



## МЕТОД СТРУКТУРНОГО СИНТЕЗУ МЕРЕЖЕВИХ ПРИКЛАДНИХ ПРОЦЕСОРІВ

І. Майків <sup>1)</sup>, А. Степаненко <sup>1)</sup>, Д. Вобшал <sup>2)</sup>

- 1) Науково дослідний інститут Інтелектуальних комп'ютерних систем, Тернопільський національний економічний університет, 46009 Україна, м. Тернопіль, вул. Львівська 11, mim@tanet.edu.te.ua, {andrew.stepanenko@gmail.com}, http://www.ics.tneu.edu.ua/  
2) Esensors Inc., 4240 Ridge Lea Road Suite 37 Amherst, NY 14226, USA 14226, darold@wobschall.com, http://www.eesensors.com/

**Резюме:** на основі методу морфологічного аналізу та синтезу запропоновано метод структурного синтезу мережеских прикладних процесорів. Метод включає етапи функціонального аналізу, структурного синтезу, та пошуку множини оптимальних рішень і поєднує лексикографічний критерій переваги (L-критерій) для відбору електронних компонентів на етапі функціонального аналізу та безумовний критерій переваги (оптимальності по Парето,  $\pi$ -критерій) на етапі пошуку множини оптимальних рішень, що розглядаються в літературі, як альтернативні методи пошуку оптимальних рішень. Поєднання L та  $\pi$ -критеріїв дозволяє зменшити число синтезованих альтернативних варіантів на етапі структурного синтезу, а всі отримані рішення є допустимими.

**Ключові слова:** структурний синтез, морфологічний аналіз, лексикографічний критерій переваги, безумовний критерій переваги.

### ВСТУП

Сучасні вимірювально-керуючі системи (ВКС) реалізуються як багаторівневі системи із розподіленими обчислювальними ресурсами на базі промислових шин (інтерфейсів). Для їх уніфікації, національним інститутом стандартів США (NIST), прийнято серію стандартів IEEE-1451, що регламентує вимоги до апаратного та програмного забезпечення ВКС. Також, введено означення мережевого прикладного процесора (МПП) – вузла обробки даних, який займає проміжний рівень між сервером мережі (верхній рівень) і сенсорами та виконавчими механізмами (нижній рівень). Основними функціями МПП є: 1) поточна обробка даних; 2) підтримка набору послідовних інтерфейсів, по яких здійснюється обмін даними як із вузлами нижнього рівня, так і з сервером системи. Однак кожна область застосування характеризується своєю множиною функціональних задач (ФЗ), що вимагає застосування спеціалізованих МПП, технічні параметри яких оптимальні для даної області, а отже створення методу структурного синтезу МПП, оптимальних за функціонально-вартісними показниками.

### 1. АНАЛІЗ ВІДОМИХ МЕТОДИК СТРУКТУРНОГО СИНТЕЗУ

Задача оптимального структурного синтезу є однією із традиційних задач, загальні підходи до вирішення якої розглянуто в [1-7], більшість із яких базується на методах дискретної оптимізації (віток та границь, морфологічного синтезу) В [8] представлено метод структурного синтезу оптимальних апаратних структур багатопроекторних інформаційних систем, який включає етапи аналізу задач, що вирішує система та саме структурного синтезу за результатами аналізу. Однак запропонований метод орієнтований на реалізацію систем на базі застарілих мікропроцесорів (МП) серії K580, K581 та ін. Виходячи із їх обмеженої продуктивності, його цільова функція полягає в визначенні необхідного числа МП та оптимальному розподілі між ними множини ФЗ, що реалізує система.

Однак сучасні МП та мікроконтролери (МК) характеризуються високою продуктивністю (швидкістю виконання операцій), що лежить в межах від одиниць до сотень мільйонів операцій за секунду (MIPS), і цільова функція нового методу оптимального структурного синтезу

повинна забезпечити вибір МК, продуктивність якого буде достатньою для вирішення множини ФЗ. Тому метод запропонований в [8], не може безпосередньо використовуватись в процесі структурного синтезу МПП і потребує вдосконалення.

Альтернативний підхід до створення компонентів дистрибутивних комп'ютерних систем, за функціонально-вартісними показниками запропоновано в [9], який базується на методі морфологічного аналізу та синтезу [10, 11]. В процесі проектування системи розглядається ряд альтернативних рішень, що в цілому дозволяє зробити вибір між: 1) необхідністю нової розробки; 2) адаптацією вже відомих рішень; 3) компонованням існуючих програмно-апаратних вузлів. Однак даний метод не містить етапу попереднього аналізу ФЗ об'єкта, який проектується, що не дозволяє об'єктивно оцінити вимоги як до апаратних так і програмних ресурсів, необхідних для ефективного функціонування ВКС. Вибір варіантів вузлів здійснюється на основі заданих технічних обмежень (ТО), які задовольняють технічне завдання (ТЗ). Також в процесі пошуку множини оптимальних рішень виконується повний перебір всіх можливих комбінацій вузлів, що часто неприйнятно збільшує час пошуку оптимальних рішень.

В [12] запропоновано метод наскрізного проектування портативних приладів та спеціалізованих мікропроцесорних систем на базі МК. Однак процес проектування орієнтовано на МК лише одного типу і не проводиться оцінка ефективності багатоваріантної реалізації на базі МК як різної архітектури так і виробників.

Таким чином, як показали результати аналізу, на даний час відсутній метод оптимального структурного синтезу цифрових систем на базі МК, що характеризуються широким набором функціональних можливостей та значною різницею в ціні.

Метою роботи є створення методу оптимального структурного синтезу МПП на базі МК виходячи із набору (множини) їх функціональних показників, що можуть виступати як критерії оцінки якості.

## 2. ЗАГАЛЬНІ ПІДХОДИ ДО ПРОЦЕСУ СТРУКТУРНОГО СИНТЕЗУ МПП

Метод структурного синтезу МПП, оптимальних за функціонально-вартісними показниками повинен включати етапи: 1) функціонального аналізу; 2) структурного синтезу; 3) пошуку структур оптимальних за

функціонально-вартісними показниками.

Узагальнений алгоритм запропонованого методу (рис.1) включає 8 етапів, що умовно розділені на дві групи: функціонального аналізу та структурного синтезу.

