то чем больше сила связи между элементами тройки  $W_i$  и  $L_1$ , тем меньше вероятность актуальности данной тройки для онтологии. Также следует отметить, что чем больше сила связи между концептами тройки, тем тройка актуальнее для проектируемой онтологии. Кроме того, при оценивании связей в онтологии по методу главного концепта можно найти или одну тройку, для которой вероятность актуальности для онтологии наиболее высока, или же найти множество троек, для которых вероятность  $P_{T_i}$  выше либо равна  $\min(P_{T_i})$ , и считать, что все тройки, входящие в это множество, актуальны для проектируемой онтологии:

$$\min(P_{T_i}) = K_1 * \max(P_{T_i}), \quad (14)$$

где  $K_1$  – настраиваемый коэффициент, позволяющий исключить из рассмотрения тройки, заведомо неактуальные для проектируемой онтологии.

#### 4. Оценка эффективности разработанного метода

По предложенному методу был разработан программный модуль ««Concept-Ont-M», который может эффективно использоваться для задач семантического поиска в системах анализа электронных текстов и автоматического создания онтологий. Проведенные экспериментальные исследования показали, что метод главного концепта в целом работает гораздо эффективнее, чем методы, основанные на поиске отдельных связей. Оценка эффективности проводилась по двум параметрам:  $R$  – точность поиска связей (отношение правильно найденных связей к общему количеству найденных связей);  $P$  – полнота поиска связей (отношение правильно найденных связей к общему количеству связей, выявленных экспертом). Результаты экспериментальных исследований (для корпуса текстов из электронной библиотеки методических указаний Харьковского национального университета радиоэлектроники по технической тематике) для метода поиска отдельных связей и метода главного концепта значения  $R$  составляют 57 и 78% соответственно; значения  $P$  – 78 и 82% соответственно.

#### 5. Выводы и перспективы дальнейших исследований

Проведенные исследования позволяют сделать вывод, что важным этапом автоматического построения онтологий является формирование релевантных связей между концептами. Модификация и программная реализация метода нахождения таких связей с учетом расположения элементов «концепт-связка» в тексте позволили повысить возможности автоматического создания онтологий. В частности, предложенный метод можно применять для поиска связей в онтологии, в которой до этого не была определена ни одна связь. При этом особое внимание следует уделить задачам ранжирования однословных/многословных концептов и выявления связей типа «отношения» между ними. Научная новизна предложенного метода состоит в возможности определения степени специфичности претендента на слово-связку в контексте леммы-понятия в рамках анализируемого текста. При проведении дальнейших исследований целесообразно усовершенствовать предложенный метод, дополнив его анализом более сложных типов связей в онтологической модели.

**Список литературы:** 1. Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Ч. 3) / В.Ф. Хорошевский // Искусственный интеллект и принятие решений. 2011. № 2. С. 15–36. 2. Ландэ Д.В. Интернетика: Навигация в сложных сетях: модели и алгоритмы / Д.В. Ландэ, А.А. Снарский, И.В. Безсуднов. М.: Либроком, 2009. 264 с. 3. Чалая Л.Э., Формирование множества связанных концептов для автоматического синтеза онтологий [Текст] / Л.Э. Чалая, А.В. Чижевский // International Journal “Information Theories and Applications”. Vol. 21, Number 3. 2014. P. 203 – 212. 4. Зябрев И.Н., Пожарков О.В., Пожаркова И.Н. Использование спектральных характеристик лексем для улучшения поисковых алгоритмов // Труды РОМИП 2010. Казань: Казанский ун-т. С. 40–48. 5. Воронина И.Е. Алгоритмы определения семантической близости по их окружению в тексте / И.Е. Воронина, А.А. Кретов, И.В. Попова // Вестник ВГУ: системный анализ и информационные технологии. 2011. № 2. С. 15–36. 6. Гинзбург Е. Л. Идиоглоссы: проблемы выявления и изучения контекста / Е. Л. Гинзбург // Семантика языковых единиц: Доклады VI Международной конференции. Т. I., М., 1998. С. 26–28.

Поступила в редколлегию 13.09.2015

**Чалая Лариса Эрнестовна**, канд. техн. наук, доцент кафедры ИИ ХНУРЭ. Научные интересы: системы и методы информационного поиска, обработка естественно-языковой информации. Увлечения и хобби: чтение. Адрес: Украина, Харьков, 61140, пр. Гагарина, 78, кв.48, тел. 050-636-06-03.

**Чижевский Антон Валерьевич**, аспирант каф. ИИ ХНУРЭ. Научные интересы: онтологический инжиниринг, обработка естественно-языковой информации. Увлечения и хобби: профессиональный туризм, программирование. Адрес: Украина, 61009, Харьков, ул. Вокзальная, 10, кв.80, тел. 097-527-96-51.

**Волощук Елена Борисовна**, канд. техн. наук, доцент каф. ИИ ХНУРЭ. Научные интересы: моделирование предметных областей, объектно-реляционные базы данных, обработка естественно-языковой информации. Увлечения и хобби: искусство движения. Адрес: Украина, 61143, Харьков, ул. Невельского, 14/2, тел. 05003016172.

## **СИНТЕЗ И АНАЛИЗ «КВАНТОВЫХ» МОДЕЛЕЙ ЦИФРОВЫХ СИСТЕМ**

---

Предлагается «квантовый» процессор для моделирования и верификации цифровых систем, основанный на транзакциях между адресуемыми компонентами памяти для реализации любой функциональности. Описывается новый подход к синтезу и анализу цифровых систем, использующий векторную форму (квант) задания комбинационных и последовательностных структур для их имплементации в элементы памяти, что существенно отличается от общепринятой теории проектирования дискретных устройств на основе таблиц истинности компонентов. Используются квантовые или кубитные структуры данных [1-5] в вычислительных процессах современных компьютеров в целях повышения быстродействия анализа цифровых систем и уменьшения объемов памяти на основе унарного кодирования состояний входных, внутренних и выходных переменных и имплементации кубитных векторов в элементы памяти FPGA, реализующих комбинационные и последовательностные примитивы. Внедрение квантовых *memory-based-only* моделей описания цифровых компонентов в практику проектирования вычислительных систем непосредственно влияет на увеличение выхода годной продукции, повышение надежности компьютерных изделий, снижение стоимости проектирования и изготовления, а также обеспечивает автономное восстановление работоспособности в режиме *remote & online* без участия человека.

### **1. Введение**

Цель – существенное улучшение качества и надежности вычислительных устройств за счет адресуемости схемных элементов, позволяющей выполнять *online* ремонт, а также повышение быстродействия методов моделирования, тестирования и верификации сложных цифровых систем, благодаря уменьшению размерности моделей функциональных примитивов и адресной реализации всех компонентов структур данных.

Задачи: 1) Разработка автоматной модели квантового процессора. 2) Синтез кубитных моделей цифровых примитивов: логика, триггеры, регистры и счетчики. 3) Синтез и анализ квантовых моделей цифровых схем. 4) Моделирование цифровых устройств на основе использования квантовых векторов описания примитивов.

Мотивация и состояние вопроса: 1) Современная система на кристалле содержит 94% памяти и лишь 6% логики, которая доставляет более 90% проблем, связанных с верификацией, тестированием, диагностированием и восстановлением работоспособности [6-7]. Конечно, быстродействие логических схем на порядок выше, чем у памяти, однако большая доля вычислительных процессов приходится на обмен информацией в структурах памяти. Поэтому преимущества комбинационной логики в реальных вычислительных системах обработки больших данных компенсируются большими временными затратами (порядка 90%), связанными с транзакциями в памяти. 2) Реализация процессора только на основе использования элементов памяти делает его однородным по структуре и типам функциональных примитивов, что доставляет очевидные технологические удобства процессам проектирования, производства и эксплуатации, включая верификацию, встроенные тестирование и диагностирование, а главное – ремонт в режиме *online* в результате использования на кристалле универсальных адресуемых *space*-компонентов памяти [6-7]. 3) Моделирование в процессе верификации проектируемых вычислителей на основе адресных моделей компонентов делает данную процедуру технологически простой за счет регулярных структур данных и применения единственной операции транзакции на элементах памяти, а также более быстродействующей, благодаря возможности параллельной квантоподобной обработки больших массивов однотиповой памяти [3-5, 8, 11, 12]. 4) Энергопотребление при замене логики на элементы памяти возрастает на несколько процентов, что на самом деле будет платой за перечисленные выше существенные преимущества, связанные, в конечном счете, с увеличением выхода годной продукции, повышением

надежности вычислительных изделий, снижением стоимости проектирования и изготовления, а также автономным восстановлением работоспособности в режиме remote & online, без участия человека. Однако энергосберегающие решения по вычислительным процессам на памяти [9-10, 13-14] дают основания полагать, что такого проигрыша вообще не будет.

## 2. Квантовые или кубитные структуры данных

На рынке электронных технологий существует конкуренция между базами имплементации идеи [1-4,12]: 1) Гибкая (мягкая) реализация проекта связана с синтезом интерпретативной модели программной формы функциональности или в аппаратном исполнении программируемых логических устройств на основе FPGA, CPLD; преимущества заключаются в технологичности модификации проекта, недостатки – в невысоком быстродействии функционирования цифровой системы. 2) Жесткая реализация ориентирована на использование компилятивных моделей при разработке программных приложений или на имплементацию проекта в кристаллы VLSI [6-7,13-14]. Преимущества и недостатки жесткой реализации инверсны по отношению к мягкому исполнению проектов: высокое быстродействие и невозможность модификации. С учетом изложенных базовых вариантов реализации идеи предлагаются квантовые структуры данных, ориентированные на повышение быстродействия гибких моделей программного или аппаратного исполнения проекта, а также на возможность online ремонта в процессе эксплуатации.

Квантовые структуры описания цифровых систем. Кубит (n-кубит) есть векторная форма унитарного (унарного) кодирования универсума из n примитивов для задания булеа-

на состояний  $2^{2^n}$  с помощью  $2^n$  двоичных переменных. Например, если  $n=2$ , то 2-кубит задает 16 состояний с помощью четырех переменных. Если  $n=1$ , то кубит задает четыре состояния на универсуме из двух примитивов (10) и (01) с помощью двух двоичных переменных (00,01,10,11) [3,12]. При этом допускается суперпозиция (одновременное суще-

ствование) в векторе  $2^n$  состояний, обозначенных примитивами. Кубит (n-кубит) дает возможность использовать параллельные логические операции вместо поэлементных теоретико-множественных для существенного ускорения процессов анализа дискретных систем.

Далее кубит отождествляется с n-кубитом или двоичным вектором, если это не мешает пониманию излагаемого материала. Поскольку квантовые вычисления связаны с анализом кубитных структур данных, то далее будем применять определение «квантовый» для идентификации технологий, использующих три свойства квантовой механики: параллелизм обработки (двоичных векторов), суперпозицию состояний и их перепутывание. Синонимами кубита при задании двоичного вектора описания логической функции являются: Q-покрытие, Q-вектор, квантовый вектор [3-4,12] как унифицированная векторная форма суперпозиционного задания выходных состояний, соответствующих адресным кодам входных переменных логического элемента.

Кубит в цифровой системе используется в качестве формы задания структурного примитива, инвариантной к технологиям реализации функциональности (hardware, software). Более того, синтез цифровых систем на основе кубитных структур не привязан жестко к теореме Поста, определяющей пять условий (классов) существования функционально полного базиса. На предлагаемом уровне абстракции n-кубит дает более широкие возможности для векторного задания любой n-входовой функции из булеана мощностью  $|B(A)| = 2^{2^n}$ , которое непременно содержит все функциональности, удовлетворяющие пяти классам теоремы Поста. 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  $k = 2^n$ .

## 3. Синтез квант-вектора комбинационной схемы

Кубит (квант) комбинационной схемы есть вектор состояний выхода на упорядоченном множестве всех входных слов, отождествляемых с адресами ячеек памяти вектора.

Синтез Q-вектора (покрытия) схемной структуры (без таблиц истинности логических элементов) на основе примитивов, заданных Q-векторами, сводится к получению обобщенного кубит-вектора путем декартова выполнения логической операции над разрядами кубитных векторов. Декартова процедура для двух четырехразрядных кубитов, которые суперпозиционируются логической операцией *or* (*and*, *xor*), представлена в следующей таблице:

, ,	b(0)				b(1)				b(2)				b(3)			
a(0)	c(0)	a(0)	b(0)	c(1)	a(0)	b(1)	c(2)	a(0)	b(2)	c(3)	a(0)	b(3)	c(4)	a(0)	b(3)	
a(1)	c(4)	a(1)	b(0)	c(5)	a(1)	b(1)	c(6)	a(1)	b(2)	c(7)	a(1)	b(3)	c(8)	a(1)	b(3)	
a(2)	c(8)	a(2)	b(0)	c(9)	a(2)	b(1)	c(10)	a(2)	b(2)	c(11)	a(2)	b(3)	c(12)	a(2)	b(3)	
a(3)	c(12)	a(3)	b(0)	c(13)	a(3)	b(1)	c(14)	a(3)	b(2)	c(15)	a(3)	b(3)	c(16)	a(3)	b(3)	

Примеры, использующие логические суперпозиции двух кубитов для получения Q-векторов схемных структур

$$c_1 ( (a_1 ( a_2 ) ) ( b_1 ( b_2 ) ), c_2 ( (a_1 ( a_2 ) ) ( b_1 ( b_2 ) ), c_3 ( (a_1 ( a_2 ) ) ( b_1 ( b_2 ) ),$$

представлены следующей таблицей:

	a( <i>and</i> )		0001
	b( <i>or</i> )		0111
c <sub>1</sub>	a( <i>and</i> )	b( <i>or</i> )	0111 0111 0111 1111
c <sub>2</sub>	a( <i>and</i> )	b( <i>or</i> )	0000 0000 0000 0111
c <sub>3</sub>	a( <i>and</i> )	b( <i>or</i> )	0111 0111 0111 1000

Здесь построены Q-покрытия трех схем, состоящих из трех элементов каждая, где два логических примитива суперпозиционно объединяются третьим элементом (*or*, *and*, *xor*). В результате получаются три вектора, каждый из которых имеет размерность в 16 бит. Вычислительная сложность процедуры синтеза Q-покрытия комбинационной схемы равна

$$\text{произведению длин Q-векторов } p \text{ примитивов, входящих в нее: } \left( \prod_{i=1}^p \text{card}(Q_i) \right).$$

Более сложной представляется проблема синтеза Q-покрытия схемы, входные линии (сходящиеся разветвления) которой имеют гальванические или проводные соединения (здесь по переменной  $a_2$ ):  $c ( (a_1 ( a_2 ) ) ( a_2 ( a_3 ) )$ . В данном случае после синтеза Q-покрытия схемы необходимо выполнить его верификацию относительно существования противоречивых адресов на переменных  $a_2$  в целях минимизации Q-вектора путем последующего исключения упомянутых адресов из рассмотрения, что уменьшает размерность Q-покрытия до  $\text{card}(Q) ( 2^q$  координат, где  $q$  – общее число входных переменных схемы:

Q	0111 0111 0111 1111	Q	0111 0111 0111 1111	Q	0111 0111	Q	0111 0111
a <sub>1</sub>	0000 0000 1111 1111	a <sub>1</sub>	0000 0000 1111 1111	a <sub>1</sub>	0000 1111	a <sub>1</sub>	0000 1111
a <sub>2</sub>	0000 1111 0000 1111	a <sub>2</sub>	00xx xx11 00xx xx11	a <sub>2</sub>	0011 0011	a <sub>2</sub>	0011 0011
a <sub>2</sub>	0011 0011 0011 0011	a <sub>2</sub>	00xx xx11 00xx xx11	a <sub>2</sub>	0011 0011	a <sub>2</sub>	0011 0011
a <sub>3</sub>	0101 0101 0101 0101	a <sub>3</sub>	0101 0101 0101 0101	a <sub>3</sub>	0101 0101	a <sub>3</sub>	0101 0101

Процедура синтеза Q-покрытия: 1) строится таблица соответствия адресов разрядам Q-вектора схемы; 2) далее противоречивые координаты по двум строкам  $a_2$  отмечаются символами *x*; 3) затем все столбцы с данными символами исключаются из таблицы; 4) после этого получаются две идентичные строки  $a_2$ , которые объединяются в одну; 5) это дает в результате Q-вектор комбинационной схемы, но уже существенно меньшей размерности. Преимущества предложенного Q-метода синтеза вычислительных устройств заключаются в компактности их описания Q-векторами и в высоком быстродействии адресного моделирования логических элементов, также создают условия для рыночно привлекательной «квантовой» теории проектирования цифровых систем на кристаллах, использующей векторно-кубитную форму задания структурных компонентов.

