

Treść zadań

Zadanie 1

Zbudować sumator dwóch liczb 2-bitowych. Wynik ma być zapisany na 3 bitach.

Zadanie 2

Zbudować sumator dwóch liczb 3-bitowych w systemie U2. Wynikiem ma być liczba 4-bitowa w systemie U2. Wykorzystać gotowy sumator liczb 4 bitowych.

Zadanie 3

Zbudować sumator dwóch liczb 4-bitowych w systemie U2. Wynikiem ma być liczba 4-bitowa w systemie U2. Wykorzystać gotowy sumator liczb 4 bitowych. Wykryć przepełnienie (ang. overflow).

github.com/krzysztiwik/polsl-sprawozdania-tuc

Zadanie 1

	b_0	0	1
a_0			
0		0	1
1		1	0

S_0

$$S_0 = a_0 \oplus b_0$$

	b_0	0	1
a_0			
0		0	0
1		0	1

C_1

$$C_1 = a_0 b_0 = \overline{\overline{a_0 b_0}} = \overline{\overline{a_0} + \overline{b_0}}$$

	c_1	0	1
$a_1 b_1$			
00		0	1
01		1	0
11		0	1
10		1	0

S_1

$$S_1 = a_1 \oplus b_1 \oplus c_1 = (a_1 \oplus b_1) \oplus c_1$$

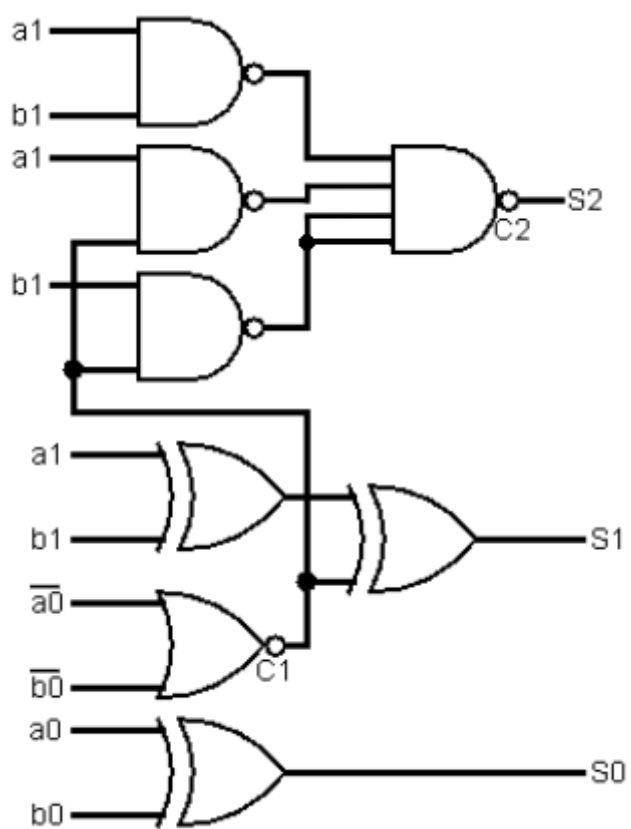
	c_1	0	1
$a_1 b_1$			
00		0	0
01		0	1
11		1	1
10		0	1

C_2

$$\begin{aligned} C_2 &= a_1 b_1 + a_1 c_1 + b_1 c_1 = \overline{\overline{a_1 b_1 + a_1 c_1 + b_1 c_1}} \\ &= \overline{\overline{a_1 b_1} \overline{a_1 c_1} \overline{b_1 c_1}} \\ &= \overline{a_1 b_1 \overline{a_1 c_1} \overline{b_1 c_1} 1} \end{aligned}$$

$$S_2 = c_2$$

Schemat układu:



Zadanie 2

W tabelach **pogrubiono** bity znaku.

Na trzech bitach w zapisie dopełnieniowym można zapisać liczby całkowite w przedziale $[-4, 3]$.

Liczba w systemie dziesiętnym	Liczba w U2
-4	1 00
-3	1 01
-2	1 10
-1	1 11
0	0 00
1	0 01
2	0 10
3	0 11

Suma dwóch takich liczb będzie należała do przedziału $[-8, 6]$.

Liczba w systemie dziesiętnym	Liczba w U2
-8	1 000
-7	1 001
-6	1 010
-5	1 011
-4	1 100
-3	1 101
-2	1 110
-1	1 111
0	0 000
1	0 001
2	0 010
3	0 011
4	0 100
5	0 101
6	0 110

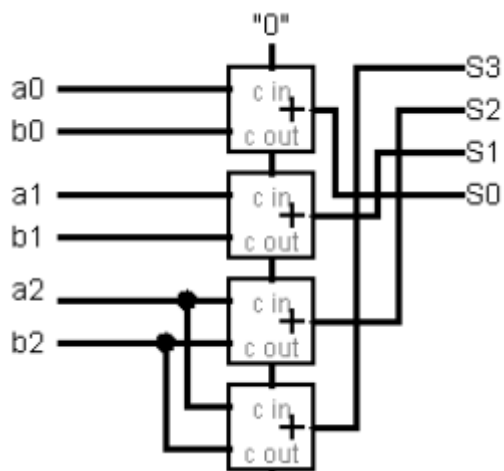
Zgodnie z teorią arytmetyki systemów cyfrowych sumator liczb w naturalnym kodzie binarnym będzie poprawnie dodawał również, dla liczb w zapisie dopełnieniowym. Jednakże, sumator, który należy użyć w zadaniu dodaje liczby 4-bitowe, a nie 3-bitowe. Sumator będzie działał poprawnie, jeżeli poda mu się na wejście liczby 4-bitowe. Można to zrobić w prosty sposób przepisując bit znaku na pozycję najbardziej znaczącego bita.

