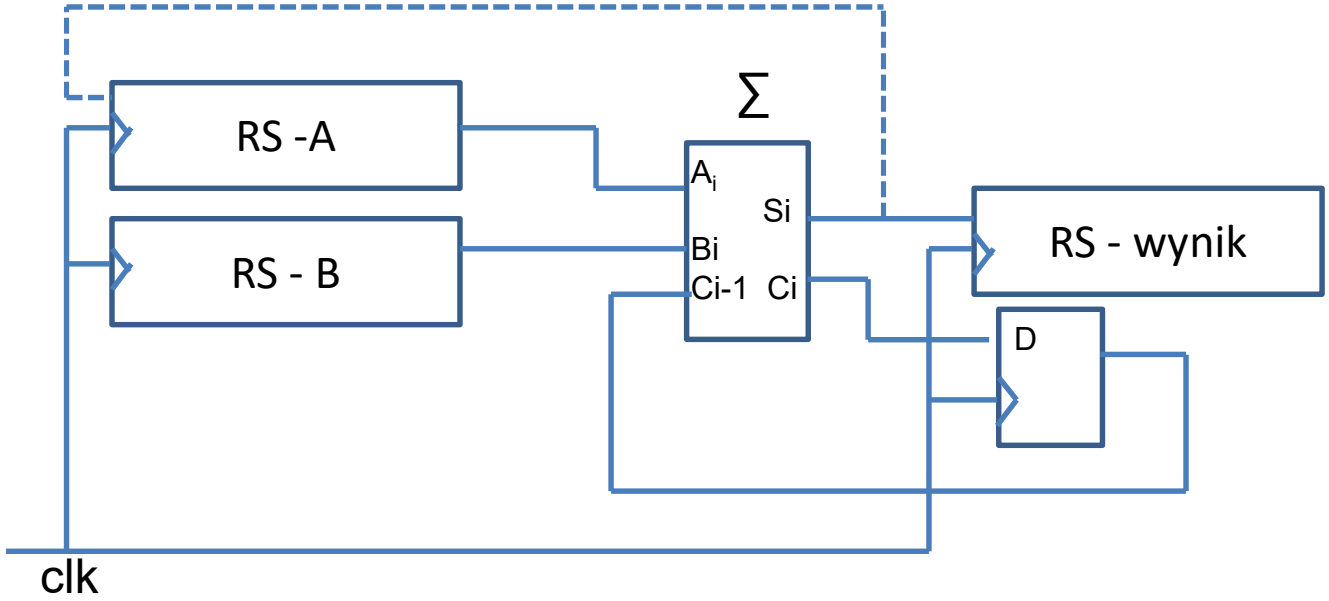
Układy arytmetyczne
dodawanie i mnożenie

Podział sumatorów

- Równoległe:
  - Z przeniesieniem propagowanym szeregowo („przeniesienie szeregowe”)
  - Z przeniesieniem propagowanym równolegle („przeniesienie równoległe”)
- Szeregowe (układy sekwencyjne)
  - Zwykłe
  - Akumulujące

Układy iteracyjne – sumator szeregowy

iteracja w czasie

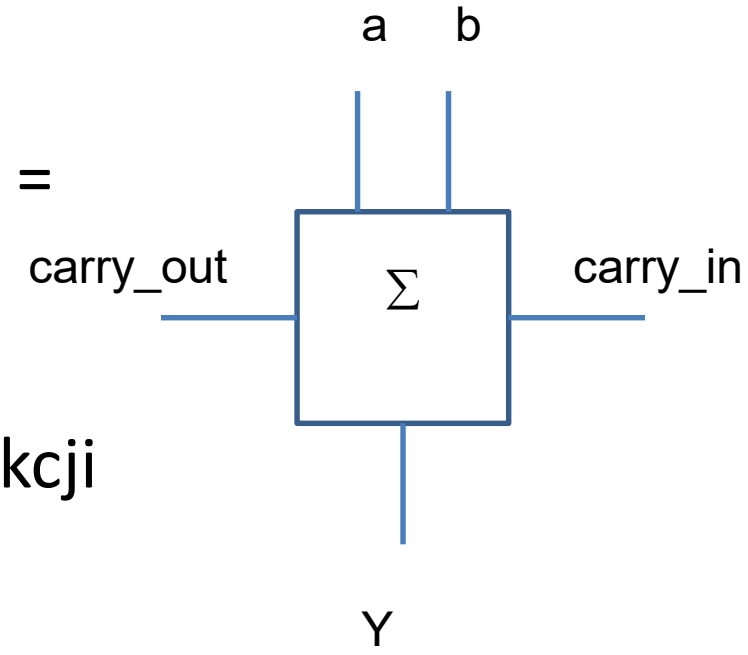


Sumator jednobitowy pełny:

- generuje wynik sumowania dla liczb dowolnego rozmiaru w czasie zależnym od rozmiaru liczb,
- liczby podawane są począwszy od najmłodszego bitu,
- linie przerywane to wersja układu będąca sumatorem akumulacyjnym – bez dodatkowego rejestru dla wyniku.

