

# **ĆWICZENIE nr 6**

## **UKŁADY ARYTMETYCZNE cz. I**

Dokument służy wyłącznie do celów indywidualnego kształcenia.  
Zabrania się utrwalania, przekazywania osobom trzecim oraz rozpowszechniania.

## 1.1 Cel ćwiczenia.

Celem ćwiczenia jest zapoznanie się z budową i działaniem sumatorów, subtraktorów i komparatorów cyfrowych.

## 1.2 Wprowadzenie.

Układy cyfrowe mogą przetwarzać tylko informację binarną, tj. dwuwartościową. Wynika stąd konieczność przejścia z ogólnie przyjętego systemu dziesiętnego na system binarny. Liczby dziesiętne zazwyczaj przedstawia się w naturalnym kodzie dwójkowym, w kodzie ósemkowym (oktalnym), szesnastkowym (heksadecymalnym) lub w kodzie BCD.

Jeżeli przy zapisie liczby binarnej uwzględnia się znak to zapisujemy go w postaci bitu przed najstarszym bitem liczby. Liczbie dodatniej odpowiada bit znaku równy 0, dla liczby ujemnej równy 1. Liczby binarne przedstawia się w zapisach: znak-moduł, uzupełnień do 1 (p.-1), uzupełnień do 2 (p.).

**Liczba w postaci znak-moduł.** Liczba ujemna daje się łatwo przedstawić, jeżeli na najstarszej pozycji zostanie umieszczony bit znaku  $s$ . Zero oznacza liczbę dodatnią, a jedynka ujemną. Poprawna interpretacja znaku liczby może nastąpić tylko przy stałej długości słowa. Przykład dla słowa 8-bitowego:

$$+118_{10} = \mathbf{0} \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0_2$$

$$-118_{10} = \mathbf{1} \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0_2$$

**Liczba w postaci uzupełnienia do dwóch (two's complement).** Przedstawienie liczby w postaci znak-moduł ma tę wadę, że nie ułatwia dodawania liczb o przeciwnych znakach. W przypadku wystąpienia znaku minus sumator musi być przestawiony na odejmowanie. Przy zapisie w postaci uzupełnienia do dwóch nie jest to potrzebne. W zapisie w postaci uzupełnienia do dwóch najstarszy bit ma wagę ujemną. Pozostała część liczby jest normalnym kodem dwójkowym. Również w tym zapisie długość słowa musi być stała, aby można było jednoznacznie zdefiniować najstarszy bit. W przypadku liczby dodatniej wartość najstarszego bitu wynosi 0. Dla liczby ujemnej bit ten ma wartość 1, bo tylko ta pozycja ma ujemną wagę. Przykład dla słowa 8-bitowego:

$$+118_{10} = \mathbf{0} \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 = B_N$$

$$-118_{10} = \mathbf{1} \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 = X$$

Przejście z liczby dodatniej na równą co do modułu liczbę ujemną jest oczywiście nieco trudniejsze niż przy zapisie znak-moduł. Załóżmy, że liczba dwójkowa  $B_N$  ma bez bitu znaku długość  $N$ . Stąd waga pozycji znaku wynosi  $-2^N$ . Liczbę  $-B_N$  przedstawia wzór:

$$-B_N = -2^N + X$$

Stąd wartość dodatniej reszty  $X$

$$X = 2^N - B_N$$

To wyrażenie nosi nazwę uzupełnienia do dwóch  $B_N^{(2)}$  liczby  $B_N$ . Można je łatwo wyliczyć na podstawie  $B_N$ . Rozważmy największą liczbę, jaką można przedstawić za pomocą  $N$  pozycji. Ma ona wartość  $11111\dots=2^N - 1$ . Odejmując od niej dowolną liczbę dwójkową  $B_N$  otrzymujemy również liczbę dwójkową, w której wszystkie bity są negacjami odpowiednich bitów  $B_N$ . Liczba taka nazywa się uzupełnieniem do jedności  $B_N^{(1)}$  liczby  $B_N$ . Stąd mamy:

