

## МЕТОДИ ВИСОКОПРОДУКТИВНИХ ПЕРЕТВОРЕНЬ ВЕЛИКОРОЗРЯДНИХ ЧИСЕЛ З БАЗИСУ РАДЕМАХЕРА У БАЗИС КРЕСТЕНСОНА

© Волинський О.І., 2012

**Представлено теоретичні основи розмежованої системи залишкових класів. Викладено інформаційну технологію реалізації методу міжбазисних перетворень (Радемахера–Крестенсона) на основі розмежованої системи числення залишкових класів. Розроблене швидкодіюче зворотне перетворення Крестенсона–Радемахера.**

**Ключові слова:** міжбазисні перетворення, залишкові класи, базис Крестенсона, базис Радемахера.

**The theoretical basis of delimited Residual Number System. Described information technology implementation method between basis transformations (Rademacher-Krestenson) based on delimited notation residual classes. Developed a Krestenson–Rademacher’s high-speed reverse transformation.**

**Key words:** basic transformations, residual classes, Rademacher basis, -Krestenson basis.

### Вступ

З розвитком сучасних технологій актуальною проблемою є збільшення швидкодії опрацювання інформації. Система числення залишкових класів (СЗК), яка породжується теоретико-числовим базисом Крестенсона, характеризується суттєвими перевагами відносно базису Радемахера під час виконання операцій додавання та множення, а наявність переважної більшості обчислювальної техніки в базисі Радемахера породжує ряд задач, для розв'язання яких потрібно використовувати переваги обох цих базисів. Однією з таких задач є створення методів високопродуктивних перетворень великорозрядних чисел між базисами Радемахера та Крестенсона.

### Аналіз публікацій і окреслення наукової задачі

Аналіз літературних джерел щодо використання базисів Радемахера та Крестенсона для перетворення, передавання та опрацювання інформаційних потоків у сучасних комп'ютерних системах показує, що:

- у базисі Радемахера [1] реалізована виключна більшість універсальних процесорів комп'ютерної техніки, а також сигнальних процесорів різних застосувань [2];
- базис Крестенсона, який породжує систему числення залишкових класів, успішно застосовується для побудови спецпроцесорів стиснення інформації [3] та реалізації високопродуктивних процесорів опрацювання інформаційних потоків [4], у системах протиповітряної оборони та для опрацювання великорозрядних чисел у системах криптозахисту інформації.

Запропонований в [5] алгоритм перетворення чисел базису Радемахера в СЗК на основі теорії розмежованої СЗК (РСЗК) дає змогу поглибити процес розпаралелювання та спрощення арифметичних операцій базису Крестенсона. Реалізація побітового розмежування чисел у базисі Радемахера істотно спростить алгоритм переходу з базису Радемахера в базис Крестенсона. Тому перспективним напрямом підвищення швидкодії обчислювальних процесів є застосування базису Крестенсона, зокрема розмежованої СЗК.

## Мета роботи

Мета роботи – розроблення теоретичних засад та реалізація методів міжбазисного перетворення для базису Радемахера та Крестенсона, а також оцінка особливостей арифметики цих базисів та їх порівняння.

### 1. Представлення чисел у розмежованій СЗК

Розмежовану систему залишкових класів, основою якої є цілочислова форма СЗК, можна подати у вигляді суми фрагментів числа:

$$N_k = N_{1k} + N_{2k} + \dots + N_{ik} + \dots + N_{nk},$$

де  $N_{ik}$  –  $m$ -розрядний компонент числа  $N_k$ , яке представлено у двійковій системі числення, числового базису Радемахера. Наприклад, 64-розрядний процесор СЗК може бути розмежований на чотири фрагменти по 16 бітів (рис.1).

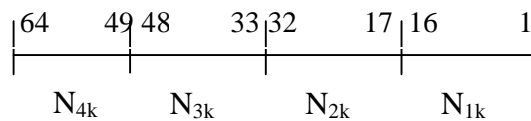


Рис. 1. Процес розмежування 64-розрядного процесора

Тоді пряме перетворення РСЗК набуде вигляду:

$$\begin{array}{l}
 \begin{array}{c} \dots \\ N_k = \\ \dots \end{array} \begin{array}{l} \nearrow \\ \longrightarrow \\ \searrow \end{array} \begin{array}{l} b_1 = (b_{11} + b_{21} + \dots + b_{r1} + \dots + b_{n1}) \bmod p_1 \\ \dots \dots \\ b_i = (b_{i1} + b_{2i} + \dots + b_{ri} + \dots + b_{ni}) \bmod p_i \\ \dots \dots \\ b_k = (b_{1k} + b_{2k} + \dots + b_{rk} + \dots + b_{nk}) \bmod p_k \end{array} \quad (1)
 \end{array}$$

Отже, математичні операції в РСЗК виконуються паралельно по кожному із фрагментів процесора, що забезпечує ще глибший рівень розпаралелювання опрацювання інформації, а відповідно підвищує швидкодію процесора СЗК [6].