#### 4. Минимизация квант-вектора схемы

Синтез квант-вектора схемы по Q-покрытиям компонентов связан с минимизацией или уменьшением размерности Q-вектора путем исключения несущественных переменных. Как правило, существенность зависит от гальванических соединений входных и внутренних линий цифрового устройства, которые накладывают ограничения, связанные с противоречивостью сигналов на линиях связи. Поэтому правило минимизации адресного пространства заключается в устранении адресных кодов, которые создают противоречия по соединенным переменным. Пусть имеется Q-вектор схемы и его адресное пространство, где переменные b,c,d (a,b,c) соединены гальванически. Ниже приведены таблицы преобразования или минимизации адресного пространства в целях получения уменьшенного Q-вектора:

$$\begin{array}{|c|c|c|c|} \hline Q & 0111 & 0111 & 0111 & 1111 \\ \hline a & 0000 & 0000 & 1111 & 1111 \\ \hline b & 0000 & 1111 & 0000 & 1111 \\ \hline c & 0011 & 0011 & 0011 & 0011 \\ \hline d & 0101 & 0101 & 0101 & 0101 \\ \hline \end{array} \quad \begin{array}{|c|c|c|c|} \hline Q & 0xxx & xxx1 & 0xxx & xxx1 \\ \hline a & 0000 & 0000 & 1111 & 1111 \\ \hline b & 0xxx & xxx1 & 0xxx & xxx1 \\ \hline c & 0xxx & xxx1 & 0xxx & xxx1 \\ \hline d & 0xxx & xxx1 & 0xxx & xxx1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline Q & 0101 \\ \hline a & 0011 \\ \hline b & 0101 \\ \hline \end{array} =$$

$$= \begin{array}{|c|c|c|c|} \hline Q & 0111 & 0111 & 0111 & 1111 \\ \hline a & 00xx & xxxx & xxxx & xx11 \\ \hline b & 00xx & xxxx & xxxx & xx11 \\ \hline c & 00xx & xxxx & xxxx & xx11 \\ \hline d & 0101 & 0101 & 0101 & 0101 \\ \hline \end{array} = \begin{array}{|c|c|} \hline Q & 0111 \\ \hline a & 0011 \\ \hline d & 0101 \\ \hline \end{array}$$

Во-первых, здесь следует отметить, что в таблицах наблюдается зеркальная осевая симметрия с инверсией сигналов на координатах адресного пространства, которая создает свойство, описываемое следующим выражением:  $L(R(1(L_{ij}(R_{ij}(1$ . Данное обстоятельство нужно использовать для уменьшения размерности анализируемого пространства в два раза и соответствующего снижения вычислительной сложности задачи синтеза квантовой вектор-функциональности цифровой схемы. Во-вторых, количество различных вариантов взаимодействий на q входных переменных, связанных с гальваническим соединением различных сочетаний входных линий, определяется функциональной зависимостью, граничные значения которой находятся в интервале:  $\text{card}(Q) \in [2^q(3^q)]$ . Тем не менее, имеется эффективная процедура для минимизации размерности Q-вектора путем выявления противоречий в кодах-столбцах, на координатах  $(A_{ij})$ , соответствующих гальванически связанным w-переменным по j-параметру. Такую процедуру достаточно выполнить на половине адресного пространства  $\text{card}(Q) \in (2^q/2)$ , а остальная часть противоречивых столбцов удаляется в соответствии с зеркальным отображением номеров тех столбцов, которые были удалены из первой половины таблицы кодов адресов:

$$\{Q_i, Q_{2^q(i)}\} \left( \left( \left( \bigwedge_{j(1}^w A_{ij} \right) \left( \bigwedge_{j(1}^w A_{ij} \right) \right) \right) \left( 1, i \in (2^q/2) \right)$$

Если в столбце  $A_i$  на группе из w связанных переменных зафиксировано, что конъюнкция их состояний равна нулю, а дизъюнкция имеет значение единицы, то i-столбец и его зеркальное отображение  $2^q(i)$  удаляются из адресного пространства A. Это автоматически приводит к исключению из Q-вектора двух полученных (i-координат (в таблицах обозначены символами x), соответствующих данным столбцам.

Естественно, что также наблюдается симметрия пространства векторов-расстояний по Хэммингу, полученных путем хог-взаимодействия между соседними строками таблицы адресного пространства, для которых суперпозиция левой и правой частей дает результат  $L(R(0(L_{ij}(R_{ij}(0$ :

$$\begin{array}{|c|c|c|c|} \hline Q & 0111 & 0111 & 0111 & 1111 \\ \hline a & b & 0000 & 1111 & 1111 & 0000 \\ \hline b & c & 0011 & 1100 & 0011 & 1100 \\ \hline c & d & 0110 & 0110 & 0110 & 0110 \\ \hline d & a & 0101 & 0101 & 1010 & 1010 \\ \hline \end{array} \quad \square (L, R); (L \square R) \square \quad \begin{array}{|c|c|c|c|} \hline Q & 0111 & 0111 & 0111 & 1111 \\ \hline a & b & 0000 & 0000 & \\ \hline b & c & 0000 & 0000 & \\ \hline c & d & 0000 & 0000 & \\ \hline d & a & 0000 & 0000 & \\ \hline \end{array} \quad \square (L \square \bar{R})$$

Целесообразно ли минимизировать логическую функцию, описанную квант-вектором? Ответ: минимизация Q-векторов для получения нормальных или скобочных форм не имеет практического значения, существенно только уменьшение размерности вектора функционального описания, что может быть лишь следствием определения несущественности некоторых входных (адресных) переменных. Тем не менее, существует проблема разбиения квант-вектора на составляющие части меньшей размерности, что связано с имплементацией функциональности в конструктивные компоненты LUT FPGA. В этом случае выполняется разбиение Q-вектора на два равных подвектора  $Q=(L,R)$ , которые соединяются в структурно-адресную организацию функциональности с помощью мультиплексора  $Q \parallel (\bar{a} \parallel L) \parallel (a \parallel R)$ . Если переменная мультиплексирования  $a=0$ , то функциональность Q формируется с помощью ячеек левого L-вектора, в противном случае, когда  $a=1$ , значение функции Q формируется битами правого R-вектора. Алгоритмы разбиения и имплементации сложных логических функций имеются в каждой промышленной системе синтеза, моделирования и верификации компонентов SoC.

### 5. Модель квантового процессора

Квантовый процессор может быть любой конечной размерности: вектор, матрица, куб. Для структуры, содержащей два измерения, он представлен матрицей столбцов или Q-векторов, которые формируют соответствующие им ячейки M-вектора моделирования (рис. 1, а). Вектор M совместно с X-вектором кортежей входных переменных примитивов создает структуру взаимных связей между столбцами-элементами. Адрес ячейки Q-покрытия, формирующей состояние невходного i-разряда M-вектора, определяется содержанием ячеек M-вектора, найденным по адресам, заданным i-кортежем вектора входных переменных. Каждый вектор  $Q_i$ , равно как и кортеж  $X_i$  вектора номеров входных линий, имеет адресную связь с  $M_i$ -ячейкой вектора моделирования. Квантовый процессор может входить компонентом в состав более сложной системы. Квантовая модель процессора имеет следующую структуру:

$$\begin{aligned} W &\parallel Q, M, X \parallel, \\ Q &\parallel (Q_1, Q_2, \dots, Q_i, \dots, Q_n), \\ Q_i &\parallel (Q_{i1}, Q_{i2}, \dots, Q_{ij}, \dots, Q_{ik_i}); \\ M &\parallel (M_1, M_2, \dots, M_i, \dots, M_n); \\ X &\parallel (X_1, X_2, \dots, X_i, \dots, X_n), \\ X_i &\parallel (X_{i1}, X_{i2}, \dots, X_{ij}, \dots, X_{im_i}); \\ M_i &\parallel Q_i[M(X_i)]; \quad k_i \parallel 2^{m_i}. \end{aligned}$$

В аналитической модели W представлены: 1) Упорядоченная адресно-доступная Q-совокупность квантовых примитивов, формирующих функциональность системы. 2) Вектор моделирования M, связывающий все примитивы в единую систему на основе идентификации эквипотенциальных линий, создающих формат из существенных переменных: входных, внутренних и выходных. 3) Вектор X кортежей упорядоченных номеров входных переменных для каждого квантового примитива, которые формируют адреса доступа к ячейкам Q-векторов примитивов (рис. 1, а). Вектор количества входных переменных примитива  $|X|$  формирует адресное пространство или длину каждого Q-покрытия. Его можно представить в виде таблицы кортежей входных переменных, которые формируют номера линий вектора моделирования для вычисления адресов доступа к квантовым покрытиям (рис. 1, б). Таблицу кортежей можно также представить в виде матрицы масок входов, определенных в формате вектора моделирования, для параллельного формирования адресов и одновременного считывания выходных состояний примитивов из матрицы Q-покрытий (рис. 1, в). Из структуры X-матрицы входных линий видно, что кванты, формирующие выходы: (8, 9, 10), (11, 12) и (13, 14), можно обрабатывать параллельно. 4) Характеристическое уравнение, задающее алгоритм функционирования квантового процессора на основе использования только операций транзакции (считывание-запись) между Q-векторами примитивов и вектором моделирования.

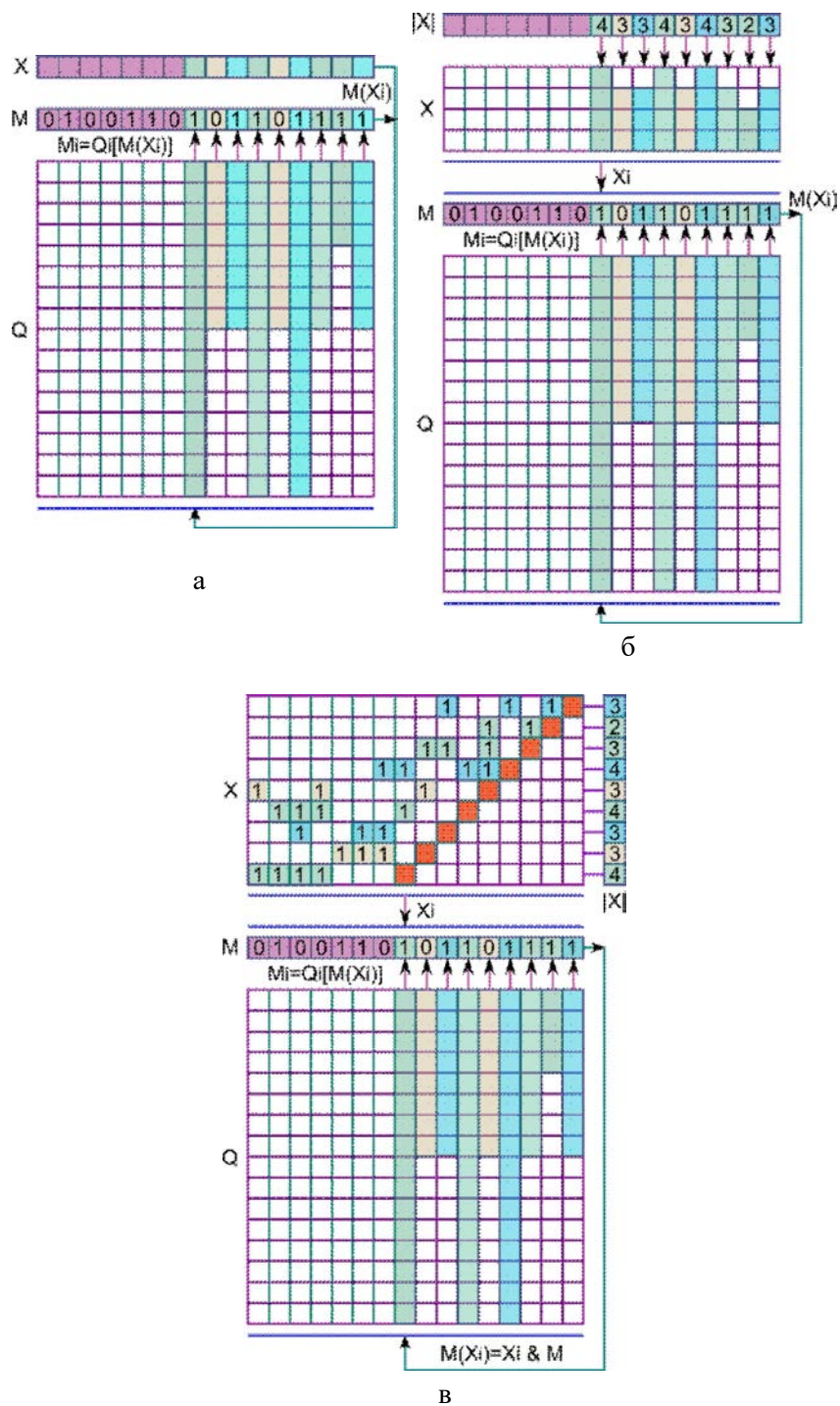


Рис. 1. Кубитные структуры данных квантового процессора

Схема цифрового устройства, соответствующая приведенному выше описанию структур данных:  $M$ -вектор моделирования,  $X$ -матрица входов,  $Q$ -матрица покрытий, представлена на рис. 2. Она содержит 9 примитивов, каждый из которых имеет  $Q$ -покрытие в форме квант-вектора, реализующего некоторую функциональность. Особенность квантовых структур данных, представляющих модель цифровой схемы, заключается в полной адресуемости всех компонентов устройства без гальванических проводных соединений.