$$B_N^{(1)} = 2^N - 1 - B_N \quad \text{i} \quad B_N^{(2)} = B_N^{(1)} + 1$$

Zapis liczby dwójkowej w postaci uzupełnienia do dwóch otrzymujemy więc przez negację wszystkich jej pozycji oraz dodanie 1. Przykład 8-bitowej liczby dwójkowej w uzupełnieniu do dwóch:

$$\begin{array}{r}
 118_{10} = \quad 01110110 \\
 \text{uzupełnienie do jedności} \quad 10001001 \\
 \quad \quad \quad \quad \quad \quad + 1 \\
 \text{uzupełnienie do dwóch} \quad 10001010 = -118_{10} \\
 \text{Przejście odwrotne:} \\
 \text{uzupełnienie do jedności} \quad 01110101 \\
 \quad \quad \quad \quad \quad \quad + 1 \\
 \text{uzupełnienie do dwóch} \quad 01110110 = +118_{10}
 \end{array}$$

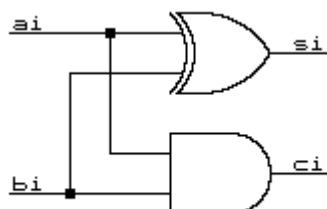
**Sumatory** są układami dodającymi dwie liczby binarne. Najprostszy przypadek występuje wówczas, gdy trzeba dodać dwie liczby jednobitowe. Aby móc zaprojektować odpowiedni układ logiczny, należy rozpatrzyć wszystkie możliwe przypadki i utworzyć tablice funkcji logicznych. Przy dodawaniu dwóch liczb jednobitowych mogą wystąpić następujące przypadki:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$ .

Jeżeli zarówno A, jak i B są jedynkami, przy dodawaniu występuje przeniesienie do starszej pozycji. Wobec tego sumator musi mieć dwa wyjścia: jedno wyjście dla sumy na odpowiadającej sumatorowi pozycji i jedno wyjście przeniesienia do następnej pozycji. Dochodząc do tablicy prawdy przedstawionej na Rys. 1. przydzielamy liczbom A i B zmienne logiczne  $a_i$  i  $b_i$ . Przeniesienie oznaczmy przez  $c_i$ , a sumę jako  $s_i$ . Otrzymujemy stąd funkcje Boole'owskie w postaci normalnej sumy  $c_i = a_i b_i$  oraz  $s_i = \bar{a}_i b_i + a_i \bar{b}_i = a_i \oplus b_i$ .

$a_i$	$b_i$	$s_i$	$c_i$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**Rys. 1. Tablica prawdy półsumatora.**

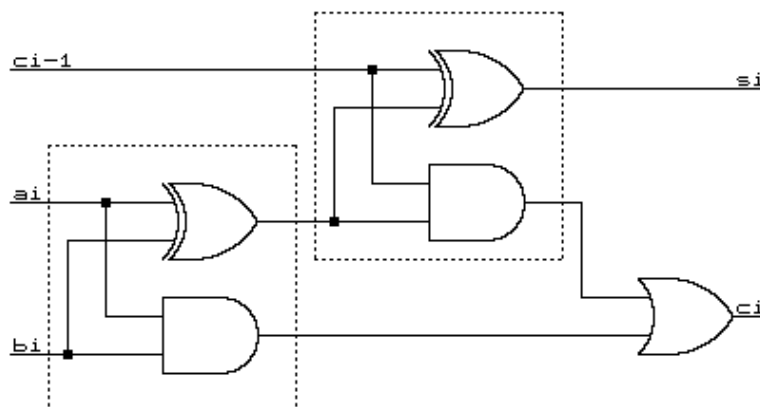
Przeniesienie jest iloczynem zmiennych wejściowych (funkcja AND), suma zaś funkcją nierównoważności (EXOR). Układ, który realizuje te dwie zależności, nazywamy półsumatorem (Rys. 2).