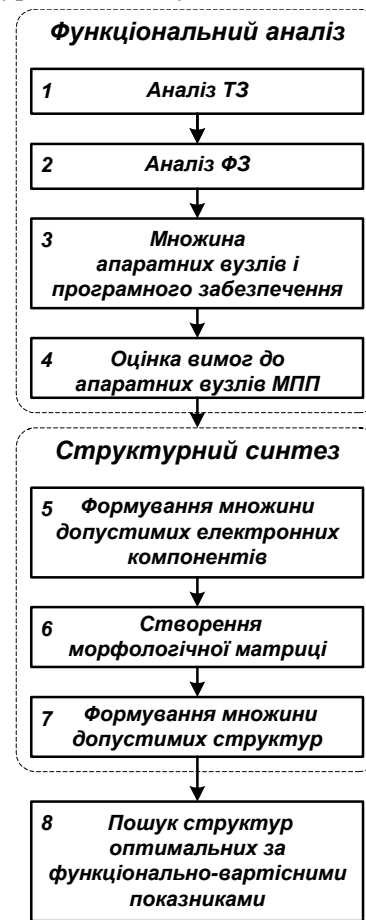


Рис. 1 – Алгоритм структурного синтезу мережевих прикладних процесорів

## 3. ФУНКЦІОНАЛЬНИЙ АНАЛІЗ МПП

На першому етапі процесу проектування виконується аналіз ТЗ на МПП, що дозволяє визначити перелік ФЗ, які він реалізує. В цілому ФЗ поділяють на системні та прикладні. Системні ФЗ реалізуються операційною системою або спеціалізованим програмним забезпеченням і забезпечують взаємодію прикладних ФЗ, визначають порядок їх виконання, розподіл пам'яті між ними і т.д. Прикладні ФЗ виконують обслуговування зовнішніх пристроїв та користувачів. Із загальної множини прикладних ФЗ наведеної в [8], МПП реалізує лише два класи: 1) передачі даних; 2) обробки даних.

Кількість прикладних ФЗ, що реалізує МПП, визначається множиною  $I$ , що уточнюється в процесі аналізу ТЗ на МПП

$$I = \{i : i = \overline{1, I}\}, \quad (1)$$

де  $I$  – загальне число ФЗ,  $i$  – порядковий номер ФЗ.

На другому етапі виконується аналіз ФЗ. В результаті аналізу для кожної ФЗ визначається множина інформаційних параметрів, необхідних для оцінки об'єму вхідних/вихідних даних  $\{l_i, \tau_i, n_i, t_i\}$ , де: 1)  $l_i$ , – середній розмір вхідних повідомлень; 2)  $\tau_i$  – періодичність рішення  $i$ -ї ФЗ; 3)  $n_i$  – середня кількість повідомлень, що поступає для  $i$ -ї ФЗ, за період  $\tau_i$ ; 4)  $t_i$  – час вирішення  $i$ -ї ФЗ. В результаті для множини  $I$  формується множина векторів (2), які задають набір функціональних обмежень для МПП

$$\langle l_i \rangle, \langle \tau_i \rangle, \langle n_i \rangle, \langle t_i \rangle, i = \overline{1, I}. \quad (2)$$

Ефективність реалізації МПП  $i$ -ї ФЗ визначається наступною умовою

$$t_i < \tau_i. \quad (3)$$

Будь-яка ФЗ є багатоетапним процесом перетворення даних, тому доцільно здійснити розподіл ФЗ на ряд окремих функціональних складових – функцій системи (ФС), кожна із яких реалізує один із етапів обробки даних, а взаємозв'язки між ними відображають процес перетворення даних. Перелік всіх ФС створює множину  $J$

$$J = \{j : j = \overline{1, J}\}, \quad (4)$$

де  $J$  – загальне число типів ФС,  $j$  – порядковий номер ФС.

Взаємозв'язок між  $i$ -ю ФЗ та  $j$ -ю ФС представляється у вигляді матриці  $[\xi_{ij}]$ ,  $i = \overline{1, I}$ ,  $j = \overline{1, J}$ , в рядки якої записується перелік ФЗ, що реалізує МПП, а у стовпці – перелік всіх ФС. Кожен елемент матриці  $[\xi_{ij}]$  є двійковим числом, що дорівнює 1, якщо для реалізації  $i$ -ї ФЗ необхідно ФС  $j$ -го типу, і 0 в протилежному випадку. Сукупність всіх ФС, необхідних для реалізації  $i$ -ї ФЗ, формує множину  $J_i$ , що належить множині  $J$ :  $J_i \in J$ . Також, через набір попередньо уточнених інформаційних параметрів для  $i$ -ї ФЗ  $\{l_i, \tau_i, n_i, t_i\}$ . визначаються параметри ФС: 1) середня довжина вхідних повідомлень –  $l_{ij}$  (біт); 2) інтенсивність вхідних повідомлень –  $\lambda_{ij}$  (повідомлення/сек.). В [8] наведено перелік основних типів ФС та аналітичні залежності для обчислення  $l_{ij}$  і  $\lambda_{ij}$ .

На третьому етапі визначається необхідне апаратне забезпечення (АЗ) та програмне забезпечення (ПЗ), що реалізує окремі ФС. Перелік всіх типів АЗ уточнюється в процесі аналізу ТЗ і створює множину  $H$

$$H = \{h : h = \overline{1, H}\}, \quad (5)$$

де  $H$  – загальна кількість типів апаратних засобів.

Множину  $H$  розділяють за функціональним призначенням на ряд підмножин: 1) МК – множина  $H^{MC}$ ; 2) пристроїв вводу/виводу – множина  $H^{IF}$ ; 3) постійних запам'ятовуючих пристроїв (ПЗП) –  $H^{ROM}$ ; 4) оперативних запам'ятовуючих пристроїв (ОЗП) –  $H^{RAM}$ . В цілому  $H = H^{MC} \cup H^{IF} \cup H^{ROM} \cup H^{RAM}$ .

В подальшому для  $j$ -ї ФС складається матриця відповідності АЗ  $[\xi_{jh}]$ ,  $j = \overline{1, J}$ ,  $h = \overline{1, H}$ , в рядки якої записується перелік ФС, що реалізує МПП, а у стовпці – перелік всіх типів АЗ. Кожний елемент матриці  $[\xi_{jh}]$  є двійковим числом, що дорівнює 1, якщо для реалізації  $j$ -ї ФС необхідно використати апаратне забезпечення  $h$ -го типу, і 0 в протилежному випадку. Сукупність всіх апаратних засобів, необхідних для реалізації  $j$ -ї ФС, формує множину  $H_j$ , що належить множині  $H$ :  $H_j \in H$ .

Аналогічним чином формується перелік необхідного ПЗ, що створює множину  $S$

$$S = \{s : s = \overline{1, S}\}, \quad (6)$$

де  $S$  – загальна кількість модулів ПЗ.