Таблица истинности триггера представлена в форме вектора выходных состояний  $Q(S, R, Q^X) \parallel (11110001)$ , который записывается в элемент постоянной памяти, имеющий три адресных входа, сигнал синхронизации, а также обратную связь, которая соединяет выход элемента памяти с одним адресным входом. HDL-реализация в системе проектирования Active HDL 9.1 (Aldec Inc.), а также результаты верификации синтезированного SR-триггера подтверждают корректность схемотехнического решения.

Другой пример связан с синтезом на элементе постоянной памяти синхронного DV-триггера. Таблица истинности триггера трансформирована в вектор выходных состояний  $Q(D, V, Q^X) \parallel (01000111)$ , который записывается в элемент памяти, имеющий три адресных входа, сигнал синхронизации, а также обратную связь, которая соединяет выход примитива памяти с одним адресным входом. Все упомянутые компоненты, включая временные диаграммы верификации HDL-кода модели DV-триггера, представлены на рис. 4.

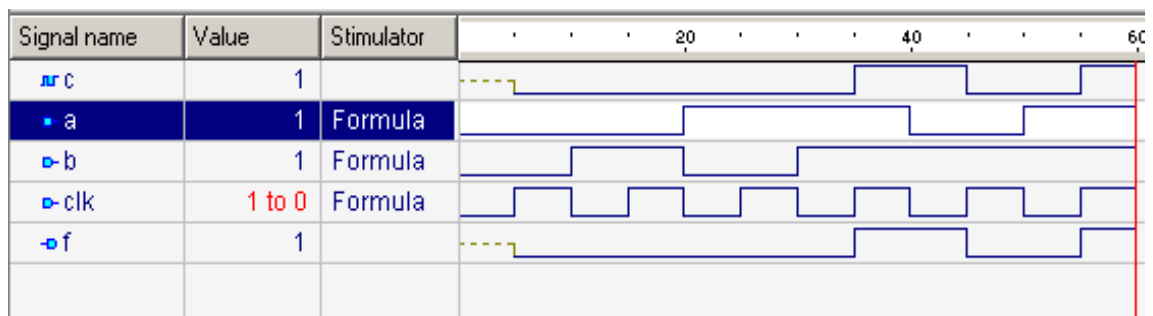
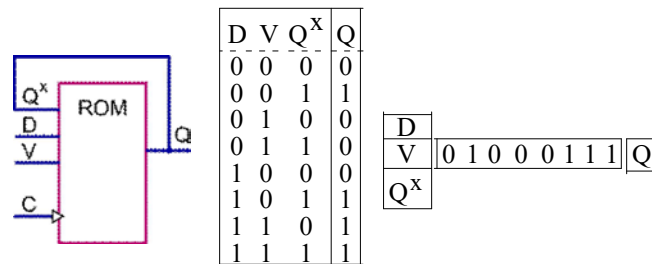


Рис. 4. Реализация DV-триггера на элементе памяти

На рис. 5 представлены модели двух последовательных примитивов: двухразрядных регистра и счетчика. Их отличие заключается в задании двух выходов, состояния которых формируются одним и тем же множеством входных переменных.

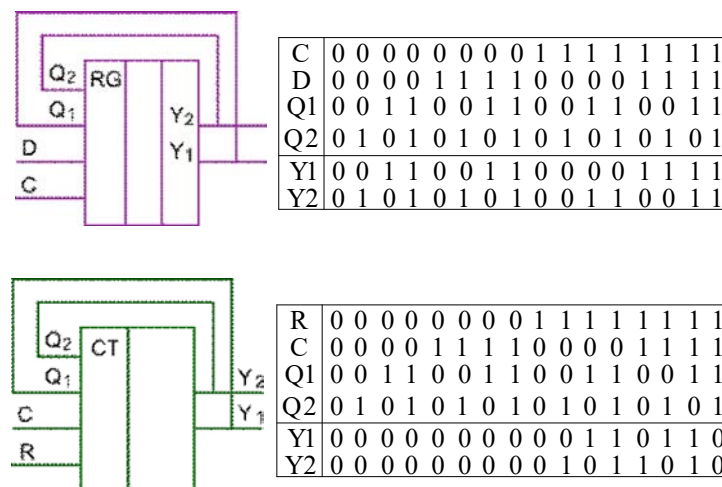


Рис. 5. Memory-based модели регистра и счетчика

Регистр на переменных (C,D,Q1,Q2,Y1,Y2) выполняет функцию сдвига вправо информации от входа D по разрядам: (D-Y1-Y2) при R=1 и сохранение данных при C=0. Счетчик, определенный на переменных (R,C,Q1,Q2,Y1,Y2), реализует функцию инкремента по разрядам (Y1,Y2) при RC = (11), а также режим хранения информации при (R or C = 0). Таким образом, для реализации двухразрядного регистра или счетчика необходимо два 16-битовых элемента памяти, работающих синхронно от одних и тех же входов:

C	
D	0 0 1 1 0 0 1 1 0 0 0 0 1 1 1 1
Q1	0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1
Q2	

R	
C	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0
Q1	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0
Q2	

Здесь каждая квантовая модель представлена двумя векторами, где каждый из них формирует функцию разряда регистра или счетчика, как состояние ячейки вектора, получаемое при формировании адреса A входными переменными:  $\{Y1, Y2\} = A(C, D, Q1, Q2)$ ,  $\{Y1, Y2\} = A(R, C, Q1, Q2)$  соответственно. Моделирование примитива сводится к тривиальной процедуре формирования адреса, по которому находится состояние выхода примитива, как содержимое ячейки квантового вектора.

### 6. Алгоритм моделирования квантовых покрытий цифровых компонентов

Использует memory-based only модели для адресного анализа цифровых систем в целях их верификации. Реализация таких структур связана с ячейками памяти (LUT (Look Up Table) FPGA), которые способны хранить информацию в виде Q-вектора, где каждый бит или разряд имеет свой адрес, отождествляемый с входным словом. Программная реализация алгоритма моделирования таких структур становится конкурентоспособной по быстродействию на рынке проектирования цифровых систем на кристаллах благодаря адресации функциональных примитивов.

Одномерный Q-вектор описания функциональности можно привязать к выходной (внутренней) линии устройства, состояние которой формируется в процессе моделирования рассматриваемого Q-покрытия. Тогда регистровая реализация комбинационного устройства может быть представлена вектором моделирования M, невходные линии которого непосредственно связаны с выходами функциональных элементов. Упорядоченные значения входных переменных задают адрес бита Q-вектора, формирующего состояние рассматриваемой невходной линии. Если функциональности описываются одновыходовыми примитивами, то каждый из них можно отождествить с номером или координатой невходной линии, на которую нагружен данный элемент. Если функциональность многовыходовая, то Q-покрытие представляется матрицей с числом строк, равным числу выходов. Эффект от такого примитива заключается в параллелизме одновременного вычисления состояний нескольких выходов за одно обращение к матрице по текущему адресу. Данное обстоятельство является существенным аргументом в пользу синтеза обобщенных кубитов для фрагментов цифрового устройства или всей схемы в целях их параллельной обработки на одном временном такте. Модель функционирования цифровой структуры упрощается до вычисления двух адресов при формировании вектора моделирования  $M_i = Q_i[M(X_i)]$  путем исключения сложного адреса выхода примитива в процессе записи состояний выходов в координаты M-вектора.

Алгоритм моделирования квантовых примитивов цифровой системы использует аналитическую структуру (k – число входных переменных i-примитива, \* – операция конкатенации битов, A – адрес бита Q-вектора):

$$M_i \square Q_i(A) \square \begin{matrix} k \\ A \square * M(X_{ij}) \\ j \square 1 \end{matrix} \square \begin{matrix} M \\ X_i \\ Q_i \end{matrix}$$

Данному аналитическому выражению можно поставить в соответствие следующие пункты алгоритма формирования двоичных состояний M-вектора моделирования цифровой схемы, изображенные на рис. 6:

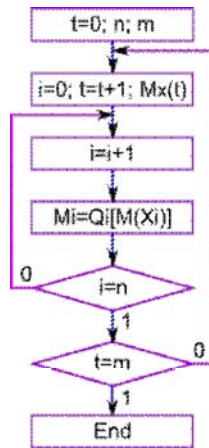


Рис. 6. Алгоритм моделирования квантовых покрытий цифровой системы

0) Инициализация начальных условий и параметров. 1) Задание очередного набора двоичных состояний на входных координатах вектора моделирования. 2) Определение  $i$ -номера очередного обрабатываемого примитива путем выполнения операции инкрементирования. 3) Выполнение процедуры конкатенации состояний битов  $M$ -вектора, соответствующих номерам вектора входных переменных  $X_i$ . Считывание соответствующего бита из функционального кубит-покрытия  $Q_i$  по двоичному вектор-адресу сконкатенированных битов  $M$ -вектора. Занесение считанного из кубита бита в вектор моделирования  $M$  по адресу  $i$ . ( $M$ -вектор может иметь координаты с символами  $X$ , что дает возможность выполнять трюичное моделирование цифровых устройств для решения задач тестирования и верификации.) 4) Если не все примитивы обработаны  $i < n$ , выполняется переход к пункту 2 алгоритма. 5) Если не все входные наборы обработаны  $t < m$ , выполняется переход к пункту 1. 6) Конец моделирования.

Исходя из характеристического уравнения квантовой модели цифровой системы можно сделать вывод, что современный <MQT> (Memory-Quant-Transaction) процессор следует представлять как адресную организацию структуры функциональных примитивов памяти без гальванических или проводных связей, на которых определены адресные транзакции данных во времени и пространстве для достижения поставленной цели.

На рис. 7 представлена схема с триггерами и комбинационной логикой, которая также описана в виде элементов памяти, куда занесены выходные состояния таблицы истинности каждого логического элемента. Структуры данных, необходимые для моделирования цифрового устройства, сведены в таблицу, где основными компонентами являются:  $M$  – вектор моделирования или состояния занумерованных линий, имеющий в данном случае 5 входных, 6 внутренних и выходных линий, состояния которых подлежат определению;  $X$  – вектор кортежей номеров входных линий примитивов, необходимых для формирования адреса в целях извлечения по нему состояния выхода элемента  $Q_i$ , функциональность которого задается  $Q$ -вектором.

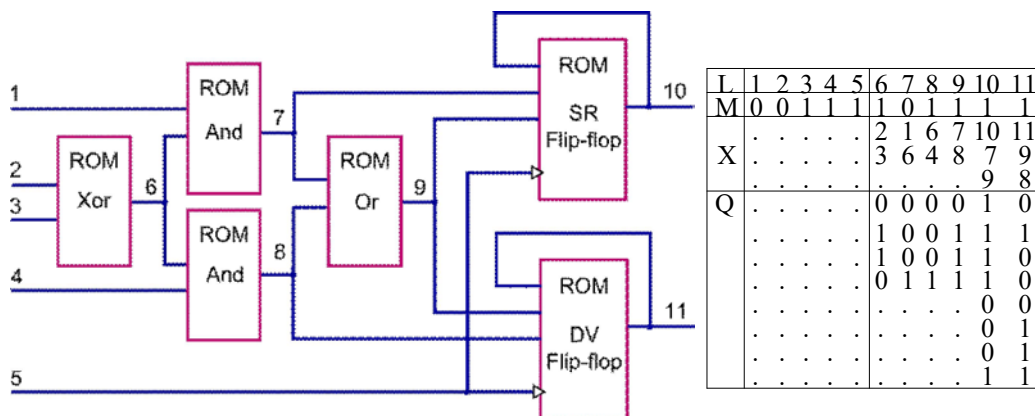


Рис. 7. Memory-based комбинационная схема с триггерами

Пример выполнения алгоритма моделирования схемной квантовой структуры. Все примитивы должны быть упорядочены по принципу: очередной элемент анализируется, если все предшественники для него были обработаны. В процессе моделирования адресно извлеченное состояние ячейки текущего Q-покрытия заносится в разряд  $M_i$  вектора моделирования. Результаты последовательной обработки всех Q-векторов схемной структуры формируют состояния линий M-вектора для приведенного выше примера ячейки (6 – 11). Первоначальные состояния неопределенностей на псевдовходах функциональных примитивов доопределяются сигналами нуля или единицы в зависимости от внутренней технологической культуры компании, предоставляющей промышленные средства моделирования и верификации. Количество входных переменных примитива  $q$  связано с длиной Q-вектора

соотношением:  $\text{card}(Q) = 2^q$ . Правильность работы алгоритма моделирования была верифицирована на тестовых и реальных схемах с привлечением средств Active HDL 9.1 (Aldec Inc.). Особенность структурно-функционального задания цифровой системы заключается в представлении всех примитивов элементами памяти, куда записываются Q-векторы выходных состояний.

Выводы: 1) Любые структурные компоненты вычислительных устройств, комбинационные и/или последовательностные, а также системы в целом можно описывать кубитными Q-векторами и реализовывать в элементах памяти FPGA, CPLD или VLSI. Это предоставляет рынку электронных технологий возможность не использовать комбинационную reusable логику при синтезе вычислительных устройств, которая доставляет разработчикам серьезные проблемы, связанные с тестированием, верификацией и ремонтом жесткой проводной реализации цифровых изделий. 2) Memory-based интерпретативное адресно-ориентированное моделирование комбинационных и последовательностных примитивов цифровых устройств становится соизмеримым по быстродействию с компилятивным анализом дискретных объектов. Кроме того, становится возможным реализовывать на программируемых логических устройствах аппаратное моделирование цифровых систем, где комбинационные и последовательностные функциональные примитивы будут представлены стандартными элементами памяти, в которые зашиваются Q-векторы.