**Rys. 2. Układ półsumatora.**

W przypadku dodawania liczb dwójkowych wielopozycyjnych półsumator można zastosować tylko na najmłodszej pozycji. Na wszystkich pozostałych trzeba

dodawać nie dwa, ale trzy bity z powodu przeniesienia z młodszej pozycji. W ogólnym przypadku dla każdego bitu jest potrzebny układ logiczny o trzech wejściach  $a_i$ ,  $b_i$ ,  $c_{i-1}$  oraz dwóch wyjściach  $s_i$  i  $c_i$ . Układ taki nosi nazwę pełnego sumatora. Można go zrealizować w sposób pokazany na Rys. 3. za pomocą dwóch półsumatorów. Tablica prawdy pełnego sumatora jest przedstawiona na Rys. 4. Funkcje Boole'owskie są następujące  $c_i = a_i b_i + a_i c_{i-1} + b_i c_{i-1}$  oraz  $s_i = a_i \oplus b_i \oplus c_{i-1}$ .



Rys. 3. Układ pełnego sumatora.

Wejście			Wyjście		
$a_i$	$b_i$	$c_{i-1}$	$s_i$	$c_i$	
0	0	0	0	0	
0	1	0	1	0	
1	0	0	1	0	
1	1	0	0	1	
0	0	1	1	0	
0	1	1	0	1	
1	0	1	0	1	
1	1	1	1	1	

Rys. 4. Tablica prawdy pełnego sumatora.

Aby móc dodawać dwie wielopozycyjne liczby dwójkowe potrzeba na każdą pozycję jeden pełny sumator. Sumatory wielobitowe można podzielić na dwójkowe (sumowanie liczb binarnych) i dziesiętne (sumowanie liczb dziesiętnych kodowanych dwójkowo). Działanie sumatora może odbywać się szeregowo (sumowane są kolejne bity dodanej i dodajnika) lub równoległe (wszystkie bity sumowane są jednocześnie).

**Subtraktor** jest układem realizującym odejmowanie dwu liczb binarnych. Subtraktor można nazwać pewnym przypadkiem sumatora ze zmianą  $a_i$  na  $\bar{a}_i$  w funkcji pożyczki. Układy wielobitowe subtraktorów budowane są analogicznie jak odpowiednie sumatory. Praktycznie do budowy wielobitowych układów odejmujących wykorzystuje się scalone sumatory uzupełnione dodatkowymi układami wejściowymi, wyjściowymi i generującymi znak wyniku. Tablica prawdy pełnego subtraktora jest przedstawiona na Rys. 5.

Wejście			Wyjście	
$a_i$	$b_i$	$v_{i-1}$	$d_i$	$v_i$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1

**Rys. 5. Tablica prawdy pełnego subtraktora.**

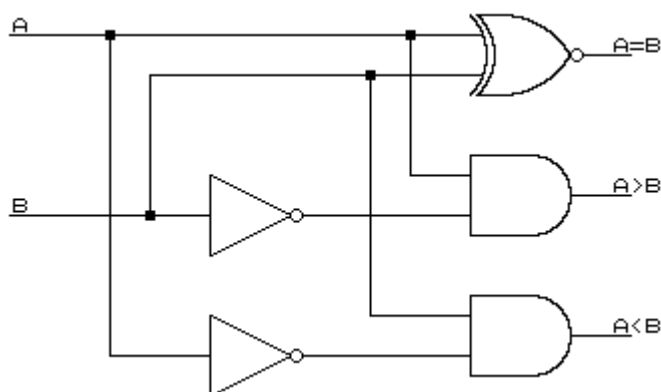
**Komparatory** to układy logiczne, które porównują ze sobą liczby. Najważniejsze kryteria porównania to  $A=B$ ,  $A>B$  i  $A<B$ . Kryterium równości dwóch liczb dwójkowych to identyczność wszystkich bitów. Komparator powinien dawać na wyjściu 1 wówczas, gdy dwie porównywane liczby są sobie równe. W przypadku dwóch liczb 1-bitowych wystarczy użyć bramki EXOR. Bardziej uniwersalne komparatory prócz wykrywania równości wskazują, która z porównywanych liczb jest większa. Określa się je mianem komparatora wartości.