Зі структури міжбазисного перетворення (1) видно, що вона потребує обчислення залишків для кожного компонента згідно з виразом

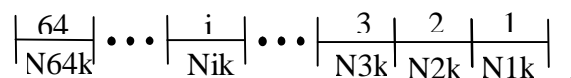
$$b_{ij} = \text{res} N_{ij} (\bmod p_i),$$

де  $\text{res}$  – символ операції отримання залишку.

При цьому процедура обчислення загального залишку виконується відповідно

$$b_i = \text{res}(b_{i1} + b_{i2} + \dots + b_{in}) \bmod p_i.$$

Наступним прикладом РСЗК є бінарне розмежування двійкових чисел базису Радемахера, тобто  $k=1$ . Структура розмежування має такий вигляд:



Розмежувавши двійкове число  $(X_{n-1}, X_{n-2}, \dots, X_i, \dots, X_1, X_0)$ , відповідно отримаємо матрицю залишків кожного  $i$ -го розряду у системі взаємопростих модулів  $P_1, P_2, \dots, P_j, \dots, P_k$  (табл. 1) [6].

Для отримання кінцевого залишку в базисі Крестенсона над елементами рядків матриці, поданої в табл. 1, виконується операція (2), в результаті виконання якої одержуємо кінцеві залишки по відповідних модулях  $P_j$ :

$$\text{res}(b_{n-1,j} + b_{n-2,j} + \dots + b_{i,j} + \dots + b_{1,j} + b_{0,j}) \bmod P_j; j \in \overline{1, k}. \quad (2)$$

Таблиця 1

## Матриця залишків числа X

	$X_{n-1}$	$X_{n-2}$	...	$X_i$	...	$X_1$	$X_0$
$P_1$	$b_{n-1,1}$	$b_{n-2,1}$	...	$b_{i,1}$	...	$b_{1,1}$	$b_{0,1}$
$P_2$	$b_{n-1,2}$	$b_{n-2,2}$	...	$b_{i,2}$	...	$b_{1,2}$	$b_{0,2}$
$P_3$	$b_{n-1,3}$	$b_{n-2,3}$	...	$b_{i,3}$	...	$b_{1,3}$	$b_{0,3}$
$P_4$	$b_{n-1,4}$	$b_{n-2,4}$	...	$b_{i,4}$	...	$b_{1,4}$	$b_{0,4}$
...	...	...	...	...	...	...	...
$P_j$	$b_{n-1,j}$	$b_{n-2,j}$	...	$b_{i,j}$	...	$b_{1,j}$	$b_{0,j}$
...	...	...	...	...	...	...	...
$P_k$	$b_{n-1,k}$	$b_{n-2,k}$	...	$b_{i,k}$	...	$b_{1,k}$	$b_{0,k}$

## 2. Методи міжбазисних перетворень Радемахера–Крестенсона

2.1. Один із методів перетворення базису Радемахера в базис Крестенсона полягає в отриманні залишку  $b_i$  з двійкового числа  $X_{(2)}$  (рис.2) [7], починаючи зі старшого розряду  $X_{(2)} = (a_0, a_1, \dots, a_i, \dots, a_{n-1})$ , за заданим модулем  $P_j$ , що описується рекурентною формулою

$$b_i = (a_i + 2 \cdot b_{i-1}) \bmod P_j,$$

де  $a_i \in \overline{0,1}$  – значення  $i$ -го біта двійкового числа;  $b_{i-1}$  – значення залишку  $i-1$ -го біта двійкового числа. Початкова умова рекурентної формули отримання залишку задається такими даними:  $i = n-1$ ,  $b_{i-1} = 0$ . Отримане  $b_0$  – шуканий залишок згідно з виразом

$$b_{0,j} = \text{res}X \pmod{P_j}. \quad (3)$$

На рис. 2 зображена функціональна структура  $j$ -го блока процесора міжбазисного перетворення Радемахера–Крестенсона.

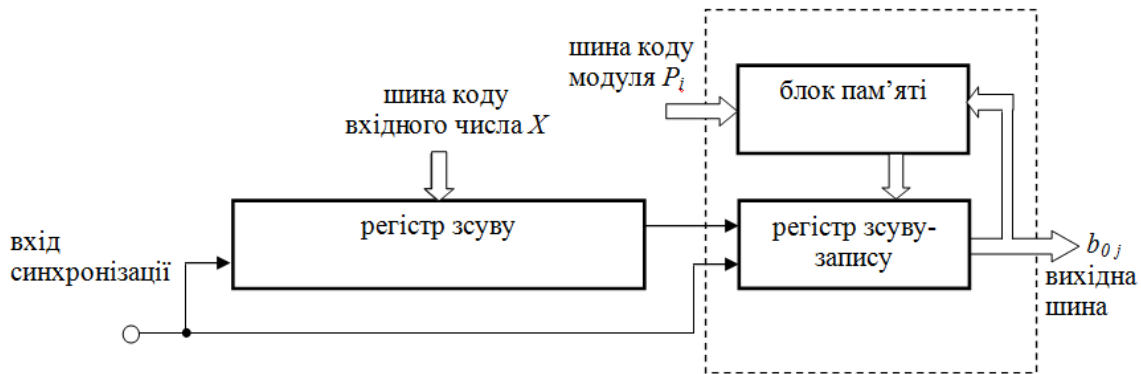


Рис. 2. Структура блока процесора міжбазисного перетворювача Радемахера–Крестенсона