## 7. Моделирование синхронных цифровых схем

Сигналы синхронизации доставляют определенные неудобства для описания моделей последовательностных компонентов (триггеры, регистры, счетчики) и реализации алгоритмов анализа. Это связано со схемотехническим исполнением управления по переднему или заднему фронту, которые разрешают выполнение транзакций между master-slave компонентами. Другими словами, синхронные примитивы имеют два последовательно соединенных элемента, ориентированных на низкий и высокий уровни сигналов записи данных в первую и вторую ступени соответственно. Однако для логического моделирования учет подробностей, связанных со схемотехническими решениями, может существенно замедлить время анализа цифровых схем. Поэтому здесь необходимы логически адекватные модели реальных процессов, приводящие к повышению быстродействия алгоритмов обработки компонентов. При этом накладывается ограничение, связанное только с адресным характером анализа всех компонентов схемы. Для обеспечения возможности рассмотрения синхровхода как логической переменной, формирующей адрес квантового вектора, предлагается модель разбиения последовательностного примитива на два элемента: 1) Логический квант выдачи разрешающего сигнала при формировании переднего (заднего) фронта в двух временных модельных тактах. 2) Квант реализации штатной функциональности (триггера регистра, счетчика) последовательностного компонента. Таким образом, модель синхронного D-триггера может быть описана в форме двух Q-покрытий, адресно вычисляющих состояния выходов:

$$\begin{array}{|c|} \hline \text{CLK}(t \square 1) \\ \hline \text{CLK}(t) \\ \hline \end{array} : 0 \ 1 \ 0 \ 0 \quad | \quad \text{C}(t)$$

$$\begin{array}{|c|} \hline \text{Q}(t \square 1) \\ \hline \text{D}(t) \\ \hline \text{C}(t) \\ \hline \end{array} : 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \quad | \quad \text{Q}(t)$$

С учетом изложенного выше модель цифровой схемы с двумя сигналами синхронизации, представленная на рис. 8, будет иметь структуры данных, состоящие из совокупности Q-покрытий, которые формируют текущий вектор моделирования M, но с учетом значений координат данного вектора в предыдущий момент времени M(t-1). Увеличение числа переменных за счет введения двух элементов синхронизации уменьшает совокупную размерность таблицы квантовых векторов, которая при 7 переменных будет иметь 56 координат.

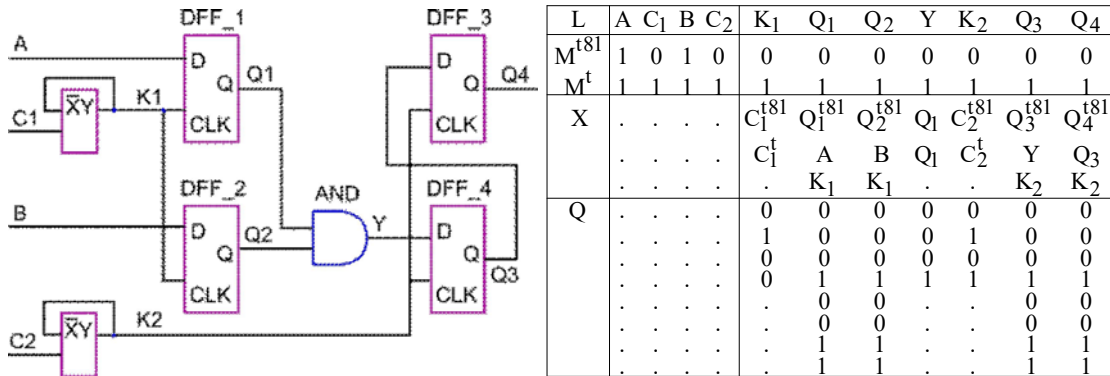


Рис. 8. Схема с D-триггерами и элементами синхронизации

Если не вводить две дополнительные переменные (элементы синхронизации), то объем памяти для Q-покрытий будет равен 80 ячейкам (рис. 9). Данная схемная реализация максимально ориентирована на структуры данных промышленных средств моделирования и верификации. Однако квантовые векторы для задания функциональностей триггеров создают необходимые условия для повышения быстродействия интерпретативного анализа, тестирования и диагностирования схемных компонентов.

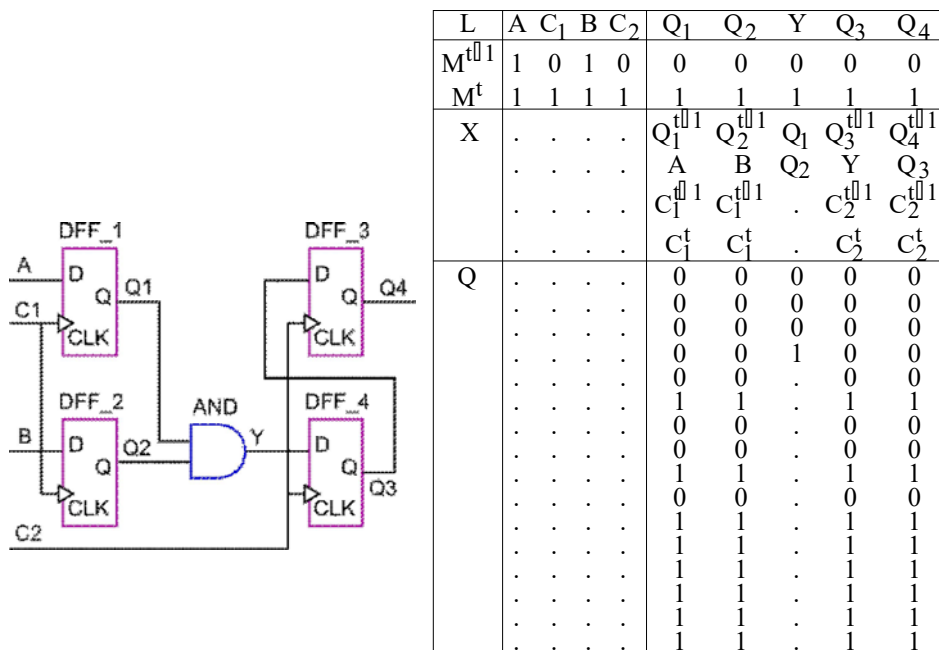


Рис. 9. Схема с D-триггерами на основе внутренней синхронизации

Проблема уменьшения совокупного объема Q-покрытий схемы связана с количеством переменных, формирующих адреса координат Q-вектора. Естественно, что любое разбиение числа переменных на два равных упорядоченных подмножества дает возможность существенно уменьшить размерность памяти для записи уже двух Q-векторов. В общем

случае функциональная зависимость уменьшения размерности исходного Q-вектора, определенного на n-переменных, при делении на 2 подсхемы с равным числом результирующих переменных (n/2) имеет следующий вид:

$$Q \approx \frac{2^n}{2^{n/2} \cdot 8 \cdot 2^{n/2}} \cdot 8 \cdot \frac{2^n}{28 \cdot 2^{n/2}} \cdot 8 \cdot \frac{2^n}{2^{n/281}} \cdot 8 \cdot 2^{n8(n/281)} \cdot 8 \cdot 2^{n/281}.$$

Например, разбиение вектора из восьми переменных на два Q-покрытия из 4 переменных уменьшает объем памяти в 8 раз. Однако следует иметь в виду, что каждое разбиение функционального модуля на k подсхем относительно внешних входов потребует k дополнительных d-циклов для вычисления состояния выхода всей схемы  $\forall d(k \forall 1)$ . Снижение быстродействия разбиенной функциональности является платой за существенное уменьшение объема памяти для хранения структур данных цифровой системы. В общем случае разбиение функциональности на k одинаковых частей приводит к получению следующей зависимости выигрыша объема памяти от числа разбиений на подмножества вектора входных переменных:

$$Q \approx \frac{2^n}{k \cdot 8 \cdot 2^{n/k}} \cdot 8 \cdot \frac{1}{k} \cdot 8 \cdot 2^{n8n/k}.$$

Здесь параметр разбиения k принимает значения, кратные степени двойки: 2, 4, 8, 16. Однако значение k не должно быть более, чем n/2. Следует заметить, что необязательно количество разбиений должно принимать значения, кратные степени двойки. В общем случае, на векторе входных переменных может существовать m разбиений, каждое из которых имеет более одной переменной. При этом выполняется условие разбиения, что сумма всех переменных, участвующих в разбиениях, не может быть больше n:

$$Q \approx \frac{2^n}{2^{k_1} \cdot 2^{k_2} \cdot \dots \cdot 2^{k_i} \cdot \dots \cdot 2^{k_m}}, \quad k_1 + k_2 + \dots + k_i + \dots + k_m = n.$$

Формула показывает выигрыш от разбиения функциональности в виде отношения размерности исходного квант-вектора к совокупному объему Q-векторов, полученных после разбиения. Чтобы оценить эффективность разбиения функциональности на схемные фрагменты, необходимо учитывать не только уменьшение объема памяти для хранения структур данных, но и негативные последствия, связанные со стоимостью анализа увеличенного количества схемных компонентов, которая регламентируется в каждом конкретном случае коэффициентом d:

$$Q \approx \frac{2^n}{d \cdot 8 \cdot m \cdot 8 \cdot (2^{k_1} \cdot 8 \cdot 2^{k_2} \cdot 8 \dots \cdot 8 \cdot 2^{k_i} \cdot 8 \dots \cdot 8 \cdot 2^{k_m})}.$$

Подводя итог в части модификации теории исправного моделирования (fault-free) цифровых систем, можно отметить следующие факты. Автомат моделирования синхронных цифровых устройств, как правило, представлен моделью Мура:

$$\begin{aligned} S(t) &\forall f[X(t), S(t \forall 1)]; \\ Y(t) &\forall f[X(t), S(t)]. \end{aligned}$$

Здесь фигурируют входные (X) и внутренние состояния автомата в двух соседних временных фреймах S(t), S(t-1), а также правила определения выходных значений Y(t) для инициирования вычислительных процедур. Предлагается модификация упомянутой модели автомата Мура для анализа цифровых систем, суть которой заключается в замене функциональных отношений адресными (A) операциями:

$$\begin{aligned} S(t) &( A[X(t), S(t-1)]); \\ Y(t) &( A[X(t), S(t)]. \end{aligned}$$

Адресный или квантовый автомат, определенный выше, позволяет: 1) Избежать жестких гальванических межсоединений между элементами комбинационных и последовательностных схем при их аппаратной имплементации только в элементы памяти. 2) Получить свойство гибкой заменяемости компонентов цифровой системы в режиме online, благодаря их адресуемости. 3) Существенно упростить все процессы моделирования, верификации и тестирования путем использования только процедур вычисления адреса компонента схемы

или ячейки его памяти. 4) Унифицировать процессы проектирования цифровых изделий путем их сведения к формированию функциональностей на основе вычисления адресов или к транзакциям на элементах памяти. 5) Повысить эффективность процедур моделирования цифровых схем путем уменьшения объема интерпретативных моделей и упрощения способа их обработки, когда вместо исчерпывающего анализа таблиц предлагается вычисление адреса ячейки квантового вектора. 6) Выполнять все вычислительные процедуры на основе использования квантовых покрытий и вектора моделирования, заданного в двух соседних автоматных тактах, согласно определению квантового автомата. 7) Технологически проще становится использовать инфраструктуру [6-7] стандартов тестопригодного проектирования (IEEE 1500, 1149) для покомпонентного тестирования, диагностирования и восстановления работоспособности адресно доступных функциональных блоков в режиме online.

## 8. Заключение

Предложен новый подход к проектированию цифровых устройств, который характеризуется: 1) Использованием элементов памяти для реализации транзакционного взаимодействия всех компонентов операционного и управляющего автоматов. 2) Методами синтеза и анализа, основанными на суперпозиции кубит-векторных примитивов задания всех типов функциональностей, имплементируемых в элементы памяти, что дает возможность существенно повысить быстродействие средств моделирования, тестирования и верификации, а также значительно упростить процедуры создания реальных и виртуальных компьютерных систем. 3) Оригинальными структурами данных для моделирования и верификации цифровых систем, которые дают возможность существенно упростить алгоритмы реализации и повысить их быстродействие за счет адресуемости функциональных квантов и параллельности обработки компонентов. 4) Реализацией всех вычислительных структур и процессов на основе использования введенного квантового адресного автомата, который дает возможность непосредственно использовать инфраструктуру стандартов тестопригодного проектирования для повышения выхода годной продукции благодаря online ремонту функциональных примитивов.

Практическая значимость нового подхода синтеза и анализа цифровых систем заключается в следующем: 1) Реализация процессора только на основе применения элементов памяти делает его однородным по структуре и типам функциональных примитивов, что доставляет очевидные технологические удобства процессам проектирования, производства и эксплуатации, включая верификацию, встроенные тестирование и диагностирование, а главное – ремонт в режиме online в результате использования на кристалле универсальных адресуемых spare-компонентов памяти. 2) Моделирование в процессе верификации проектируемых вычислителей на основе адресных моделей компонентов делает данную процедуру технологически простой из-за регулярных структур данных и применения единственной операции транзакции на элементах памяти, а также более быстродействующей за счет возможности параллельной квантоподобной обработки больших массивов однотипной памяти. 3) Имплементация квантовых only memory-based моделей описания цифровых компонентов и систем непосредственно связана с увеличением выхода годной продукции, повышением надежности вычислительных изделий, снижением стоимости проектирования и изготовления, а также автономным восстановлением работоспособности в режиме remote & online, без участия человека.

**Список литературы:** 1. *Metodi T., Chong F.* Quantum Computing for Computer Architects. Synthesis Lectures on Computer Architecture. Morgan & Claypool. 2006. 154 p. 2. *Stenholm Stig, Kalle-Antti Suominen.* Quantum approach to informatics. John Wiley & Sons, Inc. 2005. 249 p. 3. *Hahanov V.I., Wajeb Gharibi, Litvinova E.I., Shkil A.S.* Qubit data structure of computing devices // Electronic modeling, № 1. 2015. P. 76-99. 4. *Vladimir Hahanov, Tamer Bani Amer, Ivan Hahanov.* MQT-model for Virtual Computer Design // Proc. of Microtechnology and Thermal Problems in Electronics (Microtherm) 23-25 June 2015. P. 182-185. 5. *Hahanov V.I., Litvinova E.I., Chumachenko S.V.* et al. Qubit Model for solving the coverage problem // Proc. of IEEE East-West Design and Test Symposium. Kharkov. 14-17 September 2012. P. 142 – 144. 6. *Zorian Y. Shoukourian S.* Test solutions for nanoscale Systems-on-Chip: Algorithms, methods and test infrastructure. Computer Science and Information Technologies (CSIT), 2013. P. 1 – 3. 7. *Zorian Y., Shoukourian S.* Embedded-memory test and repair: infrastructure IP for SoC yield. Design & Test of Computers, IEEE

(Volume: 20, Issue: 3). P. 58 – 66. **8.** *Dugganapally I.P., Watkins S.E., Cooper B.* Multi-level, Memory-Based Logic Using CMOS Technology. VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on. Tampa, FL. P. 583-588. **9.** *Yueh W., Chatterjee S., Zia M., Bhunia S., Mukhopadhyay S.* A Memory-Based Logic Block With Optimized-for-Read SRAM for Energy-Efficient Reconfigurable Computing Fabric. Circuits and Systems II: Express Briefs, IEEE Transactions on. Volume: 62 Issue: 6. P. 593-597. **10.** *Matsunaga S., Hayakawa J., Ikeda S., Miura K., Endoh T., Ohno H., Hanyu T.* MTJ-based nonvolatile logic-in-memory circuit, future prospects and issues. Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09. P. 433 – 435. **11.** *Harada S., Xu Bai, Kameyama M., Fujioka Y.* Design of a Logic-in-Memory Multiple-Valued Reconfigurable VLSI Based on a Bit-Serial Packet Data Transfer Scheme. Multiple-Valued Logic (ISMVL), 2014 IEEE 44th International Symposium on. P. 214 – 219. **12.** *Hahanov V.I., Tamer Bani Amer, Chumachenko S.V., Litvinova E.I.* Qubit technology analysis and diagnosis of digital devices // Electronic modeling. Vol. 37, № 3. 2015. P. 17-40. **13.** *Melikyan V.Sh.* A method of eliminating false paths during statistical static analysis of timing delays of digital circuits // Elektronika i svyaz, Vol. 2-3, No. 1, 2009. P. 93-96. **14.** *Melikyan V.Sh., Vatyay A.O.* Interconnections model delays for the logic analysis of ECL circuits // SUAB, Vol. 2, Computer Engineering, Moscow, 1997. P. 187-194.