Dla utworzenia funkcji logicznych należy skorzystać z przedstawionej na Rys. 6. tablicy stanu.

A	B	$A>B$	$A=B$	$A<B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

**Rys. 6. Tablica stanu komparatora 1-bitowego.**

Na jej podstawie otrzymamy bezpośrednio układ realizujący tę funkcję, przedstawiony na Rys. 7.



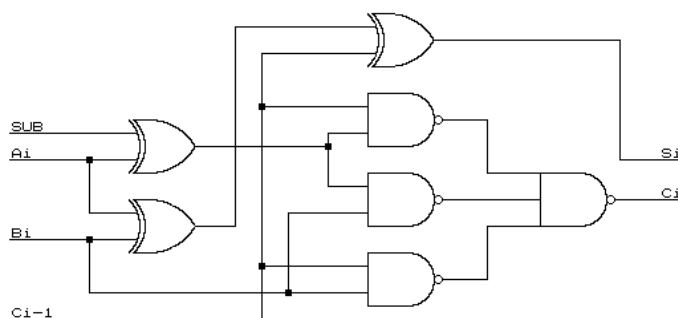
**Rys. 7. Schemat komparatora 1-bitowego.**

### 1.3 Pytania sprawdzające.

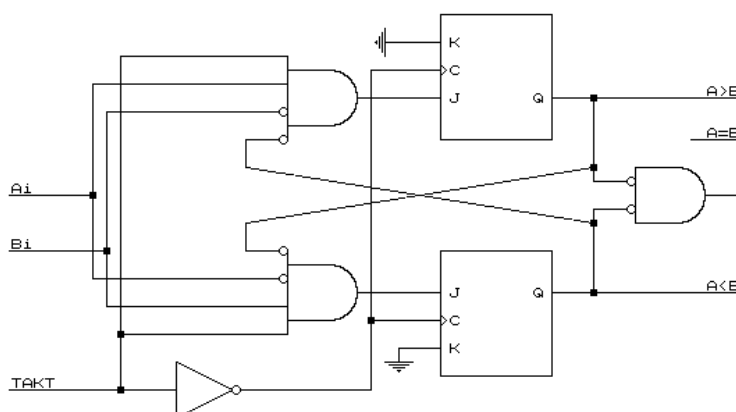
1. Omówić uzupełnienie do dwóch liczb ujemnych.
2. Przedstawić liczby 5, 9, 15, 28, 42 oraz ich uzupełnienie do dwóch.
3. Omówić działanie i narysować sumator i subtraktor 1-bitowy.
4. Dokonać klasyfikacji sumatorów.
5. Omówić budowę zasadę działania komparatora 1-bitowego.

### 1.4 Opis układu pomiarowego.

Część pierwsza ćwiczenia składa się z układu sumatora-subtraktora 1-bitowego, komparatora szeregowego oraz układów pomocniczych takich jak rejestry przesuwne, przerzutnik D oraz rejestr sumy. Na Rys. 8. przedstawiony jest schemat sumatora-subtraktora 1-bitowego wykorzystywanego w ćwiczeniu, a na Rys. 9 schemat komparatora szeregowego. Przy pomocy przełącznika SUB można wykorzystywać układ jako sumator lub subtraktor. Taktowania dokonuje przy pomocy przełącznika TAKT.