Для отримання коду залишку  $b_{0,j}$  вхідне двійкове число X записують у регістр зсуву у вигляді впорядкованому старшими розрядами до його виходу, регістр зсуву-запису скидається в нульовий стан, на шину коду модуля подається значення модуля  $P_j$ , який надходить на перший адресний вхід блока пам'яті. На вхід синхронізації надходять імпульси тактової частоти, які послідовно зсувають інформацію у регістрі зсуву та регістрі зсуву-запису. В кожному такті вхідний код залишку  $b_{i-1,j}$  надходить на другий адресний вхід блока пам'яті, з виходу якого записується код залишку  $b_{i,j}$  у регістр зсуву-запису.

Реалізація запропонованого методу виконується зчитуванням кодів залишків  $b_j$ , у випадку  $b_{i-1,j}, i \leq P_j - 1$ . Причому, якщо адресний код  $b_{i-1,j} \geq P_j$ , то з пам'яті вибирається код згідно з виразом (3). Після  $n$  тактів синхронізації на виході регістру зсуву-запису отримується код залишку  $b_{0j}$  за модулем  $P_j$ .

Розглянемо приклад: нехай потрібно обчислити залишок числа  $X=100_{(10)}=1100100_{(2)}$  за модулем  $P=11_{(10)}=1011_{(2)}$ .

Алгоритм обчислення залишку  $b_{0j}$  виконується відповідно до послідовності операторів присвоєння ( $:=$ ) (табл. 2).

Таблиця 2

Вхідні та вихідні значення залишків для блока пам'яті

	$b_{i-1,j} :=$	$b_{i,j} :=$
1	00000	00000
2	00011	00011
3	00110	00110
4	01100	00001
5	00011	00011
6	00110	00110
7	01100	00001

Після  $n=7$  зсувів отримаємо остаточне значення залишку  $b_i=00001$ , що зчитується вихідною шиною.

Операція обчислення залишку розглянутим способом потребує  $n$  тактів у регістрі зсуву та регістрі зсуву-запису та  $n$  тактів вибірки коду залишку з блока пам'яті, тобто дорівнює  $2n$  тактів.

2.2. Наступним швидкодіючим методом перетворення чисел з позиційної системи в систему залишкових класів є метод на основі рандомізаторів. Структурна схема такого міжбазисного перетворювача, зображена на рис. 3 [8], складається з: 1 – вхідних шин  $K$  –розрядного позиційного числа; 2 – комутаційних мультиплексорів, 3 – виходів коду  $b_i$  системи залишкових класів.

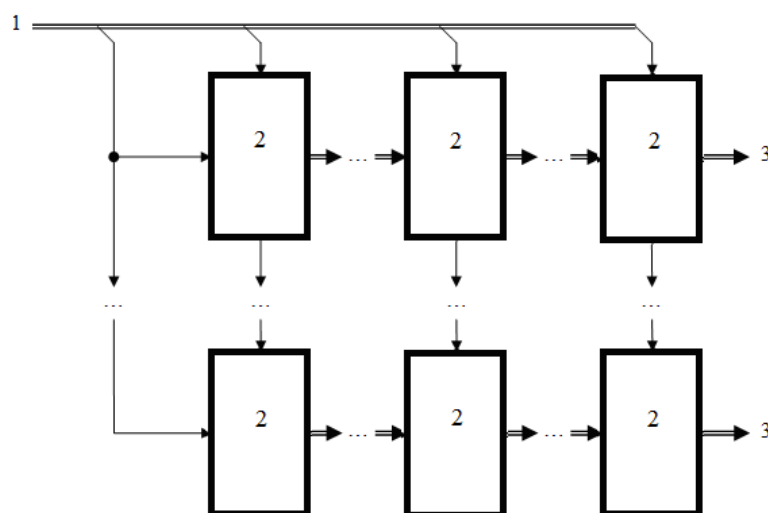


Рис. 3. Структурна схема міжбазисного перетворювача Радемахера–Крестенсона

На рис. 4 зображено структурну схему компонента міжбазисного перетворювача Радемахера–Крестенсона, який містить комутаційні мультиплексори – 1, до складу яких входять: 2.1 – рандомізатор за модулем  $P_j$  у базисі Хаара, 2.2 – інкрементний пристрій за модулем  $P_j$ , 2.3 –  $P$ -канальний двовходовий мультиплексор.