Також формується матриця відповідності ПЗ  $[\xi_{js}]$ ,  $j = \overline{1, J}$ ,  $s = \overline{1, S}$ , в рядки якої записується перелік ФС, що реалізує МПП, а у стовпці – перелік всіх модулів ПЗ. Кожен елемент матриці  $[\xi_{js}]$  є двійковим числом, що дорівнює 1, якщо для  $j$ -ї ФС необхідно використати ПЗ  $s$ -го типу, і 0 в протилежному випадку. Сукупність всіх модулів ПЗ необхідних для реалізації  $j$ -ї ФС, формує множину  $S_j$ , що належить множині  $S$ :  $S_j \in S$ . Множина АЗ та ПЗ, необхідного для реалізації  $i$ -ї ФЗ, визначається наступними співвідношеннями

$$H_i = \sum_{j \in J_i} \sum_{h \in H_j} \xi_{ij} \cdot \xi_{jh}, \quad (7)$$

$$S_i = \sum_{j \in J_i} \sum_{s \in S_j} \xi_{ij} \cdot \xi_{js} \quad (8)$$

На четвертому етапі проводиться аналіз вимог до АЗ, зокрема: 1) продуктивності МК; 2) об'ємів ПЗП та ОЗП; 3) часових обмежень.

### 3.1 ОЦІНКА НЕОБХІДНОЇ ПРОДУКТИВНОСТІ МК

В роботі [13] показано, що МПП доцільно реалізувати як двох або багато процесорну структуру, в якій незалежно, на окремих МК, реалізуються процеси обробки та обміну даними. В цьому випадку, оцінка вимог до АЗ кожного вузла виконується окремо, по загальній методиці, враховуючи особливості їх функціонування.

Сукупність МК, що реалізують МПП, задається множиною  $K = \{k : k = \overline{1, K}\}$ , де  $K$  – кількість використаних МК.

Оцінку необхідної продуктивності МК проведемо, використавши математичний апарат теорії масового обслуговування (ТМО) [14]. В цьому випадку, кожен функціональний вузол можна розглядати як систему масового обслуговування (СМО) де МК виступає в якості обслуговуючого вузла, заявками на обслуговування є повідомлення, а процесом обслуговування заявки є процес обробки повідомлення (рис. 2).

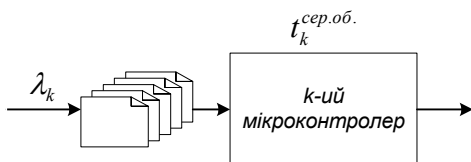


Рис. 2 – Модель СМО для оцінки продуктивності  $k$ -го мікроконтролера

Завданням є визначити продуктивність  $k$ -го МК  $P_{\Sigma k}$ , необхідну при максимальному потоці заявок на обслуговування. Оскільки, в більшості випадків, неможливо точно встановити максимальне значення потоку заявок, тому для спрощення оцінки використовують середню сумарну інтенсивність потоку заявок  $\lambda_k$ , що виконує  $k$ -й МК

$$\lambda_k = \sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \quad (9)$$

Тоді середній час обслуговування однієї заявки  $t_k^{сер.об.}$  для  $k$ -го МК визначається як

$$t_k^{сер.об.} = \frac{\sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \cdot l_{ij} \cdot \rho_j}{P_k \cdot \left( \sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \right)}, \quad (10)$$

де  $I_k$  – множина ФЗ, що вирішує  $k$ -й МК;  $\lambda_{ij}$  – інтенсивність вхідних повідомлень (повідомлення/сек.);  $l_{ij}$  – середня довжина вхідних повідомлень (біт);  $\rho_j$  – питома продуктивність МК (операцій/біт), при виконанні  $j$ -ї ФС;  $P_k$  – реальна продуктивність  $k$ -го МК.

Коефіцієнт завантаження  $k$ -го МК визначається наступним співвідношенням

$$\gamma_k = \frac{P_{\Sigma k}}{P_k} = \lambda_k \cdot t_k^{сер.об.}, \quad (11)$$

де  $P_{\Sigma k}$  – необхідна продуктивність та  $P_k$  – реальна продуктивність  $k$ -го МК (оп./с).

Враховавши  $\lambda_k$  та  $t_k^{сер.об.}$  (9, 10), необхідна продуктивність для  $k$ -го МК визначається наступним співвідношенням

$$P_{\Sigma k} = \sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \cdot l_{ij} \cdot \rho_j, \quad (12)$$

а на основі співвідношення (11) задається умова вибору типу  $k$ -го МК

$$P_k \geq \frac{P_{\Sigma k}}{\gamma_k}. \quad (13)$$

Враховуючи, що для систем реального часу, згідно [8, 15] рекомендоване значення  $\gamma_k$  становить 0,5-0,6, тип  $k$ -го МК вибирають виходячи із умови, що

$$P_k = (1,6 \div 2) \cdot P_{\Sigma k}. \quad (14)$$

### 3.2 ОЦІНКА НЕОБХІДНОГО ОБ'ЄМУ ПЗП

У ПЗП МК зберігається набір керуючих та прикладних програм. В процесі аналізу ФЗ та ФС, що реалізує  $k$ -й МК, визначається множина  $S_{\Sigma k}$  необхідного ПЗ

$$S_{\Sigma k} = S_{0k} \cup S_{J_k}, \quad (15)$$

де  $S_{0k}$  – множина керуючих програм;  $S_{J_k}$  – множина прикладного ПЗ, що реалізує  $k$ -й МК.

Зазвичай МК реалізує одну керуючу програму, а прикладне ПЗ складається із

множини підпрограм, що забезпечують реалізацію  $k$ -тим МК множини ФС  $J_k$ . Тоді необхідний об'єм ПЗП  $Q_k$  для  $k$ -го МК визначається за наступним співвідношенням

$$Q_k = Q_{0_k} + \sum_{j \in J_k} Q_{j_k}, \quad (16)$$

де  $Q_{0_k}$  – об'єм пам'яті необхідний для керуючої програми;  $\sum_{j \in J_k} Q_{j_k}$  – об'єм пам'яті необхідний для зберігання прикладного ПЗ.

Однак, пропонований в [13] підхід до реалізації МПП забезпечує можливість його дистанційного перепрограмування, тому в керуючій програмі головного МК немає необхідності, а виконуване прикладне ПЗ може мати мінімально необхідну, в даний момент часу, множину підпрограм. Таким чином, пропонований в [13] МПП має суттєву перевагу в необхідному об'ємі ПЗП відносно відомих рішень.