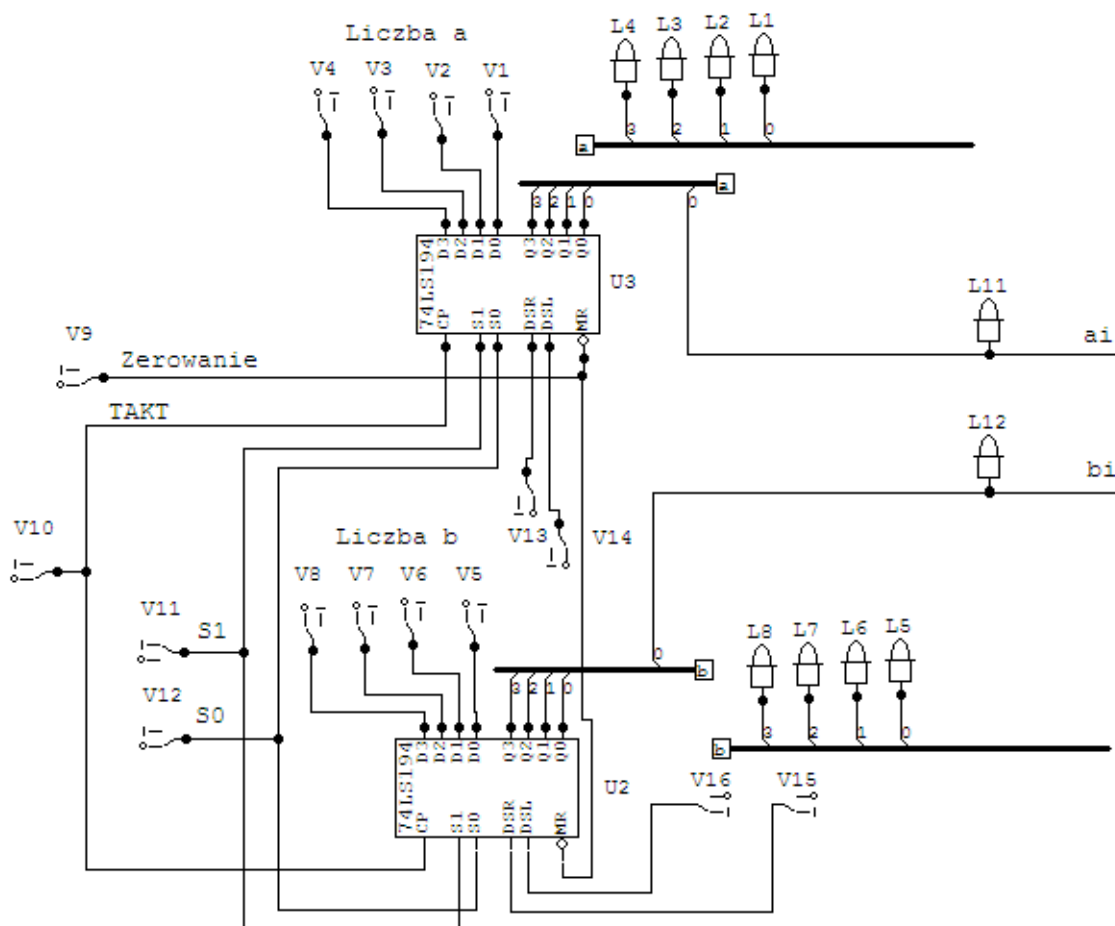
Rys. 8. Schemat układu sumatora-subtraktora 1-bitowego.



Rys. 9. Schemat układu komparatora szeregowego.

### 1.5 Program ćwiczenia.

Przedstawiony na rysunku 10 zadajnik ma za zadanie podawać dwie liczby czterobitowe bit po bicie do badanego układu. Zadajnik zbudowany jest z dwóch rejestrów przesuwanych (U3- dla liczby A i U2 – dla liczby B) pracujących synchronicznie. Układ zadajnika będzie wykorzystany przy badaniu poszczególnych układów arytmetycznych.



**Rys.10. Schemat zadajnika.**

Znaczenie poszczególnych przełączników układu jest następujące:

Przełącznik V12, V11 (S1, S0):

S0:S1 = 0:0 Blokada zegara – nic nie jest wykonywane.

S0:S1 = 1:0 Przesuw w prawo: w kierunku od Q0 do Q3.