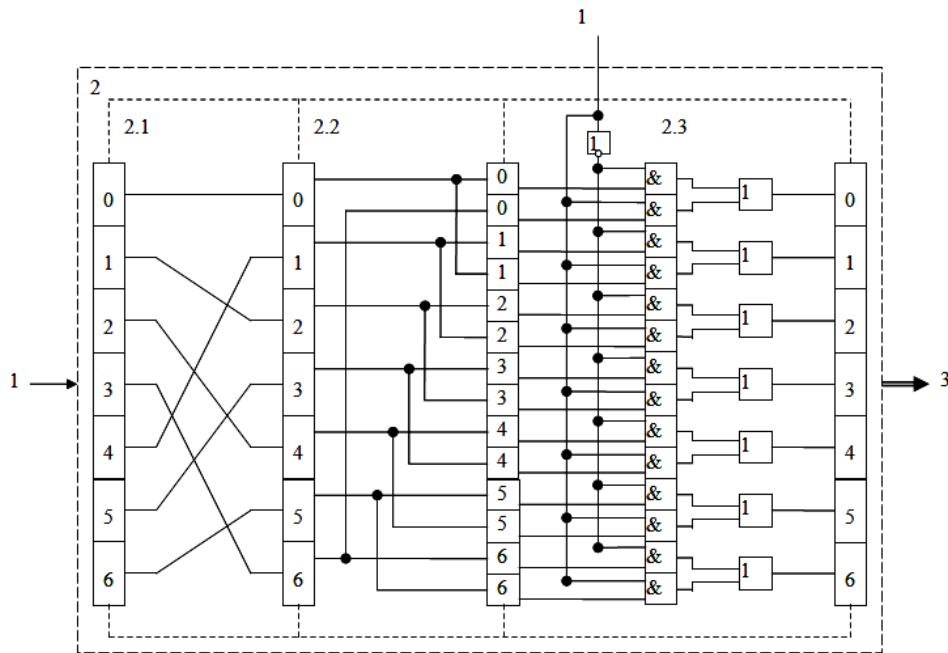


Рис. 4. Структурна схема комутаційного мультиплексора міжбазисного перетворювача Радемахера–Крестенсона ( $P=7$ )

Розглянемо для прикладу обчислення залишку двійкового числа  $X=100_{(10)}=1100100_{(2)}$  за модулем  $P=7_{(10)}=111_{(2)}$ :

– на вхідну шину подається двійкове значення числа  $X=1100100$ , старший біт якого приймаємо за перший залишок  $X_6=b_7=1$ . Цей біт надходить на вхід рандомізатора за модулем  $P=7$ , в результаті на виході отримують залишок  $b_{1,Rand}=2$ . Отримане  $b_{1,Rand}$  надходить на інкрементний пристрій, на виході якого отримують два значення  $b_{1,Incr}=2;3$ , які надходять на відповідні входи  $P$ -канального двохвального мультиплексора. На другий вхід  $P$ -канального двохвального мультиплексора попадає з вхідної шини наступний біт  $X_5=1$ , який встановлює його у відповідний режим роботи. В результаті на виході мультиплексора формується залишок  $b_{1,MX}=b_2=3$ .

Аналогічно паралельно виконуються операції для інших розрядів двійкового числа, де для вхідних даних  $b_i$  та  $X_i$  формуються вихідні дані  $b_{i,Rand}$ ,  $b_{i,Incr}$  та  $b_{i,MX}=b_{i+1}$  (рис.5):

- вхідні дані:  $b_2=3$ ,  $X_4=0$ ; вихідні дані:  $b_{2,Rand}=6$ ,  $b_{2,Incr}=6;0$  та  $b_{2,MX}=b_3=6$ ;
- вхідні дані:  $b_3=6$ ,  $X_3=0$ ; вихідні дані:  $b_{3,Rand}=5$ ,  $b_{3,Incr}=5;6$  та  $b_{3,MX}=b_4=5$ ;
- вхідні дані:  $b_4=5$ ,  $X_2=1$ ; вихідні дані:  $b_{4,Rand}=3$ ,  $b_{4,Incr}=3;4$  та  $b_{4,MX}=b_5=4$ ;
- вхідні дані:  $b_5=4$ ,  $X_1=0$ ; вихідні дані:  $b_{5,Rand}=1$ ,  $b_{5,Incr}=1;2$  та  $b_{5,MX}=b_6=1$ ;
- вхідні дані:  $b_6=1$ ,  $X_0=0$ ; вихідні дані:  $b_{6,Rand}=2$ ,  $b_{6,Incr}=2;1$  та  $b_{6,MX}=b_7=2$ .

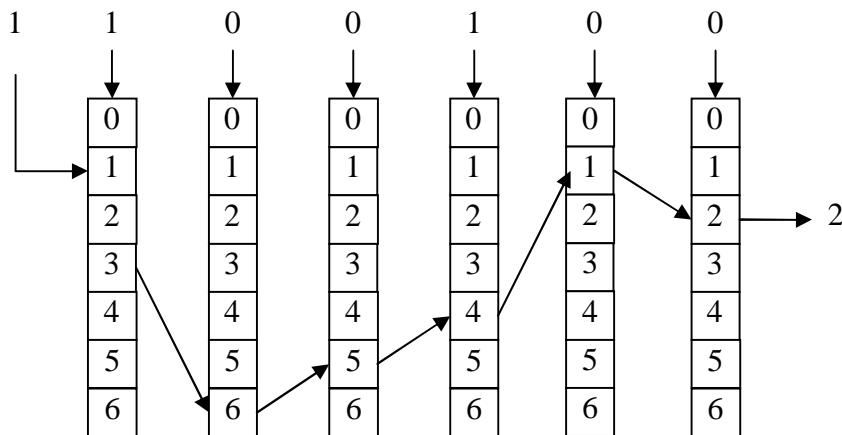


Рис. 5. Граф обчислення залишку за модулем  $P=7$  на основі комутаційних мультиплексорів

Кінцевим результатом обчислення залишку числа  $X=100$  за модулем  $P_j=7$  буде  $b_j=b_7=2$ .

