

Architektura Komputerów

Opracowanie zagadnień na kolokwium z wykładu

Krystian Kolad

Rok akademicki 2017/2018

Spis treści

1	Wzorki	4
1.1	Dodawanie pozycyjne	4
1.2	Odejmowanie pozycyjne	4
1.3	Równania logiczne sumatora binarnego	4
1.4	Równania logiczne subtraktora binarnego	4
1.5	Równania logiczne sumatora prefixowego	4
1.6	Równania logiczne subtraktora prefixowego	4
1.7	Przekształcenie sumatora w subtraktor	4
1.8	Przekształcenie subtraktora w sumator	4
1.9	Obliczanie ilości układów (3,2) w CSA w dodawaniu	5
1.10	Obliczanie ilości układów (3,2) w CSA w mnożeniu	5
1.11	Ilość poziomów redukcji w CSA	5
1.12	Operator przeniesień	5
1.13	Przekodowanie Booth'a	5
1.14	Przekodowanie Booth'a-McSorleya	5
1.15	Liczba przeciwna	6
1.16	Sprawdzanie przekroczenia zakresu w dodawaniu n-pozycyjnych argumentów w U2	6
1.17	Wartość liczby Eulera $\varphi(mn)$	6
1.18	Ile wynosi funkcja Carmichaela dla $f(mn)$	6
2	Algorytmy	7
2.1	Reprezentacja uzupełnieniowa liczby całkowitej	7
2.2	Reprezentacja pozycyjna liczby naturalnej	7
2.3	Reprezentacja ułamka właściwego	7
2.4	Dzielenie w systemie uzupełnieniowym	7
2.5	Dzielenie nieodtwarzające	7
2.6	Mnożenie Baugh'a-Wooley'a	8
2.7	Pierwiastek kwadratowy	8
3	Dowody	9
3.1	Rozmiar iloczynu nie przekracza sumy rozmiarów mnożnej i mnożnika	9
3.2	Lewostronne rozszerzenie argumentów n-pozycyjnych o 1 pozycję zapewnia wytworzenie poprawnej (n+1)-pozycyjnej różnicy	9
3.3	Podaj zależność opisującą zamanę liczby k-bitowej w kodzie U2 na sumę liczby naturalnej i stałej	9
3.4	Inkrementację wartości liczby danej w k-bitowym kodzie $+(2^{k-1}-1)$ można wykonać bezpośrednio na tym kodzie	9
3.5	Suma liczby danej w k-bitowym kodzie $+(2^{k-1}-1)$ oraz jej binarnego dopełnienia ma wartość 1	9
3.6	Sposób sprawdzania poprawności k-bitowej różnicy w kodzie U2 w odejmowaniu liczb k-bitowych danych w kodzie U2	9
3.7	Warunki, dla których $\frac{1}{m \bmod n} = a$	10
3.8	Zależność C6 od C3	10

4	Sumatory	11
4.1	Sumator 16-bitowy kodu NB z modułów subtraktora	11
4.2	11-bitowy kod $+(2^{11} - 1)$ z modułów sumatora	12
4.3	Inkrementer 16-bitowy kodu U2 z modułów subtraktora	13
4.4	Wartość przeciwna (0-X)liczby 15 bitowej z modułów sumatora 4 bit u2	13
4.5	Odwrotność liczby 15-bitowej z modułów subtraktora 4-bitowego	13

1 Wzorki

1.1 Dodawanie pozycyjne

$$s_i = \begin{cases} x_i + y_i + c_i & \text{gdy } x_i + y_i + c_i < B, \text{ wtedy } c_{i+1} = 0 \\ x_i + y_i + c_i - B & \text{gdy } x_i + y_i + c_i \geq B, \text{ wtedy } c_{i+1} = 1 \end{cases}$$

1.2 Odejmowanie pozycyjne

$$s_i = \begin{cases} x_i - y_i - c_i & \text{gdy } x_i - y_i - c_i \geq 0, \text{ wtedy } c_{i+1} = 0 \\ x_i - y_i - c_i + B & \text{gdy } x_i - y_i - c_i < 0, \text{ wtedy } c_{i+1} = 1 \end{cases}$$