Якщо ПЗ зберігається у внутрішньому ПЗП, тип МК, визначається співвідношенням

$$Q_k \leq Q_{h \in H^{MC}}^{ROM}, \quad (17)$$

де  $Q_{h \in H^{MC}}^{ROM}$  – об'єм ПЗП МК  $h$ -го типу, що належить множині  $H^{MC}$ .

У випадку, якщо ПЗ зберігається у зовнішній пам'яті, необхідна кількість мікросхем  $N_k^{ROM}$  визначається співвідношенням

$$N_k^{ROM} = \left\lceil \frac{Q_k}{Q_{h \in \{H^{ROM} \cup H^{RAM}\}}^{ROM}} \right\rceil, \quad (18)$$

де  $Q_{h \in \{H^{ROM} \cup H^{RAM}\}}^{ROM}$  – об'єм пам'яті  $h$ -го типу, що належить множині, яка об'єднує підмножини ПЗП –  $H^{ROM}$  та ОЗП –  $H^{RAM}$ , що реалізують функцію ПЗП (див [13]);  $\lceil \alpha \rceil$  – відношення  $\alpha$ , заокруглене в більшу сторону до цілого числа.

### 3.3 ОЦІНКА НЕОБХІДНОГО ОБ'ЄМУ ОЗП

ОЗП використовується для тимчасового зберігання повідомлень, організації обмінників та стеків. Необхідний об'єм ОЗП  $V_k$   $k$ -го МК визначається співвідношенням

$$V_k = V_k^{data} + V_k^{stack}, \quad (19)$$

де  $V_k^{data}$  – об'єм ОЗП, необхідний для зберігання повідомлень і організації буферів обміну

даними;  $V_k^{stack}$  – об'єм ОЗП, необхідний для організації стеків програм.

При виконанні  $k$ -тим МК множини  $J_k$  ФС всі повідомлення обробляються відповідним ПЗ або перебувають в черзі на опрацювання. Всю множину програм, що реалізує  $k$ -ий МК, можна розділити на дві групи: 1) обробки даних; 2) вводу/виводу даних. Для програм обробки даних, час обробки повідомлень і час їх перебування в черзі залежать лише від продуктивності вибраного МК. В той же час, при виконанні програм виводу даних, час обробки повідомлень та час очікування в черзі, а отже довжина черги, залежать не лише від продуктивності МК, складності програми, а також від швидкості виводу даних відповідним пристроєм виводу (швидкодії інтерфейсу). Враховуючи те, що продуктивність типових 8-ми розрядних МК лежить в межах 2÷25 MIPS, а в деяких досягає 100 MIPS, можна констатувати, що для програм виводу час обробки повідомлень зазвичай займає незначний відсоток від часу виводу повідомлень. Таким чином, довжина черги повідомлень, що виводяться з МК, в основному буде залежати від швидкодії пристрою виводу (інтерфейсу).

Використавши математичний апарат теорії масового обслуговування [14], виведемо залежності для оцінки об'єму пам'яті, необхідної для зберігання повідомлень, що перебувають в черзі на обробку та обробляються  $k$ -им МК, а також об'єму пам'яті необхідної для зберігання повідомлень, що перебувають в черзі на вивід.

Представимо  $k$ -тий МК і набір його пристроїв вводу/виводу, як дворівневу систему масового обслуговування (рис.3), в якій підсистема першого рівня (ліва частина рис.3) виконує обробку повідомлень, а підсистема другого рівня (права частина рис.3) виконує виведення повідомлень.

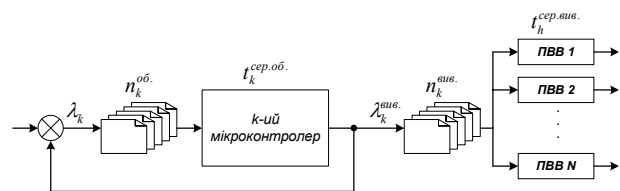


Рис. 3 – Модель СМО для оцінки об'єму ОЗП  $k$ -го мікроконтролера

В запропонованій моделі підсистема масового обслуговування першого рівня є одноканальною системою із необмеженою чергою на вході, де роль пристрою обслуговування заявок виконує  $k$ -тий МК. В ролі заявок виступають повідомлення, які поступають на обробку множиною програм  $S_{J_k}$ , що реалізують множину

$J_k$ . Вхідний потік заявок  $\lambda_k$  є сумою вхідних повідомлень, що визначається за (9). Обслуговуванням заявки є процес обробки вхідного повідомлення  $k$ -тим МК, а середній час обслуговування однієї заявки  $t_k^{сер.об.}$  визначається (10).

Підсистема другого рівня (див. рис. 3) є системою масового обслуговування, в якій кожен пристрій виводу (інтерфейс) можна розглядати як одноканальну систему із необмеженою чергою, яка працює на свого споживача. Вихідний потік  $\lambda_k^{вув.}$  – сума повідомлень, що поступають від МК на вивід, визначається співвідношенням

$$\lambda_k^{вув.} = \sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \cdot \xi_{ij} \cdot \xi_{jh}, \quad (20)$$

де  $\xi_{ij}$ , – двійкове число, що відображає зв'язок між  $i$ -ю ФЗ та  $j$ -ю ФС, а  $\xi_{jh}$  – двійкове число пов'язане із АЗ  $h$ -го типу, що належить множині  $H^{IF}$  і реалізує відповідний інтерфейс.

Означений потік повідомлень розділяється на множину потоків, число яких визначається кількістю пристроїв виводу (інтерфейсів)  $N_k^{IF}$ .

Об'єм пам'яті  $V_k^{data}$ , необхідний для зберігання даних, можна визначити за співвідношенням

$$V_k^{data} = l_k^{сер.об.} \cdot n_k^{об.} + l_k^{сер.вув.} \cdot n_k^{вув.}, \quad (21)$$

де  $l_k^{сер.об.}$ ,  $l_k^{сер.вув.}$  – середня довжина повідомлень, що обробляються  $k$ -м МК та поступають на вивід;  $n_k^{об.}$  – середня кількість повідомлень, що перебувають в черзі на обслуговування;  $n_k^{вув.}$  – середня кількість повідомлень, що перебувають в черзі на вивід.