Liczba w systemie dziesiętnym	Liczba w U2 na 3 bitach	Liczba w U2 na 4 bitach
-4	1 00	1 100
-3	1 01	1 101
-2	1 10	1 110
-1	1 11	1 111
0	0 00	0 000
1	0 01	0 001
2	0 10	0 010
3	0 11	0 011

$$a_3 = a_2$$

$$b_3 = b_2$$

Schemat układu:



Zadanie 3

Jeżeli sumowane są dwie liczby 4-bitowe w U2 i na 4 bitach zapisywany jest wynik, może dojść do przepełnienia (ang. overflow), co skutkuje błędnym wynikiem. Wystąpi ono wtedy, gdy do zapisania wyniku nie wystarczą 4 bity na niego przeznaczone. Przepełnienie można za każdym razem wykryć za pomocą prostego warunku:

Jeżeli dodawane są dwie liczby dodatnie, a suma jest ujemna lub dodawane są dwie liczby ujemne, a suma jest dodatnia, to sygnalizować przepełnienie.

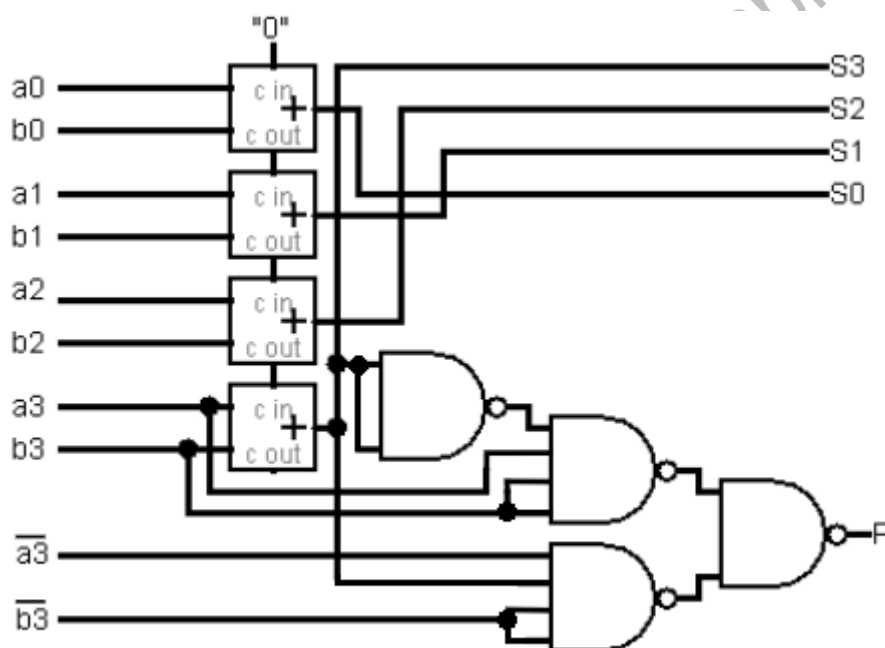
a_3 – bit znaku pierwszego składnika

b_3 – bit znaku drugiego składnika

s_3 – bit znaku sumy

$$P = \overline{a_3} \overline{b_3} s_3 + a_3 b_3 \overline{s_3} = \overline{\overline{a_3} \overline{b_3} s_3} + \overline{a_3 b_3 \overline{s_3}} = \overline{\overline{a_3} \overline{b_3} s_3} \overline{a_3 b_3 \overline{s_3}} = \overline{\overline{a_3} \overline{b_3} s_3 1} \overline{a_3 b_3 \overline{s_3} 1}$$

Schemat układu:



Wnioski

Podczas laboratorium zbudowaliśmy, uruchomiliśmy i przetestowaliśmy wszystkie opisane układy. Działały poprawnie. System U2 umożliwia optymalne zapisywanie liczb ujemnych i dodatnich oraz uniknięcie problemów związanych z istnieniem dwóch reprezentacji zera. Sumatora stworzonego do dodawania liczb dodatnich w naturalnym kodzie binarnym można bez problemu użyć do sumowania zarówno dodatnich, jak i ujemnych liczb w zapisie dopełnieniowym. Dzięki jednolitym operacjom dodawania i odejmowania, system U2 upraszcza projektowanie układów arytmetycznych. Należy uważać na przepełnienie (ang. overflow), ponieważ prowadzi ono do błędnego wyniku. W prosty sposób można wykryć przypadki przepełnienia podczas dodawania liczb w U2 i poinformować o nich.

github.com/krzysztiwik/polsl-sprawozdania-tuc