1.3 Równania logiczne sumatora binarnego

$$s_i = x_i \oplus y_i \oplus c_i \\ c_{i+1} = x_i * y_i + (x_i \oplus y_i) * c_i \text{ lub } c_{i+1} = x_i * y_i + (x_i + y_i) * c_i$$

1.4 Równania logiczne subtraktora binarnego

$$s_i = \overline{x_i \oplus y_i \oplus c_i} \\ c_{i+1} = \overline{x_i} * y_i + (\overline{x_i} \oplus y_i) * c_i \text{ lub } c_{i+1} = \overline{x_i} * y_i + (\overline{x_i} + y_i) * c_i$$

1.5 Równania logiczne sumatora prefixowego

$$s_i = x_i \oplus y_i \oplus c_i = h_i \oplus c_i \\ c_{i+1} = g_i + p_i * c_i \\ g_i = x_i * y_i \\ p_i = x_i \oplus y_i \text{ lub } p_i = x_i + y_i$$

1.6 Równania logiczne subtraktora prefixowego

$$s_i = \overline{x_i \oplus y_i \oplus c_i} = \overline{h_i \oplus c_i} \\ c_{i+1} = g_i + p_i * c_i \\ g_i = \overline{x_i} * y_i \\ p_i = \overline{x_i} \oplus y_i \text{ lub } p_i = \overline{x_i} + y_i$$

1.7 Przekształcenie sumatora w subtraktor

$$X - Y = \overline{\overline{X} + Y}$$

1.8 Przekształcenie subtraktora w sumator

$$X + Y = \overline{\overline{X} - Y} = \overline{\overline{Y} - X}$$

1.9 Obliczanie ilości układów (3,2) w CSA w dodawaniu

N2:

$$n(k-2)$$

U2:

$$n(k-2) + 2$$

gdzie:

n - liczba bitów

k - liczba liczb

1.10 Obliczanie ilości układów (3,2) w CSA w mnożeniu

$$n^2 - x * 2$$

gdzie:

n - liczba bitów

$$n \leq x \leq 2n - 1$$

Wybieramy najmniejsze x.

1.11 Ilość poziomów redukcji w CSA

Liczymy z wzoru $k_{i+1} = \text{floor}(\frac{3}{2}k_i)$, $k_1 = 3$ Liczymy k_i dopóki k_{i+1} jest mniejsze od liczby liczb wejściowych, a następnie dodajemy 1.

Przykład: dla 24 liczb 8-bitowych

$$k_1 = 3$$

$$k_2 = 4$$

$$k_3 = 6$$

$$k_4 = 9$$

$$k_5 = 13$$

$$k_6 = 19$$

Jako, że $k_7 = 28$, przerywamy, ale musimy z 24 bitów zredukować do 19, więc potrzebujemy jeszcze jeden poziom, przez co mamy 7 poziomów. Podany przykład jest w kodzie N2, w systemie uzupełnieniowym dodajemy jeszcze 1 do wyniku.

1.12 Operator przeniesień

$$(a, p)o(b, q) = (a + pb, pq)$$

1.13 Przekodowanie Booth'a

$$Y_{SD-2} = 2X_{U2} - X_{U2} \text{ lub } y_i = x_{i-1} - x_i$$

1.14 Przekodowanie Booth'a-McSorleya

$$Y_{SD-4} = 2X_{U2} - X_{U2} \text{ czyli } y_i = -2x_{i+1} + x_i + x_{i-1}$$

1.15 Liczba przeciwna

$$-X = 0 - X = \bar{x} + ulp$$

1.16 Sprawdzanie przekroczenia zakresu w dodawaniu n-pozycyjnych argumentów w U2

$$ov = c_n \oplus c_{n-1} = c_n \oplus x_{n-1} \oplus y_{n-1} \oplus s_{n-1}$$

1.17 Wartość liczby Eulera $\varphi(mn)$

$$\varphi(mn) = \varphi(m)\varphi(n), \text{ jeśli } NWD(m, n) = 1$$

1.18 Ile wynosi funkcja Carmichaela dla $f(mn)$

2 Algorytmy

2.1 Reprezentacja uzupełnieniowa liczby całkowitej

1. Weź $Q_0 = X$.
2. Obliczaj $Q_{i+1} = \text{int}(\frac{Q_i}{\beta})$ i $x_i = Q_i \bmod \beta$ dopóki $Q_{i+1} \neq Q_i$.
3. Wszystkie kolejne $x_{i+1} = x_i$