Отже, незалежно від розрядності великорозрядного числа операція обчислення двійкового числа відбувається за два такти вентильної мікроелектроніки.

2.3. Для знаходження залишку великорозрядного числа в основу методу покладено алгоритм обчислення залишку багаторозрядного двійкового числа  $Y$  за багаторозрядним цілочисловим модулем  $P$  згідно з рекурсивним виразом [9]:

$$b_i = \begin{cases} b_{i-1}, b_i \leq P_{j-1} \\ [P_j] + b_{i-1}, b_i > P_{j-1} \end{cases},$$

де  $n$  – розрядність числа  $Y$ , з якого визначається залишок  $b_i$ ;  $a_i$  – біти двійкового числа  $Y$ , починаючи зі старшого розряду  $a_n$ ;  $[P]_{\text{мд}}$  –  $k+1$  розрядна мантиса доповнювального коду модуля  $P$ ,  $b_i$  – поточне кодове значення залишку ( $b_{i-1} = 0$ ).

На рис. 6 показано структурну схему процесора обчислення залишку великорозрядного двійкового числа, де 1 – вхідний  $n$ -розрядний регістр зсуву, 2 – вхідна шина запису двійкового представлення числа  $Y$ , 3 – шина запису кодового представлення модуля  $[P]_{\text{мд}}$ , 4 – блок управління, 5 –  $k+1$ -розрядний регістр запису та зсуву доповнювального коду, 6,7 –  $k+1$ -розрядні регістри зсуву для почергового зчитування та запису проміжних залишків, 8 – однорозрядний накопичувальний суматор, 9 – мультиплексор, 10 – вихідна шина кодового представлення залишку  $b_i$ .

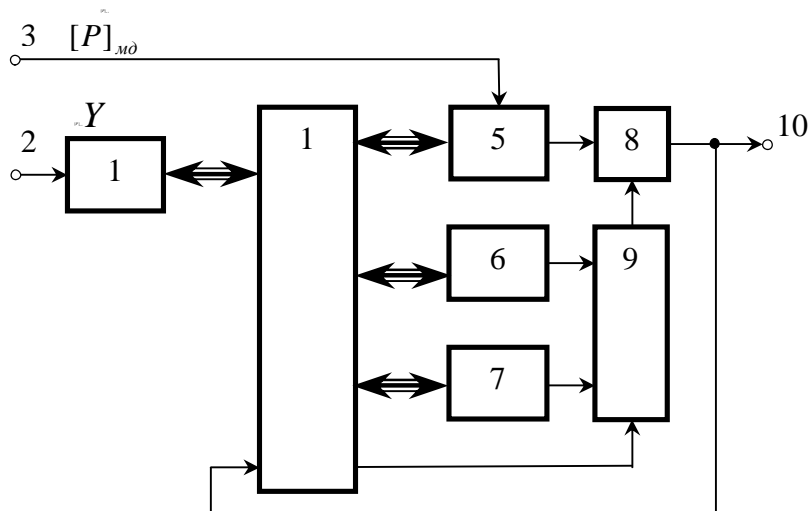


Рис. 6. Структурна схема процесора обчислення залишку великорозрядного двійкового числа

Розглянемо приклад визначення залишку за непарним модулем:

Нехай  $Y=100_{(10)}=1100100_{(2)}$ ,  $P=11_{(10)}=1011$ .

Тоді  $b_1 = \text{res}100(\text{mod}11) = 1$ .

Потрібно визначити  $b_1$  над двійковими кодами  $Y$  та  $P$ :

1. У перший регістр 1 записуємо число  $Y=1100100_{(2)}$ .

2. У другий регістр 5 записуємо  $k+1$  розрядну мантису доповнювального коду  $[P]_{\text{мд}} = 10101$ , яку отримуємо так:

- код числа  $P$  записуємо з нульовим бітом у старшому розряді:  $P=01011_{(2)}$
- інвертуємо цей код:  $\square P=10100$
- додаємо до цього коду “1”:

$$[P]_{\text{мд}} = \begin{array}{r} 10100 \\ +1 \\ \hline 10101 \end{array}.$$

3. У третій регістр 6 і четвертій 7 регістри записуємо нулі, тобто  $b_i = 00000$   $b_{i-1} = 00000$ ;  $i = n, n-1, \dots, 1$ .