Для підсистеми масового обслуговування, першого рівня, середня довжина повідомлень, що обробляються,  $l_k^{сер.об.}$  визначається співвідношенням

$$l_k^{сер.об.} = \frac{\sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij} \cdot l_{ij}}{\sum_{i \in I_k} \sum_{j \in J_i} \lambda_{ij}}. \quad (22)$$

Допускаючи безпріоритетний порядок обслуговування заявок, середнє число повідомлень, що перебувають в черзі на обслуговування  $k$ -тим МК, можна визначити за наступним співвідношенням

$$n_k^{об.} = \frac{\sum_{i \in I_k} \sum_{j \in J_i} t_{ijk} \cdot \gamma_k}{(1 - \gamma_k)}, \quad (23)$$

де  $\gamma_k$  – коефіцієнт завантаження  $k$ -го МК;  $t_{ijk}$  – середній час обробки повідомлень  $j$ -ї ФС при виконанні  $i$ -тої ФЗ  $k$ -тим МК, що визначається наступним співвідношенням

$$t_{ijk} = \frac{l_{ij} \cdot \rho_j}{P_k}, \quad (24)$$

де  $l_{ij}$  – середня довжина вхідних повідомлень;  $\rho_j$  – питома продуктивність МК при виконанні  $j$ -ї ФС;  $P_k$  – продуктивність  $k$ -го МК.

Для підсистеми масового обслуговування другого рівня середній час обслуговування заявок становить

$$t_k^{сер.вув.} = \frac{\sum_{i \in I_k} t_i^{сер.вув.} \cdot \lambda_i}{\sum_{i \in I_k} \lambda_i}, \quad (25)$$

де  $\lambda_i$  – інтенсивність вихідних повідомлень;  $t_i^{сер.вув.}$  – середній час виведення повідомлення по  $i$ -й ФЗ, що визначається співвідношенням

$$t_i^{сер.вув.} = \frac{l_i}{v_{h \in H^{IF}}}, \quad (26)$$

де  $l_i$  – середня довжина вхідних повідомлень по  $i$ -й ФЗ;  $v_h$  – пропускна швидкість інтерфейсу  $h$ -го типу, що належить множині  $H^{IF}$ .

Середня довжина повідомлень, що поступають на інтерфейс  $h$ -го типу при виконанні  $i$ -тої ФЗ визначається співвідношенням

$$l_i^{сер.вув.} = \sum_{j \in J_i} l_{ij} \cdot \xi_{jh}, \quad (27)$$

де  $\xi_{jh}$  – двійкове число, що показує зв'язок між  $j$ -тою ФС, із апаратним забезпеченням інтерфейсу  $h$ -го типу, що належить множині  $H^{IF}$ .

Об'єм пам'яті  $V_k^{stack}$ , необхідний для організації стеків, є пропорційним кількості програм  $S_{J_k}$  і обчислюється за співвідношенням

$$V_k^{stack} = \sum_{j \in J_i} v_j^{stack}, \quad (28)$$

де  $v_j^{stack}$  – об'єм пам'яті, необхідний для організації стека при виконанні ФС  $j$ -го виду.

Виходячи із попередніх результатів оцінки необхідного об'єму пам'яті, кількість мікросхем ОЗП  $N_k^{RAM}$  визначається співвідношенням

$$N_k^{RAM} = \left\lceil \frac{V_k^{data} - V_k^0}{Q_{h \in H^{RAM}}^{RAM}} \right\rceil, \quad (29)$$

де  $V_k^0$  – об'єм внутрішнього ОЗП МК;  $Q_{h \in H^{RAM}}^{RAM}$  – об'єм ОЗП  $h$ -го типу, що належить множині,  $- H^{RAM}$ ,  $\lceil \alpha \rceil$  – відношення  $\alpha$ , заокруглене в більшу сторону до цілого числа.

### 3.4 Оцінка часових обмежень

Одним із головних критеріїв оцінки ефективності пристроїв обробки даних є час рішення  $i$ -тої ФЗ, тобто проміжок часу, необхідний для перетворення вхідного повідомлення у вихідне, який можна представити як сумарний час виконання ФС, пов'язаних із даною ФЗ

$$t_i = \sum_{j \in J_i} t_{ij} \cdot \xi_{ij}. \quad (30)$$

Зазвичай на значення  $t_i$  накладається ряд обмежень, пов'язаних із характером ФЗ, які вирішує МПП, що формує вектор обмежень  $\langle t_i \rangle, i = \overline{1, I}$ .

При виконанні  $j$ -ої ФС використовується множина АЗ  $H_j$ , що дозволяє представити  $t_{ij}$  як сумарний час, затрачений АЗ на виконання необхідних функцій

$$t_{ij} = \sum_{h \in H_j} t_{ijh} \cdot \xi_{jh}, \quad (31)$$

де  $t_{ijh}$  – час виконання АЗ  $h$ -го типу  $j$ -ої ФС, яка відноситься до  $i$ -тої ФЗ;  $\xi_{jh}$  – двійкове число, яке показує зв'язок між АЗ  $h$ -го типу та  $j$ -ою ФС.

В свою чергу, процес виконання ФС АЗ розділяють на етапи обробки та очікування, які відповідно потребують часу на обробку повідомлення  $t_{ijh}^{ob}$ , та часу очікування в черзі на обробку  $t_{ijh}^{oc}$ . Тоді час виконання ФС в цілому визначається співвідношенням

$$t_{ijh} = t_{ijh}^{ob} + t_{ijh}^{oc}. \quad (32)$$

Вказані параметри  $t_{ijh}^{ob}$  і  $t_{ijh}^{oc}$  визначаються параметрами АЗ. Для МК час обробки повідомлення визначається співвідношенням (24), тобто  $t_{ijh}^{ob} = t_{ijk}^{ob}$ . Час очікування повідомлень на обробку  $t_{ijh}^{oc}$  визначається середнім часом очікування для  $k$ -го МК за співвідношенням

$$t_{ijh}^{oc} = t_k^{oc} = \frac{n_k^{ob}}{\lambda_k}, \quad (33)$$

де  $n_k^{ob}$  – середня кількість повідомлень що перебувають в черзі на обслуговування  $k$ -тим МК (23);  $\lambda_k$  – середня сумарна інтенсивність потоку заявок на обробку  $k$ -тим МК (7).

Для пристроїв виводу даних (інтерфейсів) час обробки повідомлення визначається співвідношенням (26), тобто  $t_{ijh \in H^{IF}}^{ob} = t_i^{сеп.вив.}$ .

## 4. СТРУКТУРНИЙ СИНТЕЗ МПП

В результаті проведено функціонального аналізу визначено перелік необхідних АЗ (5) і структура ПЗ (6) МПП.

Також сформовано множину вимог до: 1) продуктивності МК (13, 14); 2) об'єму ПЗП (16, 18) та ОЗП (19, 21, 28, 29); 3) вектора часових обмежень для (25, 26), (30-33), що формує вектор технічних обмежень для МПП.

Це дозволяє перейти до наступного етапу проектування МПП – структурного синтезу, що передбачає формування множини допустимих варіантів структури МПП  $\Omega_0$  та пошуку серед них множини оптимальних рішень  $\Omega_0$  по заданому критерію якості.