2.2 Reprezentacja pozycyjna liczby naturalnej

1. Weź $Q_0 = X$.
2. Obliczaj $Q_{i+1} = \text{int}(\frac{Q_i}{\beta})$ i $x_i = Q_i \bmod \beta$ dopóki $Q_{i+1} \neq Q_i$.
3. Wszystkie kolejne $x_{i+1} = 0$

2.3 Reprezentacja ułamka właściwego

Mając ułamek w postaci $x = 0,n$:

1. Weź $Q_0 = x$
2. Obliczaj $x_{-i} = \text{int}(Q_i * \beta)$ i $Q_{i+1} = Q_i * \beta - x_{-i}$ dopóki $Q_{i+1} \neq 0$ lub do otrzymania pożądanej dokładności

2.4 Dzielenie w systemie uzupełnieniowym

Zakładając, że X - dzielna, Y - dzielnik, Q - wynik, R - kolejna liczba, k - liczba przesunięć:

1. Przeskaluj dzielnik k razy tak, aby $|\beta^{-k} X| < |Y|$
2. Jeżeli znaki dzielnej i dzielnika są takie same, $q_0 = 0$, $R_0 = \beta^{-k} X$, w przeciwnym wypadku $q_0 = \beta - 1$, $R_0 = \beta^{-k} X + Y$
3. Dla $i = 0, 1, 2, \dots$ powtarzamy: Znajdź największe q_{-i-1} , dla którego $R_{i+1} = \beta R_i - q_{-i-1} Y$ oraz $R_{i+1} Y \geq 0$

2.5 Dzielenie nieodtworzące

Zakładając, że X - dzielna, Y - dzielnik, Q - wynik, R - kolejna liczba, k - liczba przesunięć:

1. Przeskaluj dzielnik k razy tak, aby $|\beta^{-k} X| < |Y|$
2. Jeżeli znaki dzielnej i dzielnika są takie same, $q_k = 0$, $R_0 = \beta^{-k} X$, w przeciwnym wypadku $q_k = 1$, $R_0 = \beta^{-k} X + Y$
3. Dla $i = 0, 1, 2, \dots$ powtarzamy: Jeśli znaki R_i i Y są takie same, to $q_{k-i} = 1$, $R_{i+1} = 2R_i - Y$, w przeciwnym wypadku $q_{k-i} = 0$, $R_{i+1} = 2R_i + Y$

2.6 Mnożenie Baugh'a-Wooley'a

Zakładając, że X - mnożna, Y - mnożnik, n - ilość cyfr mnożnej, k - ilość cyfr mnożnika:

1. Dla $i=0,1,\dots,k-2$ zamień każdy iloczyn częściowy $2^i y_i X$ na $2^i y_i X - 2^{i+k-1}$ poprzez zanegowanie najwyższego bitu
2. Zamień iloczyn częściowy $2^{n-1} y_{n-1}(-X)$ na $2^{n-1} y_{n-1}(-X) - 2^{n-1+k-1}$ poprzez zanegowanie najwyższego bitu
3. Oblicz sumę iloczynów częściowych dodając korektę $-2^{n+k-1} + 2^{k-1}$

2.7 Pierwiastek kwadratowy

1. Przeskaluj X tak, aby $R_0 = X * B^{-2k}$
2. Dla $i=\{1,2,\dots\}$ znajdź największe q_i takie, że $R_i = \beta^2 R_{i-1} - (2 * \beta * Q_{i-1} + q_i) * q_i \geq 0$, następnie $Q_i = \beta * Q_{i-1} + q_i$