*Поступила в редколлегию 15.09.2015*

**Хаханов Иван Владимирович**, студент факультета компьютерной инженерии и управления ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, программирование. Увлечения: горные лыжи, английский язык. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326.

**Литвинова Евгения Ивановна**, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: [kiu@kture.kharkov.ua](mailto:kiu@kture.kharkov.ua).

## МЕТОД КОСВЕННОГО СТЕГАНОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ФУНКЦИОНАЛА ДЛЯ АДАПТИВНОГО ПОЗИЦИОННОГО КОДИРОВАНИЯ

Разрабатывается метод косвенного стеганографического встраивания на основе использования функционала для адаптивного позиционного кодирования. Проектируется система косвенного стеганографического встраивания, которая позволяет встраивать информацию путем модификации оснований элементов изображения-контейнера. Проводится анализ изображений, полученных в результате обратного стеганографического преобразования.

### 1. Введение

Современное развитие инфокоммуникационных сетей диктует новые требования к безопасности информационных ресурсов. Актуальным направлением для решения проблемы повышения безопасности является применение методов стеганографического встраивания.

Существующие методы стеганографии представлены алгоритмами непосредственного и косвенного встраивания. Такие методы позволяют скрытно передавать информацию с помощью психовизуальной избыточности изображения-контейнера. Недостатком указанного подхода является низкая устойчивость встроенных данных к активным воздействиям злоумышленника, направленным на разрушение встроенного сообщения. Также для таких методов характерно ухудшение визуальной устойчивости стеганограммы при попытке увеличения объема встраиваемой информации.

Возможным решением проблемы улучшения показателей визуальной устойчивости стеганограммы, а также стойкости к трансформации и атакам является использование для косвенного встраивания неравновесного позиционного кодирования. Отсюда, цель исследования состоит в разработке метода косвенного стеганографического встраивания и построении структурной модели стеганографической системы.

### 2. Разработка метода косвенного стеганографического встраивания на основе функционального преобразования для неравновесного позиционного кодирования

Функциональное преобразование для адаптивного позиционного кодирования позволяет выявить структурные закономерности в изображении. Такие закономерности обусловлены ограничением на динамический диапазон. Величина  $\psi$  динамического диапазона представления фрагмента  $F$  изображения-контейнера определяется на основе следующего выражения:

$$\psi = \max_{1 \leq i \leq m} \{a_{i,j}\} + 1 \quad j = \overline{1, n}, \quad (1)$$

здесь  $a_{i,j}$  –  $j$ -й элемент в  $i$ -й строке массива  $F$ .

В процессе реализации функционального преобразования на основе адаптивного позиционного кодирования фрагмент  $F$  исходного изображения рассматривается как множество адаптивных позиционных чисел  $\{A(j)\} : A(j) = \{a_{1,j}; \dots; a_{i,j}; \dots; a_{m,j}\}$ .

Значение кода  $N(j)$  будет определяться как сумма произведений элементов позиционного числа  $A(j)$  на их весовые коэффициенты  $V_{i,j}$  по формуле:

$$N(j) = \sum_{i=1}^m a_{i,j} V_{i,j}. \quad (2)$$

Здесь  $a_{i,j}$  –  $(i; j)$ -й элемент адаптивного позиционного числа  $A(j)$ ;  $V_i$  – весовой коэффициент элемента  $a_{i,j}$  адаптивного позиционного числа  $A(j)$  фрагмента  $F$ .

Весовой коэффициент  $V_{i,j}$  элемента  $a_{i,j}$  зависит от его позиции в числе  $A(j)$  и вычисляется как произведение оснований всех младших элементов. Но учитывая, что основание  $\psi$  для всех элементов  $a_{i,j}$  позиционного числа  $A(j)$  принимает одинаковое значение, весовой коэффициент будет  $V_{i,j}$  вычисляться по следующей формуле:

$$V_i = \psi^{m-i}. \quad (3)$$

Второй этап предусматривает формирование кодограммы  $C(F)$ , которая включает служебную составляющую  $C(\Psi)$  и информационную составляющие  $C(j)$ . Данный этап реализуется при помощи оператора выделения разрядов  $\varphi_c(\bullet)$  по формуле:

$$C(F) = \varphi_c(C(j), \Psi),$$

где  $\Psi$  – базис, содержащий информацию об основаниях для адаптивных позиционных чисел фрагмента  $F$ ;  $C(j)$  – кодограмма кодового представления адаптивного позиционного числа  $A(j)$ .

Кодограмма  $C(j)$  имеет следующий вид:  $C(j) = \{c_1, \dots, c_\xi, \dots, c_{q(C(j))}\}$ , где  $q(C(j))$  – длина двоичной кодограммы  $C(j)$ ;  $c_\xi$  –  $\xi$ -й двоичный разряд кодограммы  $C(j)$ .

Процесс реконструкции элемента  $a_{i,j}$  для адаптивного позиционного числа  $A(j)$  на основе кода  $N(j)$  выполняется по формуле

$$a'_{i,j} = \left[ \sum_{i=1}^m a_{i,j} V_i / V_i \right] - \left[ \sum_{i=1}^m a_{i,j} V_i / (\psi V_i) \right] \psi.$$

Такое преобразование осуществляется без внесения искажений.

В случае адаптивного позиционного кодирования значение реконструированного элемента  $a_{i,j}$  числа  $A(j)$  фрагмента  $F$  не меняется при кодировании и декодировании с различными основаниями  $\psi$  и  $\psi'$ , т.е.

$$a'_{i,j} = a_{i,j} = \left[ N(j) / V_i \right] - \left[ (N(j) / (\psi V_i)) \right] \psi = \left[ N'(j) / V'_i \right] - \left[ (N'(j) / (\psi' V'_i)) \right] \psi' = a''_{i,j}, \quad (4)$$

где  $a'_{i,j}$  – элемент числа  $A(j)$ , реконструированный на основе системы оснований  $\psi$ ;  $a''_{i,j}$  – элемент числа  $A(j)$ , реконструированный на основе системы оснований  $\psi'$ ;  $N(j)$  – кодовое представление числа  $A(j)$ , сформированное в базисе оснований  $\psi$ ;  $N'(j)$  – кодовое представление числа  $A(j)$ , сформированное в базисе оснований  $\psi'$ ;  $\psi'$  – значение модифицированного основания элемента  $a'_{i,j}$ .

Докажем данное свойство адаптивного позиционного представления при условии, что  $\psi' \geq \psi$ . Распишем левую часть выражения (4) с учетом соотношения для кода  $N(j)$ :

$$N(j) = \sum_{i=1}^m a_{i,j} V_i.$$

Тогда получим:

$$a'_{i,j} = \left[ \sum_{i=1}^m a_{i,j} V_i / V_i \right] - \left[ \sum_{i=1}^m a_{i,j} V_i / (\psi V_i) \right] \cdot \psi. \quad (5)$$

Рассмотрим первое слагаемое выражения (5):

$$\left[ \frac{\sum_{i=1}^m a_{i,j} V_i}{V_i} \right] = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V_\xi + \sum_{\xi=i+1}^m a_{\xi,j} V_\xi}{V_i} \right].$$

Перепишем данное выражение с учетом следующего соотношения:

$$V_i > \sum_{\xi=i+1}^m a_{\xi,j} V_\xi. \quad (6)$$

В этом случае получим:

$$\left[ \frac{\sum_{i=1}^m a_{i,j} V_i}{V_i} \right] = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V_\xi}{V_i} \right]. \quad (7)$$

Теперь преобразуем полученное выражение с учетом формулы для весового коэффициента  $V_i$ :

$$V_i = \psi^{m-i}. \quad (8)$$

Тогда (7) примет следующий вид:

$$\left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V_\xi}{V_i} \right] = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} \cdot \psi^{m-\xi}}{\psi^{m-i}} \right] = \sum_{\xi=1}^i a_{\xi,j} \cdot \psi^{i-\xi}.$$

Теперь рассмотрим второе слагаемое выражения (5):  $\left[ \frac{\sum_{i=1}^m a_{i,j} V'_i}{\psi' V'_i} \right] \cdot \psi'$ .

Перепишем его с учетом следующего неравенства:

$$\psi V_{i,j} > \sum_{\xi=i}^m a_{i,j} V_i. \quad (9)$$

И тогда получим:

$$\left[ \frac{\sum_{i=1}^m a_{i,j} V_i}{\psi V_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V_\xi + \sum_{\xi=i}^m a_{\xi,j} V_\xi}{\psi V_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V_\xi}{\psi V_i} \right]. \quad (10)$$

Преобразуем полученное выражение с учетом формулы (8):

$$\left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V_\xi}{\psi V_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi^{m-\xi}}{\psi \psi_i^{m-i}} \right] = \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi^{i-\xi-1}.$$

Перепишем (5) с учетом выполненных преобразований. В этом случае выражение (5) примет следующий вид:

$$\begin{aligned} a'_{i,j} &= \sum_{\xi=1}^i a_{\xi,j} \cdot \psi^{i-\xi} - \left[ \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi^{i-\xi-1} \right] \cdot \psi = \\ &= \sum_{\xi=1}^i a_{\xi,j} \cdot \psi^{i-\xi} - \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi^{i-\xi} = a_{i,j}. \end{aligned} \quad (11)$$

Теперь перепишем правую часть выражения (4) с учетом формулы для кода  $N'(j)$ , сформированного с учетом модифицированного основания  $\psi'$ :  $N'(j) = \sum_{i=1}^m a_{i,j} V'_i$ .

Тогда получим:

$$a''_{i,j} = \left[ \sum_{i=1}^m a_{i,j} V'_i \right] / V'_i - \left[ \sum_{i=1}^m a_{i,j} V'_i \right] / (\psi' V'_i) \cdot \psi'. \quad (12)$$

Рассмотрим первое слагаемое выражения (12):

$$\left[ \sum_{i=1}^m a_{i,j} V'_i \right] / V'_i = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V'_\xi + \sum_{\xi=i+1}^m a_{\xi,j} V'_\xi}{V'_i} \right].$$

Перепишем данное выражение с учетом соотношения (6). В этом случае получим:

$$\left[ \sum_{i=1}^m a_{i,j} V'_i \right] / V'_i = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V'_\xi}{V'_i} \right]. \quad (13)$$

Теперь преобразуем полученное выражение с учетом формулы (8) для весового коэффициента  $V'_i$ :

$$\left[ \frac{\sum_{\xi=1}^i a_{\xi,j} V'_\xi}{V'_i} \right] = \left[ \frac{\sum_{\xi=1}^i a_{\xi,j} \cdot \psi'^{m-\xi}}{\psi'^{m-i}} \right] = \sum_{\xi=1}^i a_{\xi,j} \cdot \psi'^{i-\xi}.$$

Далее рассмотрим второе слагаемое выражения (12):  $\left[ \frac{\sum_{i=1}^m a_{i,j} V'_i}{\psi' V'_i} \right] \cdot \psi'$

Перепишем его с учетом неравенства (9). Тогда получим:

$$\left[ \frac{\sum_{i=1}^m a_{i,j} V'_i}{\psi' V'_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V'_\xi + \sum_{\xi=i}^m a_{\xi,j} V'_\xi}{\psi' V'_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V'_\xi}{\psi' V'_i} \right]. \quad (14)$$

Преобразуем полученное выражение (14) с учетом формулы (8):

$$\left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} V'_\xi}{\psi' V'_i} \right] = \left[ \frac{\sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi'^{m-\xi}}{\psi' \cdot \psi'^{m-i}} \right] = \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi'^{i-\xi-1}.$$

Перепишем выражение (12) с учетом выполненных преобразований. В этом случае оно примет следующий вид:

$$a''_{i,j} = \sum_{\xi=1}^i a_{\xi,j} \cdot \psi'^{i-\xi} - \left[ \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi'^{i-\xi-1} \right] \cdot \psi' = \sum_{\xi=1}^i a_{\xi,j} \cdot \psi'^{i-\xi} - \sum_{\xi=1}^{i-1} a_{\xi,j} \cdot \psi'^{i-\xi} = a_{i,j}.$$

Теперь перепишем выражение (4) с учетом преобразованной правой и левой части:

$$a'_{i,j} = a''_{i,j} = a_{i,j}.$$

Отсюда можно заключить, что свойство (4) доказано.

Предлагается использовать данное свойство для проектирования системы косвенного стеганографического встраивания. Косвенное встраивание элемента  $b_\xi$  скрываемого сообщения  $V = \{b_1; \dots; b_\xi; \dots; b_v\}$  предлагается проводить в блок изображения-контейнера путем модификации основания  $\Psi_i$  базиса  $\Psi$  на основе следующего правила:

$$\psi' = \psi + k, \text{ где } k = b_\xi.$$

Здесь  $\psi'$  – основание, модифицированное в результате косвенного стеганографического встраивания;  $k$  – коэффициент модификации.

На следующем этапе вычисляется значение кода  $N'(j)$  для числа  $A(j)$  с учетом модифицированного основания  $\Psi'$ :  $N'(j) = \sum_{i=1}^m a_{i,j} \Psi'$ .

Третий этап предусматривает формирование кодограммы  $C'(F)$ , которая включает служебную  $C(\Psi')$  и информационную составляющие  $C'(j)$ :

$$C'(F) = \varphi_c(C'(j), \Psi'),$$

где  $\varphi_c$  – оператор выделения разрядов.