4. Блок управління 4 формує біт "0" на керуючий вхід мультиплексора 9, що дає змогу записувати та зчитувати відповідну інформацію  $[P]_{m0}$ ,  $b_i$ ,  $b_{i-1}$  з відповідних регістрів 5, 6, 7, з другого регістра 5 і третього регістра 6 ( $b_i$ ) та четвертого регістра 7 ( $b_{i-1}$ ) та записувати коди порозрядно вихідні коди суматора у четвертий регістр 7.

5. У кожному циклі роботи процесора виконується підсумування мантиси доповнювального коду модуля  $P$  зі значенням поточного залишку  $b_{i-m}$ . Для розглянутого прикладу потрібна така операція

$$\begin{array}{r} [p]_{m0} \quad 10101 \\ 2b_i + a_i \quad + 00001 \quad i = n-1. \\ \hline [b_{i-m+1}]_{m0} \quad \langle 1 \rangle 1110 \end{array}$$

6. Отримуємо мантису доповнювального коду цієї операції додавання. Код цієї мантиси записуємо у регістр 6 ( $b_i$ ) або 7 ( $b_{i-1}$ ). Отримане значення "1" в  $k+1$  такті підсумування показує, що  $b_{i-1} < P$ . Тоді відбувається зсув інформації у відповідному регістрі 6 або 7 і запис у молодший розряд  $a_{i-1}$  біта числа  $Y$ , тобто

$$\begin{array}{r} 10101 \qquad \qquad \qquad 10101 \qquad \qquad \qquad 10101 \\ P > b_i \quad + 00011 \quad i=n-2, P > b_i \quad + 00110 \quad i=n-3, P \leq b_i \quad + 01100 \quad i=n-4. \\ \hline \langle 1 \rangle 1100 \qquad \qquad \qquad \langle 1 \rangle 1011 \qquad \qquad \qquad \langle 0 \rangle 0011 \end{array}$$

7. Отриманий біт "0" надходить у блок управління, який запам'ятовує його і перемикає мультиплексор 9, що приводить до подвоєння коду залишку в регістрі 7 ( $b_{i-1}$ ) та запису в молодший розряд поточного біта числа  $Y$ .

8. Після цього використовується операція підсумування згідно з процедурою:

$$\begin{array}{r} 10101 \quad 10101 \quad 10101 \\ + 00011 \rightarrow + 00110 \rightarrow + 01100 \quad . \\ \hline \langle 1 \rangle 1000 \quad \langle 1 \rangle 1011 \quad \langle 0 \rangle 0001 = b_1 \end{array}$$

Тобто  $b_1 = 00001_{(2)} = 1_{(10)}$ , що відповідає  $b_1 = \text{res}100(\text{mod}11) = 1$ .

Аналогічні розрахунки можна виконати для визначення залишку за парним модулем:

Нехай  $Y=25_{(10)}=11001_{(2)}$ ,  $p=6_{(10)}=110$ . Запишемо  $[P]_{m0} = 0110 \xrightarrow{-p} 1001 \xrightarrow{+1} 1010$ .

Виконаємо такі операції:

$$\begin{array}{r} 1010 \quad 1010 \quad 1010 \\ + 0001 \rightarrow + 0011 \rightarrow + 0110 \\ \hline 1011 \quad 1101 \quad 0000 \end{array}$$

Оскільки в старшому розряді "0", то цей залишок зсувається на біт і додається новий біт числа:

$$\begin{array}{r} 1010 \quad 1010 \\ + 0000 \rightarrow + 0001 = b_1 \quad . \\ \hline 1010 \quad 1011 \end{array}$$

Оскільки використано всі біти числа  $Y$  і  $b_i < P$ , то  $b_1 = 0001_{(2)}$ , що відповідає  $b_1 = \text{res}25(\text{mod}6) = 1$ .

### 3. Швидкодіюче зворотне перетворення перетворення Крестенсона–Радемахера

Високопродуктивне перетворення Крестенсона–Радемахера доцільно виконувати на основі досконалої форми СЗК. Математичною основою такої СЗК є виконання міжбазисного перетворення Крестенсона–Радемахера згідно з виразом:

$$[N_k]_0 = \text{res} \sum_{i=1}^k [b_i]_0 \pmod{1} ,$$

де  $[N_k]_0$ ,  $[b_i]_0$  – відповідно нормалізовані представлення коду числа та його залишків у системі взаємопростих модулів  $p_i$ .