S0:S1 = 0:1 Przesuw w lewo: w kierunku od Q3 do Q0.

S0:S1 = 1:1 Wpis równoległy

Zmiana trybu pracy powinna się odbywać przy stanie wysokim na wejściu zegarowym.

Przełącznik V10, (CP – TAKT)

Przełącznik V9, (MR – Zerowanie) Aktywny stan niski

Przełącznik V13, V15 (DSR – wejście danych szeregowych przy przesuwaniu danych w prawo.)

Przełącznik V14, V16 (DSL – wejście danych szeregowych przy przesuwaniu danych w lewo).

Przełącznik V1- V4, (D0-D3 liczba A)

Przełącznik V5- V8, (D0-D3 liczba B)

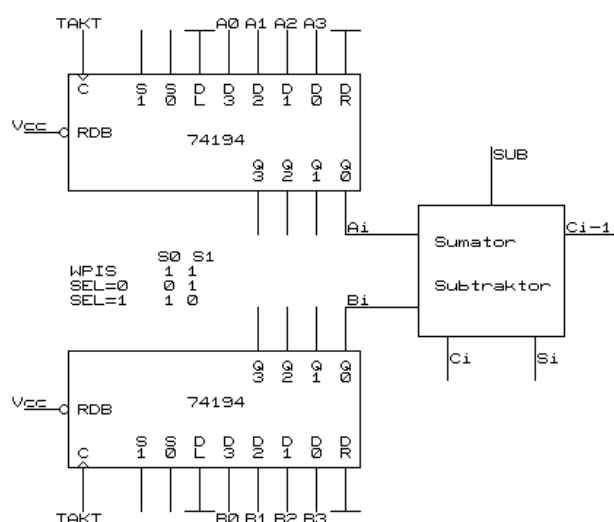
$a_i, b_i$  – wyjścia zadajnika

L1...L4 – wyjścia równoległe (liczba A).

L5...L8 – wyjścia równoległe (liczba B).

### 1.5.1 Sumator-subtraktor 1-bitowy.

Zmontować układ według rysunku 11 korzystając z zadajnika przedstawionego na rysunku 10. Wyzerować układ przy pomocy zadajnika V9 (Zerowanie). Połączyć punkty SUB i  $C_{i-1}$  ze stanem L a do  $S_i, C_i, C_{i-1}$  podłączyć wskaźniki (*logic display*). Ustawić na zadajnikach A=1100 i B 1010. Ustawić przełącznik V11=1, V12=1 i podać jeden impuls zegarowy (TAKT). Następnie ustawić V11=1, V12=0 i podawać impulsy zegarowe przy pomocy przełącznika V10 TAKT. Wyniki należy wpisać do Tabela 1 ( $A_i - L11\ 12, B_i - L12, S_i, C_i, C_{i-1}$ ). Następnie powtórzyć badanie dla  $C_{i-1}=H$ .



**Rys. 11. Układ do badania sumatora-subtraktora 1-bitowego.**

**Tabela 1.**

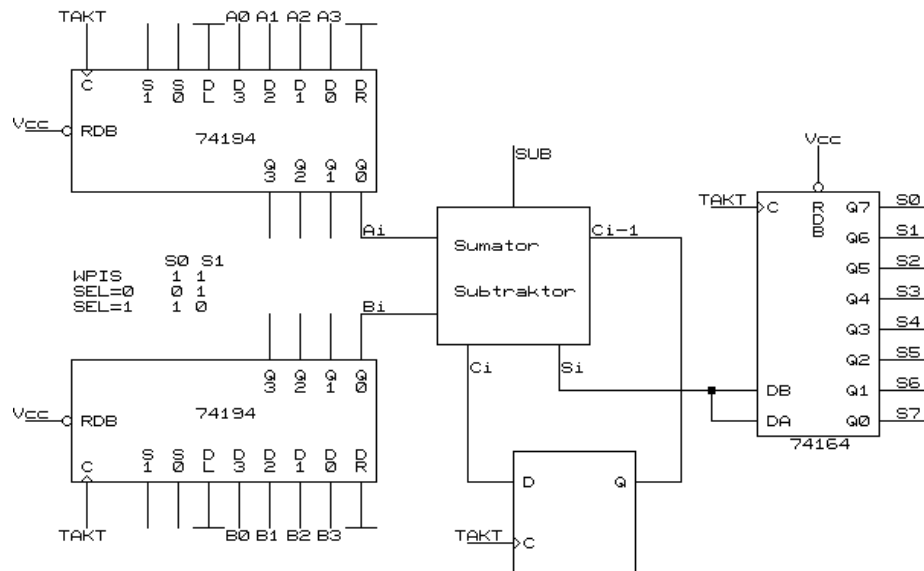
Wejście		Wyjście	
$a_i$	$b_i$	$S_i$	$C_i$
0	0		
0	1		
1	0		
1	1		