Косвенное встраивание бита  $b_\xi \in [0; 1]$  будет выполняться по формуле:

$$\Psi' = \begin{cases} \max_{1 \leq j \leq n} \{a_{i,j}\} + 1, & b_\xi \rightarrow 0; \\ \max_{1 \leq j \leq n} \{a_{i,j}\} + 1 + 1, & b_\xi \rightarrow 1. \end{cases} \quad (15)$$

Перепишем выражение (15) с учетом формулы (1). Тогда получим:

$$\Psi' = \begin{cases} \Psi, & b_\xi \rightarrow 0; \\ \Psi + 1, & b_\xi \rightarrow 1. \end{cases} \quad (16)$$

Или

$$\Psi' = \Psi + b_\xi. \quad (17)$$

Таким образом, предложенный подход позволяет осуществить косвенное стеганографическое встраивание сообщения  $V = \{b_1; \dots; b_\xi; \dots; b_v\}$ ,  $b_\xi \in [0; 1]$ ,  $\xi = \overline{1, v}$  в блоки исходного изображения-контейнера.

### 3. Выводы

Разработан подход для улучшения характеристик косвенного стеганографического встраивания, включающий в себя прямое и обратное стеганографическое преобразование при помощи кодообразующего функционала для адаптивного позиционного кодирования. Косвенное встраивание предлагается проводить путем модификации оснований для блоков изображения-контейнера.

Разработана схема системы косвенного стеганографического встраивания. Система включает в себя следующие базовые составляющие:

- косвенное стеганографическое встраивание;
- косвенное стеганографическое изъятие при наличии ключевой информации.

**Список литературы:** 1. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М.: Солон-Пресс, 2002. 272 с. 2. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. К.: МК-Пресс, 2006. 288с. 3. Тарасов Д.О., Мельник А.С., Голобородько М.М. Класифікація та аналіз безкоштовних програмних засобів стеганографії // Інформаційні системи та мережі. Вісник НУ “Львівська політехніка”. 2010. № 673. С. 365-374.

*Поступила в редколлегию 13.09.2015*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры автоматизированных систем управления, Харьковский университет Воздушных Сил им. И. Кожедуба. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

**Юдин Александр Константинович**, д-р техн. наук, профессор, директор института компьютерных информационных технологий Национального авиационного университета. Адрес: Украина, 01000, Киев, пр.Космонавта Комарова, 1.

**Фролов Олег Владимирович**, соискатель Национального авиационного университета. Адрес: Украина, 01000, Киев, пр.Космонавта Комарова, 1.

## МОДЕЛЬ ОЦЕНКИ ИНФОРМАТИВНОСТИ СЛОТА Р-КАДРОВ НА ОСНОВЕ ВЫЯВЛЕНИЯ СТРУКТУРНО-ГРАДИЕНТНЫХ МЕЖТРАНСФОРМАНТНЫХ ОГРАНИЧЕНИЙ

Разрабатывается модель оценки битовой интенсивности слотов Р-кадров на основе эффективного синтаксического представления информативных дифференциально-описанных спектрограмм с учетом обобщенной характеристики градиентного пространства локально-структурных (градиентных) ограничений. Показывается, что такой тип структурных закономерностей обусловлен целым рядом причин, а именно: наличие коррелированности между видеокадрами; незначительные изменения освещенности объектов видеосцен; зависимости между компонентами ДКП соседних трансформант; предварительное снижение динамического диапазона элементов спектрограмм в результате их дифференциального описания. Обосновываются потенциальные характеристики создаваемого подхода относительно дополнительного снижения битовой интенсивности динамического информационного ресурса.

### 1. Введение

**Постановка проблемы и анализ литературы.** Снижение безопасности информации и качества предоставления видеoinформационных услуг с использованием инфокоммуникаций в значительной степени вызвано превышением битовой интенсивности видеопотока относительно пропускной способности сети [1]. Такое противоречие особенно обостряется в условиях селективной обработки видеопотока на основе закрытия базового кадра.

Использование технологий обработки видеопотока обеспечивает уменьшение его битовой интенсивности. Здесь применяются такие технологии как MPEG [2 – 4]. В то же время такое синтаксическое представление потока кадров связано с внесением потерь целостности информации и дополнительными временными задержками на обработку. Интегрирование в информационные системы базовых технологий обработки динамических видеоресурсов (ДВИР) неоднозначно влияет на эффективность их функционирования. Особенно это проявляется в случае необходимости беспроводного доступа к ДВИР, источник формирования которых находится на значительных расстояниях. Значит, актуальным требованием, определяющим *научно-прикладную задачу*, является создание систем эффективного синтаксического представления динамических видеoinформационных ресурсов в режимах сохранения необходимого уровня безопасности информации и сокращения сложности реализации кодеков. Для решения такой задачи необходимо построение технологий обработки ДВИР с использованием методов, устраняющих межкадровую избыточность с учетом форм представления информации на внутрикадровом уровне [5]. Для MPEG-технологий обработка потока кадров проводится по группам с применением процесса формирования Р-кадров, т.е. образования слотов Р-кадров [4]. Поэтому *цель исследования* заключается в создании математической модели для оценки информативности слота Р-кадров на основе выявления межкадровых закономерностей.

### 2. Обоснование подхода для эффективного синтаксического представления слота Р-кадров

Для повышения эффективности формирования и обработки кадров Р-типа *предлагается* предварительно осуществлять трансформирование видеокadra из пространственно-временного в пространственно-спектральное описание. После получения трансформант двумерного преобразования проводится построение дифференциального представления на основе формирования величин разностей между соответствующими компонентами по позициям в кадрах.

В результате этого группа  $Y_T$  трансформированных кадров заменяется на совокупность, состоящую из одного базового трансформированного кадра  $Y(1)$  и последовательности  $E_{T-1}$  дифференциально представленных трансформированных кадров  $E(\tau)$ ,  $\tau = \overline{2, T}$ , т.е.  $Y_T \rightarrow \{Y(1); E_{T-1}\}$ . В данной формуле составляющая  $E_{T-1}$  представляет собой последовательность длиной  $(T-1)$  кадр и записывается как  $E_{T-1} = \{E(2), \dots, E(T)\}$ . Здесь каждый дифференциально описанный трансформированный (ДОТ) кадр  $E(\tau)$  образуется по совокупности двумерных массивов  $E(\tau)_{\chi, \gamma}$  размером  $w \times w$  элементов. При этом срез по  $\tau$  составляющей в последовательности ДОТ кадров образует слот ДОТ кадров, состоящий из последовательности двумерных массивов  $E(\tau)_{\chi, \gamma}$  для  $\tau = \overline{2, T}$ . Слот  $E(T-1)_{\chi, \gamma}^{(2)}$  ДОТ кадров представляется следующим выражением:

$$E(T-1)_{\chi, \gamma}^{(2)} = \{E(2)_{\chi, \gamma}, \dots, E(\tau)_{\chi, \gamma}, \dots, E(T)_{\chi, \gamma}\},$$

где  $E(T-1)_{\chi, \gamma}^{(2)}$  – последовательность (слот) двумерных массивов с координатами  $(\chi; \gamma)$  в ДОТ кадрах длиной  $(T-1)$ .

Слот  $E(T-1)_{\chi, \gamma}^{(2)}$  фактически представляет собой сквозную квадратную трубу, вырезанную вдоль последовательности ДОТ кадров на позиции  $(\chi; \gamma)$ , в каждом сечении которого находится двумерный массив  $E(\tau)_{\chi, \gamma}$  дифференциального описания трансформированного кадра.

Далее *предлагается* обработку Р-кадров проводить в межтрансформантном направлении по временной оси по слотам. Для этого нужно выявить структурные свойства слотов Р-кадров с учетом структурных особенностей дифференциально-описанных трансформант. При этом для повышения чувствительности выявления закономерностей относительно динамического изменения содержания слотов предлагается ввести такую структурную единицу как дифференциально описанная спектрограмма  $E^{(k, \ell)}$ .

Дифференциально описанная спектрограмма (ДОС)  $E^{(k, \ell)}$  представляет собой срез вдоль слота по координате  $(k; \ell)$  в каждой трансформанте, т.е.

$$E^{(k, \ell)} = \{e(2)^{(k, \ell)}, \dots, e(\tau)^{(k, \ell)}, \dots, e(T)^{(k, \ell)}\}.$$

Структурной характеристикой ДОС является динамический диапазон  $d_e^{(k, \ell)}$  ее элементов. Поскольку в общем случае межтрансформантные разности  $e(\tau)^{(k, \ell)}$  принимают как положительные, так и отрицательные значения, величина  $d_e^{(k, \ell)}$  без учета знака находится по формуле  $d_e^{(k, \ell)} = \max_{2 \leq \tau \leq T} \{|e(\tau)^{(k, \ell)}|\}$ , где  $d_e^{(k, \ell)}$  – динамический диапазон элементов  $(k; \ell)$ -й дифференциально описанной спектрограммы без учета знака величин  $e(\tau)^{(k, \ell)}$ ;  $|e(\tau)^{(k, \ell)}|$  – абсолютное значение межтрансформантной разности между  $(\tau-1)$ -й и  $\tau$ -й трансформантами.

Для ДОС характерны следующие закономерности (свойства). *Первое свойство.* Динамический диапазон для элементов ДОС в низкочастотной области трансформант больше, чем динамический диапазон для высокочастотной области. *Второе свойство* состоит в том, что значения динамических диапазонов элементов ДОС внутри слота Р-кадров распределены неравномерно. *Третье свойство.* Количество ДОС с высоким динамическим диапазоном меньше, чем количество ДОС с меньшим динамическим диапазоном. *Четвертое свойство.* Для кадров с наличием высокой когерентности динамические диапазоны элементов ДОС в области низкочастотных компонент также имеют ограниченные значения.

### 3. Разработка модели оценки информативности слота Р-кадров на основе выявления межкадровых закономерностей

Для выполнения требований относительно повышения безопасности динамических ВИР на основе повышения эффективности синтаксического представления слотов Р-кадров с учетом структурных и статистических свойств ДОС *предлагается* выбирать в качестве базового подхода – структурно-комбинаторный принцип описания.

Данный подход базируется на синтезе функционала  $F(E^{(k,\ell)})_{\text{apг}}$ , задающего синтаксическое представление (аппроксимацию) дифференциально описанной спектрограммы  $E^{(k,\ell)}$ . При построении функционала  $F(E^{(k,\ell)})_{\text{apг}}$  необходимо учитывать, что элементы  $e(\tau)^{(k,\ell)}$  ДОС принимают как положительные, так и отрицательные значения, т.е.

$$e(\tau)^{(k,\ell)} = \begin{cases} e(\tau)^{(k,\ell)}, & \rightarrow e(\tau)^{(k,\ell)} \geq 0; \\ -e(\tau)^{(k,\ell)}, & \rightarrow e(\tau)^{(k,\ell)} < 0. \end{cases}$$

Отсюда значения элементов  $e(\tau)^{(k,\ell)}$  в общем случае будут изменяться в диапазоне  $e_{\min}^{(k,\ell)} \leq e(\tau)^{(k,\ell)} \leq e_{\max}^{(k,\ell)}$ . Здесь приняты такие обозначения:  $e_{\min}^{(k,\ell)}$  – минимальное значение элемента в ДОС, вычисляемое как:  $e_{\min}^{(k,\ell)} = \min_{2 \leq \tau \leq T} \{e(\tau)^{(k,\ell)}\}$ ;  $e_{\max}^{(k,\ell)}$  – максимальное значение элемента в ДОС, которое определяется по формуле  $e_{\max}^{(k,\ell)} = \max_{2 \leq \tau \leq T} \{e(\tau)^{(k,\ell)}\}$ .

Соответственно, динамический диапазон  $\bar{d}_e^{(k,\ell)}$  с учетом разброса в области положительных и отрицательных значений будет равен  $\bar{d}_e^{(k,\ell)} = |e_{\min}^{(k,\ell)} - e_{\max}^{(k,\ell)}| + 1$ .

Максимальная битовая интенсивность  $V(T-1)_e^{(k,\ell)}$  синтаксического представления ДОС на основе структурно-комбинаторного подхода для заданного значения динамического диапазона  $\bar{d}_e^{(k,\ell)}$  оценивается по формуле

$$V(T-1)_e^{(k,\ell)} = ((T-1)[\log_2 \bar{d}_e^{(k,\ell)}] + 1).$$

Анализ полученного выражения позволяет заключить, что поскольку динамический диапазон ДОС не превышает значения динамических диапазонов для спектрограммы и для последовательности элементов исходных кадров группы, то в результате предложенного подхода создаются условия для повышения эффективности синтаксического представления динамического ВИР.

Однако необходимо выделить недостатки такого подхода относительно формирования синтаксического представления ДОС, которые связаны с тем, что:

- длина  $(T-1)$  дифференциально описанной спектрограммы выбирается заранее и является фиксированной на локальном промежутке видеопотока;
- динамический диапазон элементов ДОС определяется по факту обработки фиксированного количества межтрансформантных разностей.

Это может привести к потере адаптивности относительно изменения структурных и статистических характеристик кадров группы. В результате увеличивается битовая интенсивность потока Р-кадров.

Для устранения недостатков *предлагается* разрабатывать направление, базирующееся на дополнительном выявлении закономерностей, основанных на учете локально-структурных свойств для дифференциально описанных спектрограмм в направлении временной оси. Данный вариант базируется на дополнительном выявлении закономерностей, основанных на учете локально-структурных свойств ДОС для последовательности смежных видеокад-

ров. Такое свойство заключается в наличии закономерности относительно ограниченного значения градиента между значениями соседних элементов ДОС.

Величина градиента  $g(\tau)^{(k,\ell)}$  между информативными элементами ДОС задается следующей формулой:

$$g(\tau)^{(k,\ell)} = |e(\tau+1)_{\text{inf}}^{(k,\ell)} - e(\tau)_{\text{inf}}^{(k,\ell)}|c,$$

где  $e(\tau)_{\text{inf}}^{(k,\ell)}$ ,  $e(\tau+1)_{\text{inf}}^{(k,\ell)}$  – информативные элементы  $(k; \ell)$ -й интерполированной ДОС соответственно расположенных в ней на  $\tau$ -й и  $(\tau+1)$ -й позициях;  $g(\tau)^{(k,\ell)}$  – отклонение (градиент) между значениями  $(k; \ell)$ -й ДОС для  $\tau$ -й и  $(\tau+1)$ -й трансформант.

По результатам такой обработки для всех информативных элементов интерполируемой  $(k; \ell)$ -й ДОС, т.е.  $\tau = \overline{1, n_{\text{inf}}}$  формируется вектор градиента  $G^{(k,\ell)}$ . Это задается следующим соотношением:  $G^{(k,\ell)} = \{g(1)^{(k,\ell)}, \dots, g(\tau)^{(k,\ell)}, \dots, g(n_{\text{inf}} - 1)^{(k,\ell)}\}$ , где  $n_{\text{inf}}$  – количество информативных элементов в ИДОС.