3 Dowody

3.1 Rozmiar iloczynu nie przekracza sumy rozmiarów mnożnej i mnożnika

$$\begin{array}{c} \text{Jeżeli:} \\ -\frac{1}{2}B^n \leq X < \frac{1}{2}B^n \\ \text{oraz:} \\ -\frac{1}{2}B^m \leq Y < \frac{1}{2}B^m \\ \text{to:} \\ -\frac{1}{2}B^{n+m} < -\frac{1}{4}B^{n+m} < XY \leq \frac{1}{4}B^{n+m} < \frac{1}{2}B^{n+m} \end{array}$$

3.2 Lewostronne rozszerzenie argumentów n-pozycyjnych o 1 pozycję zapewnia wytworzenie poprawnej (n+1)-pozycyjnej różnicy

$$\begin{array}{c} \text{Jeżeli:} \\ -\frac{1}{2}B^n \leq X, Y < \frac{1}{2}B^n \\ \text{to:} \\ -\frac{1}{2}B^{n+1} \leq -B^n < X - Y < B^n \leq \frac{1}{2}B^{n+1} \end{array}$$

3.3 Podaj zależność opisującą zamianę liczby k-bitowej w kodzie U2 na sumę liczby naturalnej i stałej

$$\begin{array}{l} -x_{k-1}2^{k-1} + x_{k-2}2^{k-2} + \dots + x_12^1 + x_0 = \\ -2^{k-1} + 2^{k-1} - x_{k-1}2^{k-1} + x_{k-2}2^{k-2} + \dots + x_12^1 + x_0 = \\ -2^{k-1} + (1 - x_{k-1})2^{k-1} + x_{k-2}2^{k-2} + \dots + x_12^1 + x_0 \end{array}$$

3.4 Inkrementację wartości liczby danej w k-bitowym kodzie $+(2^{k-1} - 1)$ można wykonać bezpośrednio na tym kodzie

3.5 Suma liczby danej w k-bitowym kodzie $+(2^{k-1} - 1)$ oraz jej binarnego dopełnienia ma wartość 1

Jeśli x jest w kodzie $+(2^{k-1} - 1)$, to $-x = \bar{x} - 1$, więc $\bar{x} = -x + 1$, więc $x + \bar{x} = x - x + 1 = 1$

3.6 Sposób sprawdzania poprawności k-bitowej różnicy w kodzie U2 w odejmowaniu liczb k-bitowych danych w kodzie U2

Rozszerzając k-bitową liczbę o 1 w lewo otrzymujemy $x_k = x_{k-1}$ oraz $y_k = y_{k-1}$. Mając s_{k-1} obliczamy $s_k = x_k - y_k - c_k$, jeśli $s_k > 0$ lub $s_k = x_k - y_k - c_k + \beta$, jeśli $s_k < 0$, gdzie c_k jest równe przeniesieniu z s_{k-1} . Jeśli $s_k * s_{k-1} > 0$, odejmowanie jest poprawne.

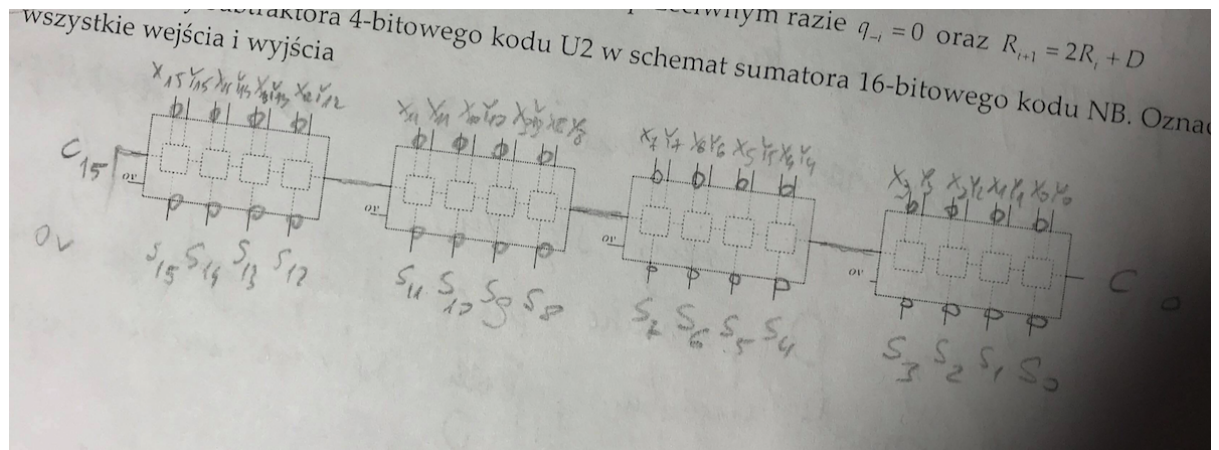
3.7 Warunki, dla których $\frac{1}{m \bmod n} = a$