Po wykonaniu badania sumatora 1-bitowego należy połączyć punkt SUB ze stanem H (załączenie subtraktora) i dla różnych wartości  $C_{i-1}$  powtórzyć badanie jak dla sumatora.



### 1.5.2 Sumator-subtraktor szeregowy.

Zmontować układ według rysunku 12 korzystając z zadajnika przedstawionego na rysunku 10. Wyzerować układ przy pomocy zadajnika V9 (Zerowanie). Połączyć punkty SUB ze stanem L a do  $S_i$ ,  $C_i$ ,  $C_{i-1}$  podłączyć wskaźniki (*logic display*). Ustawić na zadajnikach A=1100 i B 1010. Ustawić przełącznik V11=1, V12=1 i podać jeden impuls zegarowy (TAKT). Następnie ustawić V11=1, V12=0 i podawać impulsy zegarowe przy pomocy przełącznika V10 TAKT. . Podczas taktowania należy zwrócić uwagę na stan wskaźników LED ( $A_i$  – L11 12,  $B_i$  – L12,  $S_i$ ,  $C_i$ ,  $C_{i-1}$ ). Wynik odczytać w rejestrze sumy po odpowiedniej liczbie impulsów taktujących. Z kolei łącząc punkt SUB ze stanem L (sumowanie) lub H (odejmowanie) przeprowadzić następujące działania arytmetyczne dla dwóch różnych liczb A i dwóch różnych liczb B:  $A+0=A$ ,  $A+B=7$ ,  $A+B=S$ ,  $A-0=A$ ,  $A-B=S$  ( $S>0$ ),  $A-B=S$  ( $S<0$ ),  $0-B= -B$ . Przed wykonaniem każdego działania układ należy wyzerować. Wyniki przedstawić w Tabela 2.



**Rys. 12. Układ sumatora-subtraktora szeregowego.**

**Tabela 2.**

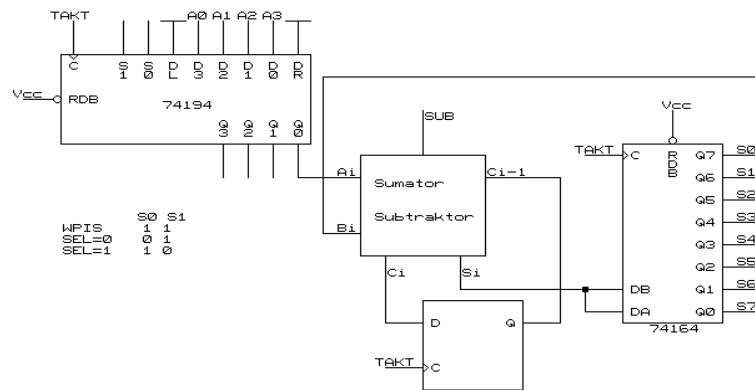
Binarnie A				Binarnie B				Działanie	Binarnie S								Hex S	Ilość Taktów
A3	A2	A1	A0	B3	B2	B1	B0		S7	S6	S5	S4	S3	S2	S1	S0		
								$A+0=A$										
								...										
								$0-B = -B$										