Враховуючи те, що, множина доступних комплектуючих є достатньо великою на п'ятому етапі (див. рис.1) виконується процедура попереднього вибору електронних компонентів (ЕК), параметри яких відповідають множині ТО. Для цього доцільно використати один із формальних методів багатокритеріального відбору. Будь який ЕК характеризується множиною технічних параметрів, які формують множину показників якості

$$K_{як.} = \{k_{як.} : k_{як.} = \overline{1, K_{як.}}\} \quad (34)$$

де  $K_{як.}$  – загальна кількість показників якості,  $k_{як.}$ -й показник якості.

Множину  $K_{як.}$  можна розділити на підмножини, що визначаються набором





варіант вузла, множина яких утворена після декомпозиції МПП, характеризується множиною показників якості (36), що формують вектор  $\vec{K}_{\text{сід.}} = \langle k_{\text{сід.}1}, k_{\text{сід.}2}, \dots, k_{\text{сід.}G} \rangle$ .

Основними критеріями оптимізації є: 1) собівартість вузла  $C_{nm}$  – визначається затратами на реалізацію  $m$ -го варіанту  $n$ -го вузла; 2) показник функціональної ефективності вузла  $E_{nm}$ . Для  $m$ -го варіанту  $n$ -го вузла значення показника  $E_{nm}$  визначається співвідношенням:

$$E_{nm} = \alpha_n \cdot (100 - E_m) \quad (40)$$

де  $\alpha_n$  – коефіцієнт, що відображає вагу вузла в МПП;  $E_m$  – відносна ефективність вузла у відсотках.

Враховуючи, що на попередніх етапах проведено відбір ЕК, технічні параметри яких відповідають множині обмежень заданих ТЗ та отриманих на етапі функціонального аналізу, то для оптимізації структури МПП достатньо двох критерії оцінки ефективності  $m$ -го варіанту  $n$ -го вузла ( $C_{nm}, E_{nm}$ ). В цілому для МПП можна записати

$$C_{\Sigma} = \sum_{n=1}^N \sum_{m=1}^M C_{nm} \cdot \xi_{nm}, \quad (41)$$

$$E_{\Sigma} = \sum_{n=1}^N \sum_{m=1}^M E_{nm} \cdot \xi_{nm}, \quad (42)$$

де  $\xi_{nm}$  – двійкове число, що дорівнює 1, при виборі  $m$ -го варіанту  $n$ -го вузла.

Умовами оптимізації є: 1) альтернативність рішень за сторонами; 2) перевага одних рішень над іншими.

Перша умова (альтернативність) передбачає, що всі  $M$  можливих варіантів реалізації вузла в межах  $n$ -го рядка ММ є альтернативним

$$\sum_{m=1}^M \xi_{nm}, \text{ для } n = 1 \dots N. \quad (43)$$

Друга умова (перевага одних рішень над іншими) представляється наступним чином: на скінченій множині  $P = P_1 \cup P_2 \dots \cup P_N$ , що задана ММ (39). Необхідно знайти вектор

$$\vec{X} = \langle j_1 \dots j_i \dots j_N \rangle \in \{X\}. \quad (44)$$

де  $\{X\}$  – скінченна множина, сила якого визначається множиною всіх можливих комбінацій індексів  $j_i$ .

Вектор  $\vec{X}$  визначає конфігурацію вибірки (одну із можливих комбінацій індексів  $j_i$ ), яка забезпечує мінімізацію цільової функції, за критеріями (41) і (42), що в загальному виражається наступним співвідношенням:

$$F_{\text{opt}} = f(C_{\Sigma}, E_{\Sigma}) \rightarrow \min. \quad (45)$$

Усі відомі методи двокритеріальної оптимізації зводять векторний синтез до скалярного [2, 6, 7]. Найбільш поширене [6] знаходження оптимального рішення за:

1) результирующим показником якості  $k_p$  як мінімумом добутку критеріїв якості:

$$k_p = C * E \rightarrow \min. \quad (46)$$

або зваженої їх суми (коефіцієнти  $c_1$  та  $c_2$  задаються експертом):

$$k_p = (c_1 * C + c_2 * E) \rightarrow \min. \quad (47)$$

2) показником ефективності (розглядається як функція собівартості та функціональної ефективності), при цьому пряма задача полягає в знаходженні системи з  $E_{\min}$  при  $C \leq C_{\max}$ , а обернена задача – у знаходженні  $C_{\min}$  при заданому обмеженні  $E \leq E_{\max}$ ;

3) мінімаксим методом, при якому критерії синтезованих варіантів нормуються:

$$C_j = \frac{C_j - C_{\min}}{C_{\max} - C_{\min}}, \quad E_j = \frac{E_j - E_{\min}}{E_{\max} - E_{\min}}. \quad (48)$$

де, для кожного  $j$  варіанту вибирається більше значення з нормованих критеріїв  $R_j = \max(C_j, E_j)$ , а далі вибирається система з найменшим значенням  $R = \min(R_j)$ .

Протиріччя “вартість-функціональні можливості” виключає рішення, краще за обома критеріями, тому доцільне поєднання знань, досвіду та інтуїції експерта-розробника з автоматизованим синтезом множини не гірших рішень (МНР). При цьому спочатку згідно безумовного критерію переваги (БКП) (принцип оптимальності за Парето) [2] знаходять множину не гірших рішень (МНР), а потім з отриманої МНР експерт вибирає кінцеве рішення з допомогою умовного критерію переваги (46-48) з врахуванням ТО і перспективи використання МПП. Згідно із БКП приймається, що, варіант  $\omega_i$  домінує над варіантом  $\omega_j$ , якщо для кожного з критеріїв якості  $k_l \in K_{\text{сід.}}$  виконується умова:

$$k_l(\omega_i) \leq k_l(\omega_j) \quad (49)$$

## 5. ВИСНОВКИ

На основі методу морфологічного аналізу та синтезу представлено метод структурного синтезу мережевих прикладних процесорів, що включає етапи функціонального аналізу, структурного синтезу, та пошуку множини оптимальних рішень. Запропонований метод поєднує лексикографічний критерій переваги (L-критерій) для відбору електронних компонентів на етапі функціонального аналізу та безумовний критерій переваги (оптимальності по Парето,  $\pi$ -критерій) на етапі пошуку множини оптимальних рішень, що розглядаються в літературі як альтернативні методи пошуку оптимальних рішень. Поєднання L та  $\pi$  – критеріїв дозволяє зменшити число синтезованих альтернативних варіантів на етапі структурного синтезу.