Недостатком здесь является то, что мерность градиентного пространства совпадает с мерностью информативной ДОС. Значит, требуется сократить мерность вектора  $G^{(k,\ell)}$ . В этом направлении *предлагается* для вектора  $G^{(k,\ell)}$  градиента информативной ДОС формировать такую характеристику как величина  $g(\text{max})^{(k,\ell)}$  динамического диапазона его компонент. Величина  $g(\text{max})^{(k,\ell)}$  находится по формуле  $0 \leq g(\tau)^{(k,\ell)} \leq g(\text{max})^{(k,\ell)}$ .

Здесь  $g(\text{max})^{(k,\ell)}$  определяется как  $g(\text{max})^{(k,\ell)} = \max_{1 \leq \tau \leq n_{\text{inf}}} \{g(\tau)^{(k,\ell)}\}$ .

Данная формула учитывает в процессе определения обобщенного градиентного ограничения только информативные элементы. Это позволяет сузить величину разброса между компонентами ДОС, т.е. уменьшить значение градиентного ограничения. В связи с этим, в дальнейшем предлагается использовать сокращенное описание градиентного пространства. Тогда для информативных элементов ДОС будут выполняться ограничения, заданные следующей системой соотношений:

$$e(\tau)_{\text{inf}}^{(k,\ell)} - g(\text{max})^{(k,\ell)} \leq e(\tau+1)_{\text{inf}}^{(k,\ell)} \leq e(\tau)_{\text{inf}}^{(k,\ell)} + g(\text{max})^{(k,\ell)}, \quad \tau = \overline{1, n_{\text{inf}}}. \quad (1)$$

Величина обобщенного градиентного ограничения  $g(\text{max})^{(k,\ell)}$  вычисляется для каждой информативной дифференциально описанной спектрограммы. Это обеспечивает адаптивную подстройку характеристики градиентного пространства под особенности локально-структурных закономерностей межтрансформантных последовательностей (слотов) на каждой частотной составляющей.

Для определения битовой интенсивности созданного синтаксического представления ДОС с учетом ее интерполирования и выявления градиентных ограничений необходимо оценить соответствующую информативность. В соответствии со структурно-комбинаторным подходом для определения степени информативности базовым понятием является количество  $Q(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e$  различных информативных ДОС, длиной  $n_{\text{inf}}$ , которое можно составить в случае наложения ограничений на динамические диапазоны  $\overline{d}(\tau)_{\text{inf}}^{(k,\ell)}$  изменения значений их элементов  $e(\tau)_{\text{inf}}^{(k,\ell)}$  в условиях выявления локально-структурных закономерностей.

Для определения величин  $\overline{d}(\tau)_{\text{inf}}^{(k,\ell)}$  элементов  $e(\tau)_{\text{inf}}^{(k,\ell)}$  информативной дифференциальной по описанной спектрограммы  $E_{\text{inf}}^{(k,\ell)}$  используется выражение (1). Анализ данного выражения позволяет заключить следующее:

1. Динамический диапазон первого элемента  $e(1)_{\text{inf}}^{(k,\ell)}$ ,  $\tau=1$ , оценивается из условия независимости от значений последующих элементов информативной ДОС. Тогда его динамический диапазон  $\bar{d}(1)_{\text{inf}}^{(k,\ell)}$  будет равен общему динамическому диапазону  $\bar{d}_e^{(k,\ell)}$  последовательности  $E_{\text{inf}}^{(k,\ell)}$ , т.е.

$$\bar{d}(1)_{\text{inf}}^{(k,\ell)} = \bar{d}_e^{(k,\ell)} = |e_{\text{min}}^{(k,\ell)} - e_{\text{max}}^{(k,\ell)}| + 1. \quad (2)$$

2. Величина  $\bar{d}(\tau)_{\text{inf}}^{(k,\ell)}$  динамического диапазона остальных элементов ДОС,  $\tau = \overline{2, n_{\text{inf}}}$ , в соответствии с формулой (2) будет формироваться с учетом градиента относительно предыдущего значения. Поэтому величина  $\bar{d}(\tau)_{\text{inf}}^{(k,\ell)}$  будет оцениваться как

$$\bar{d}(\tau)_{\text{inf}}^{(k,\ell)} = 2 g(\text{max})^{(k,\ell)} + 1, \quad \tau = \overline{2, n_{\text{inf}}}. \quad (3)$$

В таком случае динамический диапазон элементов информативной ДОС на позициях  $2 \leq \tau \leq n_{\text{inf}}$  полностью определяется значением  $g(\text{max})^{(k,\ell)}$  обобщенной характеристики градиентного пространства.

Значит, на максимальные абсолютные значения (спецификации) элементов  $e(\tau)_{\text{inf}}^{(k,\ell)}$  последовательности  $E_{\text{inf}}^{(k,\ell)}$  в градиентном пространстве  $G^{(k,\ell)}$  наложены ограничения, записываемые следующей системой неравенств:

$$|e(\tau)_{\text{inf}}^{(k,\ell)}| = \begin{cases} \bar{d}_e^{(k,\ell)}, & \rightarrow \tau=1; \\ 2 g(\text{max})^{(k,\ell)} + 1, & \rightarrow 2 \leq \tau \leq n_{\text{inf}}. \end{cases} \quad (4)$$

Отсюда формулируем следующие определения.

**Определение 1** (комбинаторная трактовка). Комбинации  $E_{\text{inf}}^{(k,\ell)}$ , составленные из элементов  $e(\tau)_{\text{inf}}^{(k,\ell)}$ , удовлетворяющие спецификациям в соответствии с системой соотношений (4), будем называть перестановками с повторениями, на динамические диапазоны элементов которых наложены двухтиповые ограничения.

**Определение 2** (структурная трактовка). Последовательности  $E_{\text{inf}}^{(k,\ell)}$ , элементы которых имеют ограничения на динамический диапазон в соответствии с выражениями (2) и (3), будем называть дифференциально описанными спектрограммами с двухкомпонентными локально-структурными ограничениями (двухкомпонентными градиентными ограничениями).

С учетом этого величина  $Q(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e$ , определяемая как количество различных дифференциально-описанных спектрограмм с двухкомпонентными градиентными ограничениями  $\{ \bar{d}_e^{(k,\ell)}; g(\text{max})^{(k,\ell)} \}$ , будет оцениваться с помощью такого выражения:

$$Q(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e = \prod_{\tau=1}^{n_{\text{inf}}} \bar{d}(\tau)_{\text{inf}}^{(k,\ell)} = \bar{d}_e^{(k,\ell)} \cdot (g(\text{max})^{(k,\ell)})^{n_{\text{inf}} - 1}.$$

Отсюда количество двоичных разрядов  $V(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e$  на синтаксическое представление информативной ДОС в двухкомпонентном градиентном пространстве ограничений определяется по формуле

$$V(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e = [\ell \log_2 \bar{d}_e^{(k,\ell)}] + (n_{\text{inf}} - 1) [\ell \log_2 (g(\text{max})^{(k,\ell)})] + 2.$$

Полученное выражение позволяет оценить максимальную битовую интенсивность  $V(n_{\text{inf}}; g(\max)^{(k,\ell)})_e$  синтаксического представления информативной ДОО на основе структурно-комбинаторного подхода с учетом ее интерполирования и выявления двухкомпонентных градиентных ограничений  $\{ \bar{d}_e^{(k,\ell)}; g(\max)^{(k,\ell)} \}$ .

#### 4. Выводы

1. Разработана модель оценки битовой интенсивности слотов Р-кадров на основе эффективного синтаксического представления информативных ДОО с учетом обобщенной характеристики градиентного пространства локально-структурных (градиентных) ограничений. Такой тип структурных закономерностей обусловлен целым рядом причин, а именно: наличие коррелированности между видеокадрами; незначительные изменения освещенности объектов видеосцен; зависимости между компонентами ДКП соседних трансформант, характерные в области как низких, так и высоких частотных составляющих; предварительное снижение динамического диапазона элементов спектрограмм в результате их дифференциального описания.

**Научная новизна.** Впервые создана модель оценки информативности эффективного синтаксического представления слотов Р-кадров на основе обработки дифференциально описанных спектрограмм. Отличия заключаются в том, что сокращается межтрансформантная (временная) избыточность, обусловленная наличием таких закономерностей: межкадровая (временная) психовизуальная избыточность на уровне обработки дифференциально описанных спектрограмм, что обусловлено особенностями ограниченной чувствительности зрительной системы относительно коррекции отдельных частотных составляющих; межкадровая (временная) структурная избыточность, обусловленная наличием локально-структурных ограничений между соседними информативными элементами дифференциально описанной спектрограммы. Это обеспечивает оценку минимального уровня снижения битовой интенсивности видеопотока при сохранении требуемой целостности информационного ресурса.

**Список литературы:** 1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / В.Г. Олифер, Н.А. Олифер. СПб.: Питер, 2006. 958 с. 2. Миано Дж. Форматы и алгоритмы сжатия изображений в действии: учебное пособие / Дж. Миано. Пер. с англ. М.: Триумф, 2003. 336 с. 3. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. М: Техносфера, 2004. 368 с. 4. Баранник В.В. Структурно-комбинаторное представление данных в АСУ / В.В. Баранник, Ю.В. Стасев, Н.А. Королева. Х.: ХУПС, 2009. 252 с. 5. Акимов Р.И. Технология кодирования пакетов предсказанных кадров в инфокоммуникационных системах // Сучасна спеціальна техніка. 2012. № 4. С. 17 – 18.

*Поступила в редколлегию 11.09.2015*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры Харьковского университета Воздушных Сил им. И. Кожедуба. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

**Шульгин Сергей Сергеевич**, соискатель Киевского Национального авиационного университета, г. Киев. Научные интересы: обработка видеоинформации и безопасность инфокоммуникационных систем. Адрес: Украина, 01000, Киев, пр. Космонавта Комарова, 1.

## СЕЛЕКТИВНИЙ МЕТОД ШИФРУВАННЯ ВИДЕОПОТОКУ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ ПРИХОВУВАННЯ БАЗОВОГО І-КАДРУ

Розглядається селективний метод шифрування відеокадрів, заснований на приховуванні базового І-кадру. Робота даного методу базується на основі обробки групи кадрів з урахуванням алгоритму MPEG, який реалізований за принципом формування послідовності відеокадрів різних типів. Представляються алгоритми і схеми кодування і декодування відеопотоку із застосуванням даного методу. Розробляється метод оцінки обсягу прихованого І-кадру і його стисненого подання без приховування щодо групи кадрів в процентному співвідношенні. Аналізується зміна обсягу стисненого уявлення групи кадрів з прихованим базовим кадром щодо стисненого початкового об'єму групи кадрів залежно від різних значень пікового відношення сигнал / шум для різних типів кадрів. Розробляється методологічна база для визначення різниці між обсягами стисненого уявлення групи кадрів із застосуванням приховування базового І-кадру і без приховування в процентному співвідношенні. Ключові слова: *відеокадр, група кадрів, обсяг кадру, шифрування, коефіцієнт стиснення, відношення сигнал/шум, зображення середньої насиченості, селективний метод.*

### Вступ

Інтенсивні розробки систем телебачення високої роздільної здатності HDTV та стрімкий розвиток мультимедіа додатків останнім часом різко збільшили частку відеотрафіка в телекомунікаційних системах. Впровадження таких технологій набирає масову популярність серед користувачів Інтернету, в комерційних та державних організаціях. Тому виникає необхідність у розвитку механізмів та методів для організації конфіденційності інформації в телекомунікаційних системах (організація безпечного з'єднання та захист потокового відео, відеоконференцз'язку) [1-3]

Існуючі технології приховування відеоінформаційних ресурсів забезпечують необхідну конфіденційність. Однак вони мають істотний недолік: їх робота заснована на закритті усього потоку інформації, що передається незалежно від типу та змісту відеосцени. Такий підхід закриття інформації називається повним. Його використання для відкритих відеоінформаційних ресурсів в інфокомунікаційних системах реального часу є непрактичним. Це обумовлено такими причинами: уся структура відеоданих руйнується; у разі виникнення помилки в каналі передачі даних збільшується час обробки. Для вирішення цієї проблеми застосовується селективний підхід шифрування. Його суть полягає в приховуванні найбільш значущих компонент відеопотоку. Ці компоненти формуються в процесі стиснення відеоданих. Тому таке шифрування відноситься до селективного. Реалізація селективного підходу приховування можлива на різних рівнях формування MPEG-потoku. Ієрархія потоку включає в себе кілька рівнів: власне сам відеопотік (sequence), група кадрів (GOP -Group Of Pictures), слайс (slice), макроблок (macroblock) та блок (block). Пропонується закрити тільки базовий І-кадр. Це дозволить зменшити обсяг та час обробки шифрованих стислих відеоданих.

### Розробка селективного методу шифрування відеопотоку

З урахуванням того, що вихідні обсяги всіх кадрів до обробки рівні, то формула для визначення обсягу стисненого представлення  $V_{гк}^{(сж)}$  групи кадрів має вигляд [4]:

$$V_{гк}^{(сж)} = V_k \left( \frac{N_I}{K_I} + \frac{N_P}{K_P} + \frac{N_B}{K_B} \right), \quad (1)$$

де  $V_k$  – обсяг вихідного відеокадру;  $N_I$  – кількість І-кадрів у групі кадрів;  $N_P$  – кількість Р-кадрів у групі кадрів;  $N_B$  – кількість В-кадрів у групі кадрів;  $\kappa_I$  – коефіцієнт стиснення для І-кадрів;  $\kappa_P$  – коефіцієнт стиснення для Р-кадрів;  $\kappa_B$  – коефіцієнт стиснення для В-кадрів.

У випадку з закриттям І-кадру його структура руйнується – знижується потенційна кількість статистичної, психовізуальної та структурної надлишковості, аж до нульового рівня. У результаті цього зменшується коефіцієнт стиснення  $\kappa_{I,3}$  для І-кадру. Обсяг закритого І-кадру визначається величиною  $V_{I,3}^{(сж)}$ . Відповідно, обсяг стисненого представлення групи кадрів  $V_{гк,3}^{(сж)}$ , що містить закритий І-кадр, розраховується за формулою:

$$V_{гк,3}^{(сж)} = V_k \left( \frac{N_I}{\kappa_{I,3}} + \frac{N_P}{\kappa_P} + \frac{N_B}{\kappa_B} \right). \quad (2)$$

Оцінимо зміни обсягу групи кадрів з приховуванням  $V_{гк,3}^{(сж)}$  та без приховування  $V_{гк}^{(сж)}$  базового І-кадру. Для цього необхідно знати коефіцієнти стиснення для реалістичного середньонасиченого кадру при певних значеннях PSNR пікового відношення сигнал/шум.