3.8 Zależność C6 od C3

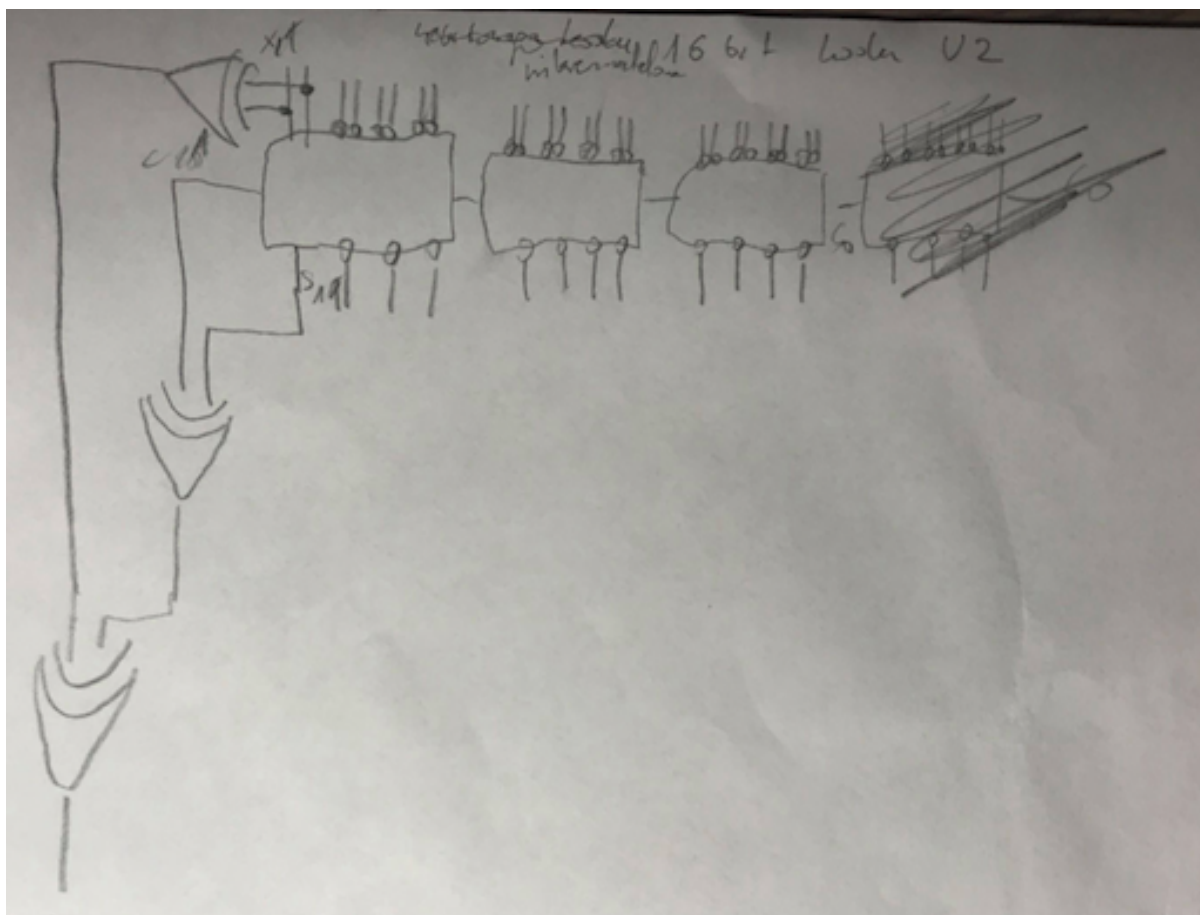
$$C_6 = g_5 + p_5 * c_5 = g_5 + p_5(g_4 + p_4 * c_4) = g_5 + p_5 * (g_4 + p_4 * (g_3 + p_3 * c_3))$$