Отримане формалізоване рішення задачі дискретної оптимізації МПП, що є універсальним для широкого кола задач оптимального структурного синтезу цифрових пристроїв.

## ПОДЯКА

Дана робота виконана за підтримки фонду цивільних досліджень і розвитку США, при виконання міжнародного українсько-американського проекту #УКС2-005073-KV-07, "Dynamically reprogrammable network control application processor with Internet capability".

## 6. СПИСОК ЛІТЕРАТУРИ

- [1] Норенков И.П. *Основы автоматизированного проектирования: Учеб. для вузов. 2-е изд.* – М.: Изд-во МГТУ им. Н.С. Баумана, 2002. – 366 с.
- [2] Кандырин Ю.В. *Методы и модели многокритериального выбора вариантов в САПР: Учеб. Пособие для вузов.* – М.: Изд-во МЭИ, 2004. – 172 с.
- [3] Половинкин А.И. *Основы инженерного творчества: Учеб. пособие для студентов вузов.* – М.: Машиностроение, 1988. – 368 с.
- [4] Подиновский В.В., Ногин В.Д. *Парето-оптимальное решение многокритериальных задач.* – М.: Наука, 1982. – 256 с.
- [5] Сергиенко И.В., Капшицкая М.Ф. *Модели и методы решения на ЭВМ комбинаторных задач оптимизации.* – К.: Наукова думка, 1981. – 288 с.
- [6] Гуткин Л.С. *Оптимизация радиоэлектронных устройств по совокупности показателей качества.* – М.: Сов. радио, 1975. – 368 с.
- [7] Саченко А.О. *Теория дискретной оптимизации компонентов корректирующего канала ИИС для измерения температуры* // Н.-г. сборник "Метрологическое обеспечение производства и КИР", 1988, С. 95-101.
- [8] *Выбор микроЭВМ для информационных систем: Учеб. Пособие для вузов* / Н.М. Соломатин, Р.П. Шертвитис, М.М. Макшанцев. – М.: Высш. шк., 1987. – 120 с.
- [9] Кочан В.В., Тимчишин В.О. *Синтез оптимальных структур низковартисных комп'ютерных систем* // Вісник ДУ "Львівська політехніка". – 1998. – № 356. – С. 134 -144.
- [10] Одрин В.М. *Методы морфологического анализа технических систем.* – М.: ВНИИПИ, 1989. – 312 с.
- [11] *Морфологический синтез систем: морфологические методы поиска* / Одрин В.М. – Киев, 1986. – 40с. – (Перепринт / АН УССР. Ин-т кибернетики им. В.М. Глушкова; 86-5).
- [12] Палагін О.В., Романов В.О., Галелюка І.Б. *Віртуальні методи проектування складних систем: оцінка надійності* // Інформаційні технології та комп'ютерна інженерія. – 2005. – № 3. – С. 147–150.
- [13] I. Maykiv, A. Stepanenko, D. Wobshal, V. Kochan, R. Kochan, A. Sachenko *Remote Reprogrammable NCAPs: Issues and Approaches* // *Proceedings of 4th IEEE international workshop on Intelligent DataAcquisition and Advancing ComputingSystems*, 6-8 September, 2007, Dortmund, Germany, pp.109-113.
- [14] Клейнрок Л. *Теория массово обслуживания.* – М.: Машиностроение, 1979. – 408 с.
- [15] Phillip A. Laplante *Real-Time Systems Design and Analysis. Third edition/* – IEEE Press, Wiley-Interscience, 2004. – p. 506.
- [16] Кандырин Ю.В. *Автоматизированный многокритериальный выбор альтернатив в инженерном проектировании.* – М.: Изд-во МЭИ, 1992. – 52 с.
- [17] Одрин В.М., Картавов С.С. *Морфологический анализ систем. Построение морфологических таблиц.* – К.: Наукова думка, 1977. – 148 с.
- [18] Колдасов Г.Д. *Оптимизация решений при морфологическом синтезе технических систем.* / *Механизация и автоматизация управления*, 1987, № 3, С. 3 – 6.



**Ігор Майків** закінчив державний університет Львівська політехніка, по спеціальності "Радіотехніка" і отримав диплом радіоінженера (1996). Працював інженером-конструктором відділу спецтехніки на Тернопільському радіозаводі "Оріон" (1998-2002). Навчання в аспірантурі за спеціальністю 05.13.05 – Комп'ютерні системи та компоненти (2005-2009). В даний

час працює молодший науковий співробітник НДІ Інтелектуальних Комп'ютерних систем Тернопільського національного економічного університету (ТНЕУ).

Наукові інтереси: дослідження і створення цифрових систем на базі МК та програмованих логічних матриць.



**Андрій Степаненко** закінчив у 2005 році магістратуру ТНЕУ за спеціальністю Комп'ютерні системи та мережі. Продовжив своє навчання в аспірантурі за спеціальністю 05.13.06 "Інформаційні технології".

Наукові інтереси: методи і засоби автоматизованої верифікації програмного забезпечення інтелектуальних сенсорних мереж, розробка спеціалізованих мов програмування для вбудованих апаратних платформ, інженерія програмного забезпечення загалом



**Дарольд Вобшал**, президент компанії Esensors Inc; м. Амхерст, шт. Нью-Йорк, США. Отримав ступінь доктора наук по біофізиці від Університету шт. Нью-Йорк в Баффало. Має більше 50-ти років досвіду розробки і побудови сенсорів та сенсорних систем. Виступає

співзасновником та одним з головних експертів групи, яка розробляє стандарт IEEE 1451.



## A METHOD FOR STRUCTURAL SYNTHESIS OF NETWORK CAPABLE APPLICATION PROCESSORS

I. Maykiv <sup>1)</sup>, A. Stepanenko <sup>1)</sup>, D. Woschall <sup>2)</sup>

<sup>1)</sup> Research Institute of Intelligent Computer Systems, Ternopil National Economic University,  
46009 Ukraine, Ternopil, 11 Lvivska str.,

mim@tanet.edu.te.ua, {andrew.stepanenko@gmail.com}, <http://www.ics.tneu.edu.ua/>

<sup>2)</sup> Esensors Inc., 4240 Ridge Lea Road Suite 37 Amherst, NY 14226, USA 14226,  
darold@woschall.com, <http://www.eesensors.com/>

**Abstract:** A method for structural synthesis of Network Capable Application Processors (NCAPs) is proposed. It is based on a method of morphological analysis and synthesis and includes phases of functional analysis, structural synthesis, and search for a set of optimal solutions. The proposed method combines lexicographical criterion of preference (L- criterion) at a stage of functional analysis and unconditional criterion of preference (Pareto optimality) during the search phase, which are considered in literature as alternative methods of search for optimal solutions. Combining of criteria, makes it possible to reduce the number of synthesized alternative variants at a stage of structural synthesis and all obtained solutions are allowable.