### 1.5.3 Akumulacyjny sumator szeregowy.

Zmontować układ według rysunku 13, wyzerować układ. Połączyć punkt SUB ze stanem L. Ustawić na zadajniku A=0001. Ustawić przełącznik V11=1, V12=1 i podać jeden impuls zegarowy (TAKT). Następnie ustawić V11=1, V12=0 i podawać impulsy zegarowe przy pomocy przełącznika V10 TAKT. Podczas taktowania należy zwrócić uwagę na stan wskaźników LED ( $A_i$  – L11 12,  $B_i$  – L12,  $S_i$ ,  $C_i$ ,  $C_{i-1}$ ). Wynik należy

odczytać w rejestrze sumy po ośmiu impulsach taktujących. Przed 8 impulsem należy ustawić przełącznik  $V11=1$ ,  $V12=1$ , na zadajniku A zadać następny składnik sumowania i podać jeden impuls zegarowy (TAKT). Następnie ustawić  $V11=1$ ,  $V12=0$  i podawać impulsy zegarowe przy pomocy przełącznika V10 TAKT

Z kolei łącząc punkt SUB ze stanem L (sumowanie) lub H (odejmowanie) przeprowadzić następujące działania arytmetyczne dla dwóch różnych liczb A:  $A+A=2A$ ,  $A_1+A_2+A_3=S$ ,  $A_1-A_2+A_3=S$ ,  $A_1+A_2+\dots+A_n=S$  oraz własne trzy działania. Przed wykonaniem każdego działania układ należy wyzerować. Wyniki przedstawić w Tabela 3.



**Rys. 13. Układ sumatora akumulującego.**

**Tabela 3.**

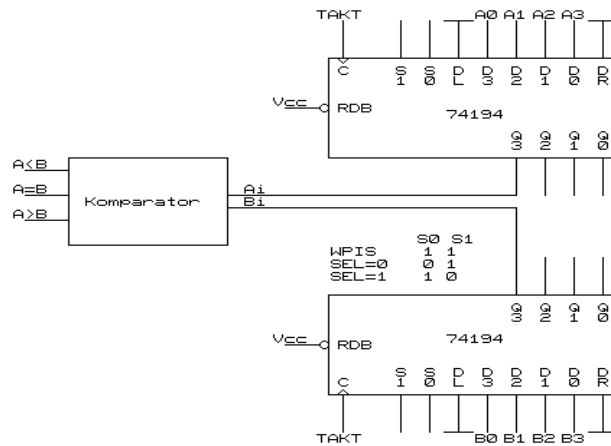
Binarnie A				Działanie	Binarnie S								Hex S	Ilość Taktów
A3	A2	A1	A0		S7	S6	S5	S4	S3	S2	S1	S0		
				$A+0=S$ $A+S=2A$										
				...										
				$A_1+0=S$ ...										
				$A_n+S=S_n$										

#### 1.5.4 Komparator szeregowy.

Zmontować układ według rysunku 14. Do wyjść komparatora podłączyć wskaźniki (*logic display*), wyzerować układ. Ustawić na zadajnikach  $A=0011$  i  $B=0010$ . Ustawić przełącznik  $V11=1$ ,  $V12=1$  i podać jeden impuls zegarowy (TAKT). Następnie ustawić  $V11=1$ ,  $V12=0$  i podawać impulsy zegarowe przy pomocy przełącznika V10 TAKT. Podczas taktowania należy zwrócić uwagę na stan wskaźników LED ( $A_i$ ,  $B_i$ ,  $A>B$ ,  $A=B$ ,  $A<B$ ). Wyniki przedstawić w Tabela 4.

**Tabela 4.**

A	B	A <sub>i</sub>	B <sub>i</sub>	A>B	A=B	A<B	TAKT
0011	0010	0	0				1
		0	0				2
		1	1				3
		1	0				4
...	...						



**Rys. 14. Układ komparatora szeregowego MSB.**