Для розрахунку ступеня стиснення різних типів кадрів використовуються такі значення:

– для визначення коефіцієнта стиснення  $\kappa_I$  для І-кадрів використовується пікове відношення сигнал/шум PSNR = 50; 45; 40 дБ. Великі значення задаються для збереження високого деталювання І-кадрів. Тому для них задаються таблиці квантування з найменшими коефіцієнтами. Також І-кадри мають найменшу ступінь стиснення через застосування алгоритмів тільки всередині кадрового передбачення;

– для визначення коефіцієнта стиснення  $\kappa_P$  для Р-кадрів використовується пікове відношення сигнал/шум PSNR = 40; 35; 30 дБ. Це пов'язано з тим, що Р-кадри несуть менше візуального навантаження. При їх кодуванні застосовуються алгоритми компенсації руху та міжкадрового передбачення за попередніми І- або Р-кадрами. Для Р-кадрів задаються таблиці квантування з середніми коефіцієнтами;

– для визначення коефіцієнта стиснення  $\kappa_B$  для В-кадрів використовується пікове відношення сигнал/шум PSNR = 30; 25; 23 дБ. В процесі їх формування застосовуються алгоритми компенсації руху і двонаправленого передбачення за попередніми та наступними І- або Р-кадрами. Для них задаються таблиці квантування з найбільшими коефіцієнтами. Тому ступінь стиснення для В-кадрів є максимальною в порівнянні з іншими типами відеокадрів.

Проведемо розрахунок обсягу  $\Delta V_I^{(сж)}$  стисненого представлення І-кадру щодо обсягу групи кадрів у процентному співвідношенні з урахуванням виразів (1) і (2):

$$\Delta V_I^{(сж)} = \frac{100\%}{1 + \frac{\kappa_I}{\kappa_P} N_P + \frac{\kappa_I}{\kappa_B} N_B}. \quad (3)$$

Значення обсягу  $\Delta V_{I,3}$  прихованого І-кадру щодо обсягу груп кадрів у процентному співвідношенні розраховується так:

$$\Delta V_{I,3} = \frac{100\%}{1 + \frac{\kappa_{I,3}}{\kappa_P} N_P + \frac{\kappa_{I,3}}{\kappa_B} N_B}. \quad (4)$$

Значення приросту обсягу  $V(h_I; h_P; h_B)$  прихованого кадру щодо обсягу стисненого неприхованого кадру в групі кадрів у процентному співвідношенні розраховується за формулою:

$$V(h_I; h_P; h_B) = \Delta V_I^{(сж)} - \Delta V_{I,3}. \quad (5)$$

На рис. 1 зображена діаграма залежності значень обсягів стисненого представлення I-кадру та значень обсягів прихованого I-кадру щодо обсягу групи кадрів у процентному співвідношенні.

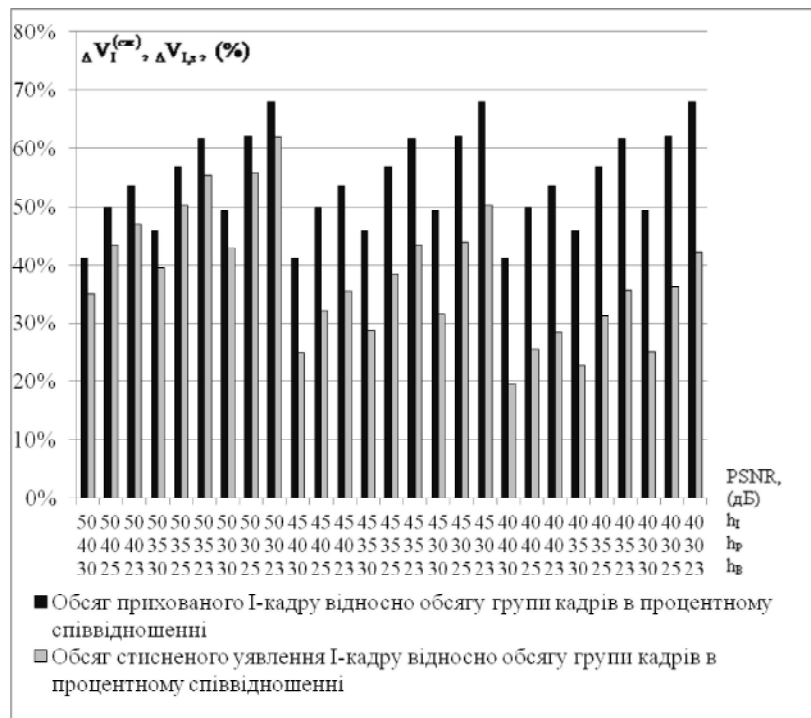


Рис. 1. Діаграма значень обсягу  $\Delta V_{I,3}$  стисненого представлення I-кадру і прихованого обсягу  $\Delta V_I^{(сж)}$  I-кадру залежно від пікового відношення сигнал/шум в групі кадрів у процентному співвідношенні

З розрахунків видно, що приріст обсягу прихованого кадру у процентному співвідношенні щодо обсягу стисненого неприхованого кадру в групі кадрів залежно від значень PSNR становить 7 - 25%. Це обумовлено утаєнням базового I-кадру щодо варіанту стиснення I-кадру без приховування.

З аналізу діаграми на рис. 1 видно, що обсяг прихованого I-кадру більше обсягу стисненого представлення I-кадру щодо обсягу групи кадрів у процентному співвідношенні. Також видно, що зі зниженням значень пікового відношення сигнал/шум вага (у відсотках) прихованого I-кадру в групі кадрів збільшується. Це пов'язано з тим, що при низьких значеннях PSNR для P- та B-кадрів ступінь стиснення збільшується, а для прихованого I-кадру - ступінь стиснення є постійним ( $\kappa_I = 1$ ). Розрахунки показали, що при високих значеннях PSNR обсяг стисненого представлення I-кадру становить 35% від усього обсягу групи кадрів, а обсяг прихованого I-кадру в групі кадрів склав 42%. При низьких значеннях PSNR обсяг стисненого представлення I-кадру щодо обсягу групи кадрів дорівнює 43%, а прихованого I-кадру - 68%.

У селективному підході прихованню піддається тільки I-кадр, який є базовим та має максимальний обсяг у групі кадрів. Тому зі зменшенням значень пікового відношення сигнал/шум ступінь стиснення для I-кадру буде постійним ( $\kappa_I = 1$ ), а для P та B кадрів ступінь стиснення буде рости. У результаті цього обсяг I-кадру щодо обсягу групи кадрів у процентному співвідношенні з застосуванням селективного шифрування залежно від пікового відношення сигнал/шум може коливатися від 41 до 68%.

Також слід відзначити те, що при використанні алгоритмів шифрування після квантування структура проміжного представлення руйнується. Змінюються структурні характерис-

тики (значення компонент та залежність між ними) в матрицях дискретного косинусного перетворення. Це руйнує імовірнісні та статистичні характеристики, призводить до відсутності ланцюжків нульових значень при зигзаг-скануванні, в результаті чого ступінь стиснення I-кадру зменшується. Тому відмінності між значеннями обсягів стисненого представлення групи кадрів з утаєнням базового I-кадру та без приховування у процентному співвідношенні стають явно вираженими та можуть досягати 35%.

Для визначення зміни візуальної якості зображень при різних режимах обробки проведемо оцінку середніх значень пікового відношення сигнал/шум  $PSNR(8)_{cp}$  для 8 відеокадрів середньонасичених зображень, які визначаються так:

$$PSNR(8)_{cp} = \frac{PSNR(K_I) + 2 \cdot PSNR(K_B) + 5 \cdot PSNR(K_P)}{8} \quad (6)$$

Залежність середнього значення пікового відношення сигнал/шум у групі з 8 кадрів для середньонасичених зображень від коефіцієнтів стиснення при різних режимах обробки розрахована за формулою (6) та представлена у вигляді діаграми на рис. 2.

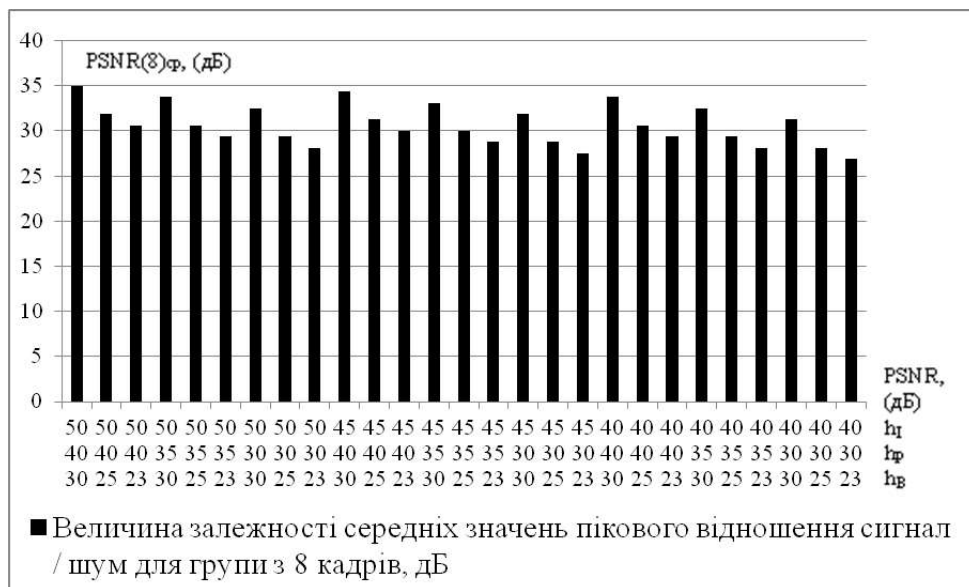


Рис. 2. Середні значення  $PSNR(8)_{cp}$  по групі з 8 кадрів для різних режимів обробки кадрів у середньонасичених зображеннях

З аналізу діаграми на рис. 2 видно, що:

- середні значення для групи з 8 кадрів при різних режимах обробки кадрів у середньонасичених зображеннях коливаються в межах 27-35дБ. Це свідчить про значні зміни якості зображень при різних режимах обробки;
- при зміні PSNR для I-кадру на 5дБ середнє пікове значення відношення сигнал/шум для групи кадрів зменшується на 1дБ. При цьому такі зміни незначно впливають на обсяг стисненого представлення групи кадрів;
- на середній PSNR групи кадрів суттєво впливають значення відношення сигнал/шум для P- та B-кадрів.

На рис. 3 зображена діаграма приросту обсягу  $D(h_I; h_P; h_B)$  стисненого представлення групи кадрів у процентному співвідношенні з застосуванням приховування базового I-кадру та без приховування залежно від середнього пікового відношення сигнал/шум для середньонасичених зображень.



Рис. 3. Діаграма приросту  $D(h_I; h_P; h_B)$  стисненого представлення обсягу групи кадрів з прихованим I-кадром відносно стисненого представлення обсягу групи кадрів без приховування в процентному співвідношенні з урахуванням середніх значень пікового відношення сигнал/шум для середньонасичених зображень

На діаграмі (див.рис.3) чітко спостерігається збільшення приросту обсягу стисненого представлення обсягу  $V_{ГК}^{сж,3}$  групи кадрів з прихованим базовим I-кадром по відношенню до стисненого подання обсягу  $V_{ГК}^{сж}$  групи кадрів без приховування зі зменшенням середнього пікового відношення сигнал/шум для групи кадрів. Видно, що при зменшенні середнього PSNR для групи кадрів на 9 дБ приріст обсягу  $D(h_I; h_P; h_B)$  стисненого представлення  $V_{ГК}^{сж,3}$  групи кадрів з прихованим базовим I-кадром по відношенню до стисненого подання обсягу  $V_{ГК}^{сж}$  групи кадрів без приховування досягає 35%.

### Висновки

1. Розроблено метод селективного шифрування в процесі стиснення відеопотоку, заснований на приховуванні базового I-кадру. Робота даного методу базується на основі обробки групи кадрів, з урахуванням алгоритму MPEG, який реалізований за принципом формування послідовності відеокадрів різних типів. В результаті його роботи ховається весь відеопотік при шифруванні від 8% до 15% початкового об'єму відеоданих. Даний метод шифрування застосовується після етапу квантування. Наукова новизна - пропонується метод приховування відеопотоку, заснований на шифруванні тільки базового I-кадру. Це дозволяє забезпечити приховування групи кадрів в умовах мінімізації втрат за ступенем стиснення. Він враховує ступінь стиснення в процесі кодування в залежності від пікового відношення сигнал/шум. У роботі представлена схема і алгоритм кодування відеопотоку в селективному підході з приховуванням I-кадру.

2. Розроблено метод оцінки обсягу прихованого I-кадру та його стисненого подання без приховування щодо групи кадрів у процентному співвідношенні. Проведено аналіз зміни обсягу стисненого представлення I-кадру та прихованого обсягу I-кадру щодо групи кадрів у процентному співвідношенні. Його результати показали, що залежно від пікового відношення сигнал/шум обсяг прихованого I-кадру в порівнянні зі стисненим його поданням щодо групи кадрів збільшується від 7 до 20%.

3. Розроблена методологічна база для визначення різниці між обсягами стисненого представлення групи кадрів із застосуванням приховування базового I-кадру та без приховування у процентному співвідношенні. Проведено розрахунки щодо зміни обсягу стиснено-

го представлення групи кадрів з прихованим I-кадром та без його приховування. Вони показали, що зі зменшенням пікового відношення сигнал/шум обсяг стисненого представлення групи кадрів з прихованим I-кадром збільшується з 9 до 44% відносно обсягу стисненого представлення групи кадрів без приховування I-кадру.

**Список літератури:** 1. *Ватолін Д., Ратушняк А., Смирнов М., Юкин В.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: Диалог-Мифи, 2003. 381 с. 2. *Ян Ричардсон.* Видеокодирование. H.264 и MPEG-4 — стандарты нового поколения. М.: Техносфера, 2005. 368 с. 3. *Баранник В.В.* Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 212 с. 4. *Баранник В.В.* Методологическая база для управления битовой скоростью видеопотока в процессе компрессии / Баранник В.В. // Праці УНДІРТ. 2013. 22с.

*Надійшла до редколегії 09.09.2015*

**Бараннік Володимир Вікторович**, д-р техн. наук, професор, начальник кафедри Харківського університету Повітряних Сил ім. І. Кожедуба. Наукові інтереси: системи, технології перетворення, кодування, захисту та передачі інформації, семантичної обробки зображень. Адрес: Україна, 61023, Харків, 23, вул. Сумська, 77/79.

**Комолов Дмитро Іванович**, пошукач Харківського національного університету радіоелектроніки. Наукові інтереси: системи, технології перетворення, кодування, захисту та передачі інформації, семантичної обробки зображень. Адрес: Україна, 61166, Харків, пр. Науки, 14.