**Keywords:** structural synthesis, morphological analysis, lexicographical criterion of preference, unconditional criterion of preference.

Modern measurement and control systems are implemented as multilevel systems with the distributed computing resources on the basis of industrial buses. The IEEE-1451 series of standards has been developed by IEEE for unification of requirements for hardware and software of such systems. Also a term *Network Capable Application Processor (NCAP)* as a data processing device is defined by this set of standards. NCAP occupies an intermediate level between a network server (high level) and sensors and actuators (low level). The main functions of the NCAP are data processing and support of a set of serial interfaces. However, each area of application is characterized by the set of functional tasks which demand use of the specialized device technical parameters, which are optimized for the specified area of use.

The analysis of available literature on this subject has shown, that at present the methods for optimum structural synthesis of digital systems based on microcontrollers which are characterized by a wide set of features with a significant price range is absent. The objective of this work is creation of a method for optimum structural synthesis of NCAP on the basis of microcontrollers, with optimal cost-functionality relation obtained from a set functional parameters which can appear as criterion for estimation of the device quality.

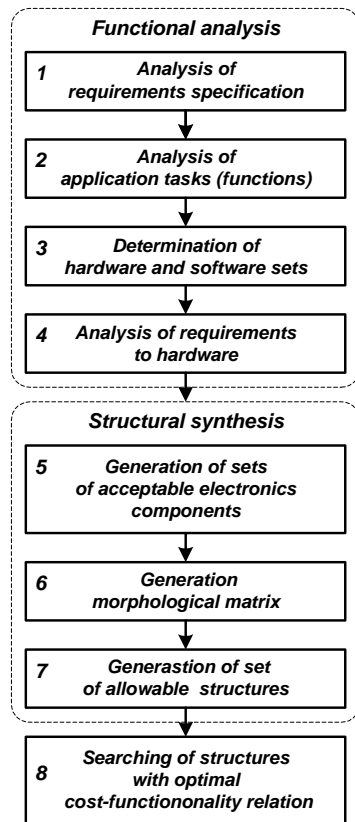
The effective technique of structural synthesis of

devices with optimal cost-functionality relation includes the next stages: 1) functional analysis; 2) structural synthesis; 3) search for optimal structures. The generalized algorithm of the proposed technique includes eight stages (Fig.1) which are divided into groups: functional analysis and structural synthesis.

The requirement specification for the NCAP is analyzed at the first stage of the design process. At this stage the list of functional tasks which are executed by the NCAP is created. Functional tasks are divided on system and application tasks. System tasks are realized by operational system (OS) or the specialized software and provide interaction for the application tasks. Application tasks serve external devices and users and are divided into two groups: 1) data exchange; 2) data processing.

The analysis of application tasks is performed at the second stage. The set of information parameters necessary to make estimation on the amount of input and output data for each application task is defined as a result of the analysis. As a result of this stage a set of vectors specifying functional restrictions for the NCAP is created.

The hardware and software necessary for implementation of all application functions of system is determined at the third stage as result of the requirement specification analysis and analysis of the list of application functions executed by the NCAP.



**Fig.1 – Algorithm of method of structural synthesis**

At the fourth stage the analysis of requirements for the device hardware is performed in particular: 1) processing power of a microcontroller; 2) memory size of ROM and RAM; 3) timing performances and temporal constraints. The estimation is made using elements of the theory of queues where the microcontroller is considered as the processing node, messages considered as service requests and message processing corresponds to service requests processing.

At the fifth stage the selection of electronic components is performed on the conducted functional analysis, where the list of the required hardware and software was created with timing characteristics and restrictions as well. The set of possible electronic components is created using lexicographic criterion of preference, which is a conditional criterion. A decision making person enters quality indicators priority list as additional information. As a result vector task of optimal selection of electronic components is transformed into a scalar task. This allows creation of a subsystem of a preliminary selection of electronic components which: 1) do not meet technical restrictions for the device being designed; 2) possess considerably better technical parameters. Such a preliminary selection enables to decrease the number of selection choices almost to the order of magnitude.

At the sixth stage a decomposition of the device

into a set of basic nodes is performed using morphological method. Then a morphological matrix is created ( $N \times M$ ), where  $N$  is the number of basic nodes, and  $M$  is the maximum number of alternate variants of the  $N$ -th node.

At the seventh and eighth stages the set of possible device structures is generated among which the search for optimal cost-functionality combination is performed. The process of this set creation is based on principles of discrete optimization, where each node variant is characterized by implementation costs  $C_{nm}$  and functional effectiveness  $E_{nm}$ . In general for the device the following is true

$$C_{\Sigma} = \sum_{n=1}^N \sum_{m=1}^M C_{nm} \cdot \xi_{nm}, \quad E_{\Sigma} = \sum_{n=1}^N \sum_{m=1}^M E_{nm} \cdot \xi_{nm}$$

In this case the criterion function is described by the following equation  $F_{opt} = f(C_{\Sigma}, E_{\Sigma}) \rightarrow \min$ .

The known methods of bi-criterion optimization transform vector synthesis into scalar synthesis. Widespread is the search for optimal solution by:

1) resulting quality indicator as a multiply of quality criteria  $k_p = C * E \rightarrow \min$  or weighted sum

$k_p = (c_1 \cdot C + c_2 \cdot E) \rightarrow \min$ ; 2) effectiveness

indicator (is considered as a function of self-cost and functional effectiveness). In this case the direct task lies in selection of a device with  $E_{\min}$  and set restriction  $C \leq C_{\max}$ , and the reverse task lies in determination of  $C_{\min}$  with the set restriction  $E \leq E_{\max}$ ; 3) mini-max method.

The cost-functionality contradiction excludes the best solution by two criteria. Therefore it is useful to join the knowledge and experience of a developer with the automated synthesis of a set of not worse solutions. In this case according to the Pareto's unconditional criterion the set of not worse solutions is determined. Then, the expert selects the final solution from within this set using one conditional criterion of preference and taking into consideration technical perspectives of the device application.

As a result we have received a formal solution of the discrete optimization task for design of digital structures based on microcontrollers. This solution is universal for a wide set of optimization tasks.

## ACKNOWLEDGMENT

The work presented in this paper is performed under the project #UKC2-005073-KV-07, "Dynamically reprogrammable network control application processor with Internet capability", supported by the U.S. Civilian Research & Development Foundation.