Відомі такі набори модулів, які утворюють досконалу форму СЗК [10]: (2, 3, 5); (2, 3, 7, 41); (2, 3, 11, 59) та інші, що використовуються для побудови високопродуктивного перетворювача чисел з базису Крестенсона в базис Радемахера. Наприклад, у системі взаємопростих модулів (2, 3, 5) з набором базисних чисел:  $B_1 = 15$ ;  $B_2 = 10$ ;  $B_3 = 6$  і значенням числа  $N_k = 1$  маємо:

$$N_k = (1 \cdot 15 + 1 \cdot 10 + 1 \cdot 6) \pmod{30} = 31 \pmod{30} = 1 .$$

Представимо базисні числа у вигляді 5-розрядних двійкових кодів:

$$15_{(10)} = \begin{matrix} 00000_{(2)} \\ \underline{01111}_{(2)} \end{matrix} ; 10_{(10)} = \begin{matrix} 00000_{(2)} \\ \underline{01010}_{(2)} \\ 10100_{(2)} \end{matrix} ; 6_{(10)} = \begin{matrix} 00000_{(2)} \\ \underline{00110}_{(2)} \\ 10010_{(2)} \\ 11000_{(2)} \end{matrix} .$$

Отже, у цій СЗК для модуля 2 код Хаара матиме 15 розрядів, для модуля 3 – відповідно 20 розрядів і для модуля 5 – відповідно 24 розряди (рис.7).

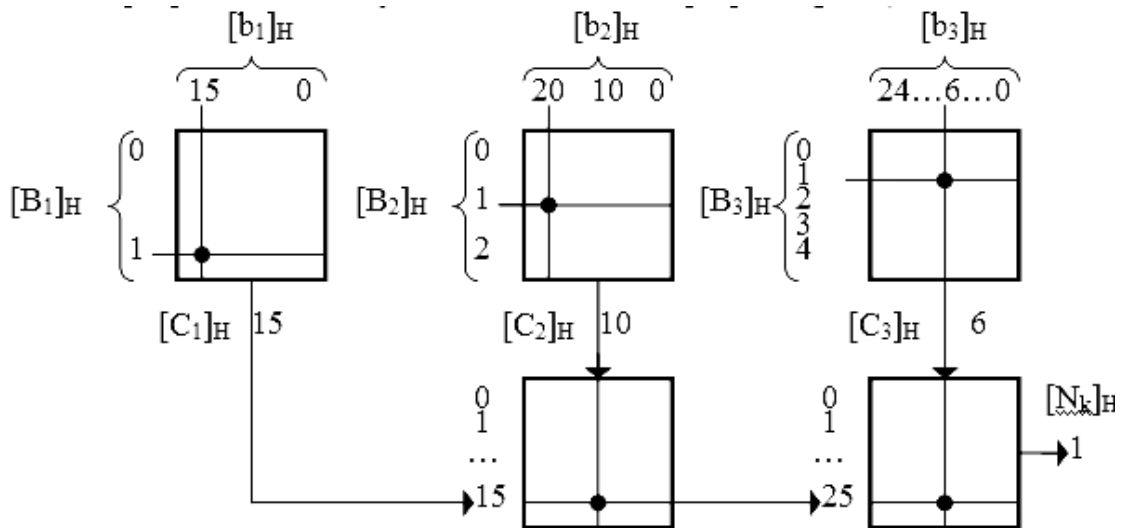


Рис. 7. Структура міжбазисного перетворення Крестенсона–Радемахера у системі взаємопростих модулів (2, 3, 5)

У загальному випадку, коли зростає величина модулів та базисних чисел досконалої СЗК, доцільно використовувати постійно запам'ятовуючі пристрої (ПЗП), наприклад, кристал флеш-пам'яті, у яких операції перетворення двійкових кодів залишків  $b_i$  у базис Хаара можна запрограмувати відповідно. Структуру такого міжбазисного перетворювача показано на рис. 8.



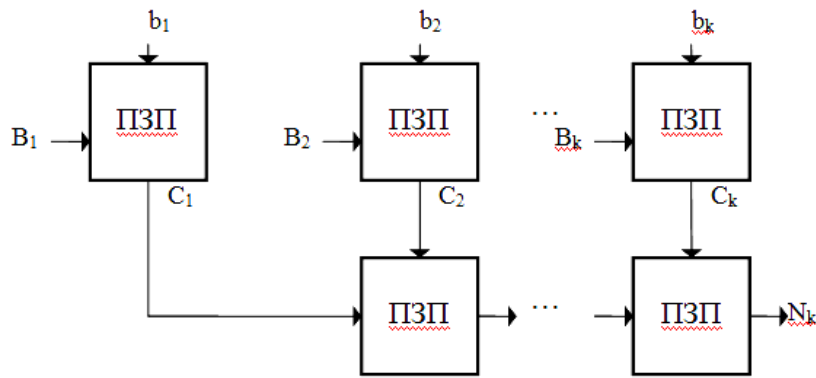


Рис. 8. Структура міжбазисного перетворення Крестенсона-Радемахера для досконалої форми СЗК

Використовуючи цілочислову СЗК, міжбазисне перетворення Крестенсона-Радемахера виконують згідно з виразом:

$$N_k = \text{res} \sum_{i=1}^k b_i \cdot B_i \cdot m_i \pmod{P}, \quad (4)$$

де  $0 \leq m_i \leq p_i - 1$ .

Структура такого спецпроцесора, який реалізує перетворення (4), показана на рис. 9.