4 Sumatory

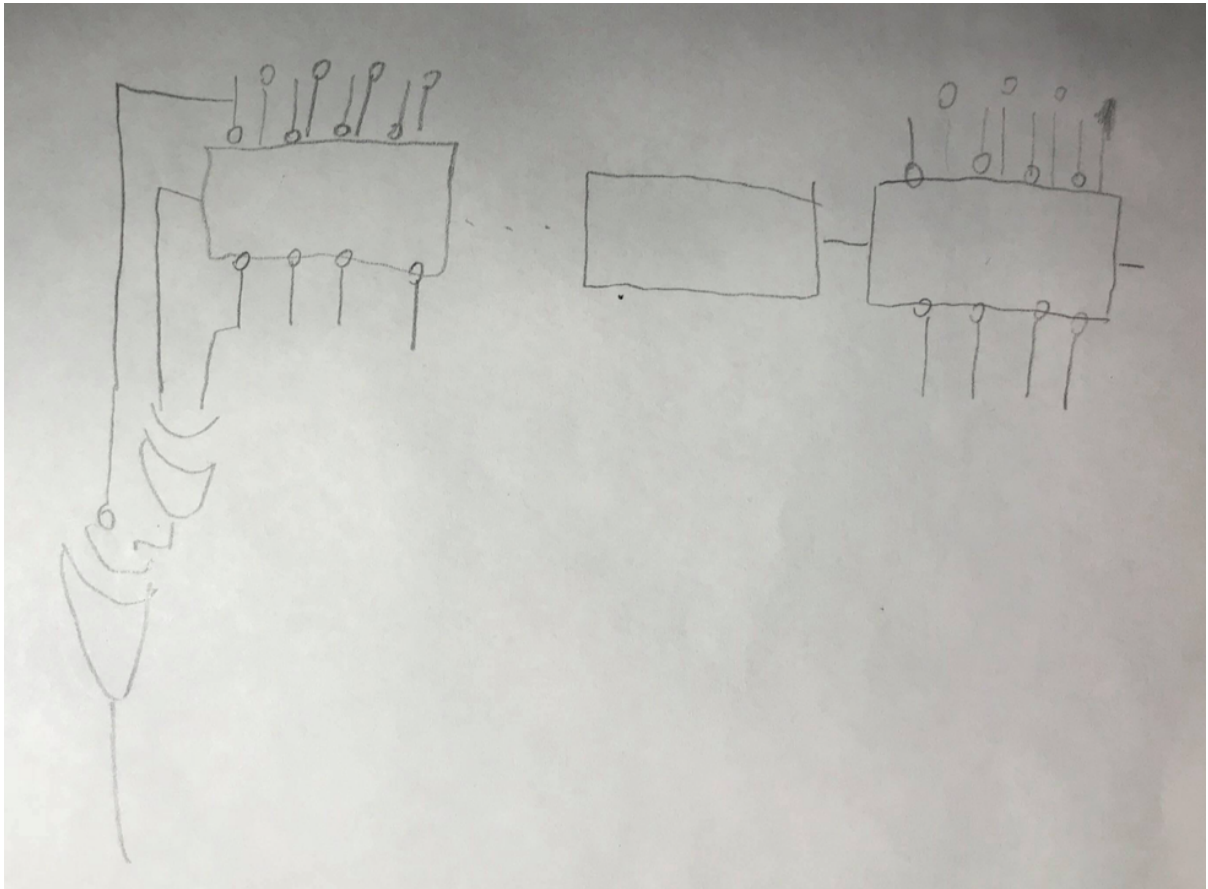
4.1 Sumator 16-bitowy kodu NB z modułów subtraktora



4.2 11-bitowy kod $+(2^{11} - 1)$ z modułów sumatora



4.3 Inkrementer 16-bitowy kodu U2 z modułów subtraktora



4.4 Wartość przeciwna (0-X) liczby 15 bitowej z modułów sumatora 4 bit u2

4.5 Odwrotność liczby 15-bitowej z modułów subtraktora 4-bitowego