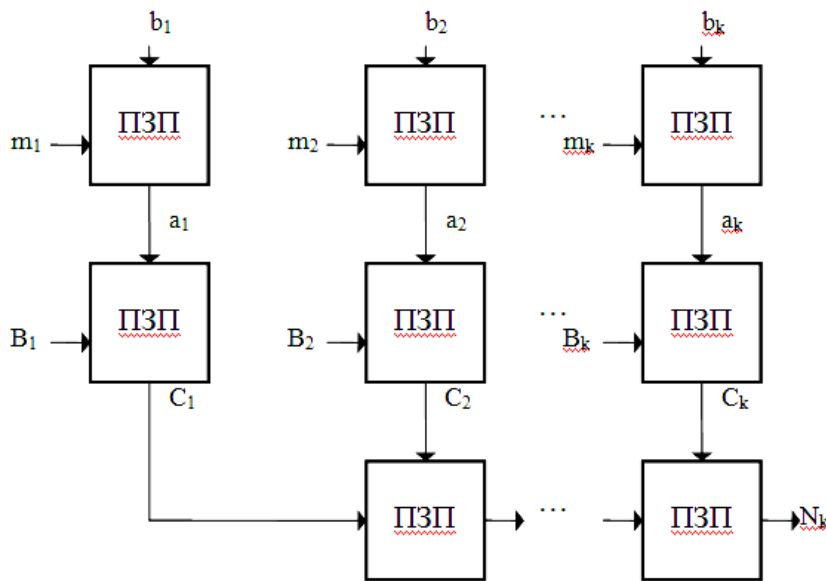


Рис. 9. Структура міжбазисного перетворення Крестенсона-Радемахера для цілочислової форми СЗК

Розрахунок ПЗП на базі кристалів сучасної флеш-пам'яті виконується, з умови адресної розрядності ( $A_{ПЗП}$ ) використаних кристалів, так:  $A_{ПЗП} \geq \log_2(p_{i-1}) + \log_2(P-1)$ . При цьому швидкодія процесора міжбазисного перетворення відповідатиме  $k+1$ - тактам доступу та вибірки інформації з ПЗП.

### Висновки

Розроблені методи міжбазисних перетворень великорозрядних чисел з теоретико-числового базису Радемахера в базис Крестенсона, а також запропоновані структури процесорів, які їх реалізують. Розроблене швидкодіюче зворотне перетворення Крестенсона-Радемахера, що

забезпечує велику швидкодію міжбазисного перетворення за рахунок використання постійно запам'ятовуючого пристрою. Ці дослідження дали змогу спроектувати пристрої міжбазисних перетворювачів, що, своєю чергою, дає можливість інтегрувати спецпроцесори базису Крестенсона в сучасну обчислювальну техніку.

1. Мельник А.О. *Архітектура комп'ютера* / А.О. Мельник // *Наукове видання*. – Луцьк: Волинська обласна друкарня – 2008. – 470 с. 2. *Advanced Micro Devices, AMD - Processor Homepage* [Електронний ресурс]. – Режим доступу: <http://amd.com>. 3. Николайчук Я.М. *Теорія джерел інформації* / Я.М. Николайчук. – Видання друге, виправлене. – Тернопіль: ТзОВ “Терно-граф”. – 2010. – 536 с. 4. Круцкевич Н.Д. *Принципи побудови RCG процесора* / Н.Д. Круцкевич, Я.М. Николайчук // *Тези міжнар. науково-техн. конф. “Контроль і управління в складних системах” (КУСС - 2003)* – Вінниця: “УНІВЕРСУМ – Вінниця”, 2003. – С. 73. 5. Николайчук Я.М. *Теоретичні основи побудови спецпроцесорів у базисі Крестенсона* / Я.М. Николайчук, О.І. Волинський, С.В. Кулина // *Вісник Хмельницького національного університету* – 2007. – № 3. Т.1(93). – С. 85–90. 6. Волинський О.І. *Методи порівняння та сумування в розмежованій системі числення* // *Поступ в науку: зб. праць Бучацького інституту менеджменту і аудиту*. – Бучач. – 2009. – № 4. Т1. – С. 91–94. 7. Волинський О.І. *Швидкодія міжбазисних перетворювачів Радемахера-Крестенсона* / О.І. Волинський // *Збірник матеріалів проблемно-наукової міжгалузевої конференції “Юриспруденція та проблеми інформаційного суспільства” (ЮПИС - 2011)*. Івано-Франківськ, 2011. – С. 71–75. 8. Волинський О.І. *Теорія, алгоритми та спецпроцесори міжбазисних перетворень Радемахера-Крестенсона* / О.І. Волинський // *Поступ в науку: зб. наук. праць Бучацького інституту менеджменту і аудиту*. – Бучач. – 2012. – № 8. – С. 50–54. 9. Николайчук Я.М., Якименко І.З., Воронич А.Р., Волинський О.І. *Пристрій визначення залишку багаторозрядного числа: патент на корисну модель № 68872. МПК G 06 F7/00. Опубл. 10.04.2012. Бюл. № 7*. 10. Николайчук Я.М. *Коди поля Галуа: теорія та застосування: монографія* / Николайчук Я.М. – Тернопіль: ТзОВ “Терно-граф”, 2012. – 576 с.