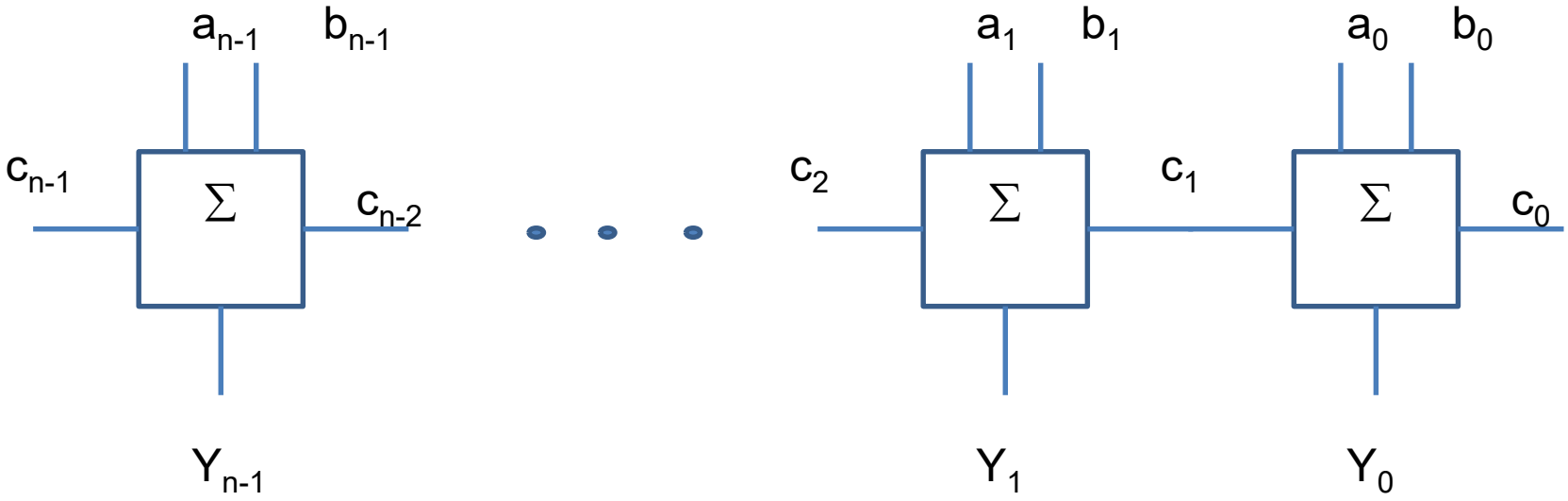
Sumator jednobitowy pełny

- Sumator jednobitowy pełny
- Y=a ⊕ b ⊕ carry_in
- carry_out =ab+ (a+b)carry_in = ab+(a⊕b) carry_in
- Zależności wynikają z tablicy prawdy dla funkcji sumy i funkcji przeniesienia



Sumator

z przeniesieniem szeregowym



Czas działania układu o takiej strukturze jest sumą czasów działania układów składowych ze względu na sekwencyjną propagację przeniesienia - 2 poziomy bramkowania na jeden bit.

Sumator z przeniesieniami równoległymi

- G_i - Warunek generacji przeniesienia – gdy 2 sumowane bity są „1”
- P_i - Warunek propagacji przeniesienia – gdy jeden sumowany bit jest „1”
- Jeżeli $G_i = 1$ to $P_i = 0$
- $G_i = A_i B_i$ $P_i = A_i \oplus B_i$ 1 bramka (funkcje nie są jednocześnie =1)
- $c_{i+1} = G_i + P_i\, c_i$ – przeniesienie generowane lub propagowane
- $Y_i = c_i \oplus P_i$ – suma gdy wyłącznie przeniesienie lub warunek propagacji
- $c_1 = g_1 + p_1 c_0$
- $c_2 = g_2 + p_2 c_1 = g_2 + p_2(g_1 + p_1 c_0) = g_2 + p_2 g_1 + p_2 p_1 c_0$
- $c_3 = g_3 + p_3 c_2 = g_3 + p_3(g_2 + p_2 g_1 + p_2 p_1 c_0) = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 c_0$ 2 bramki
- $c_4 = g_4 + p_4 c_3 =$

$$g_4 + p_4(g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 c_0) = g_4 + p_4 g_3 + p_4\, p_3 g_2 + p_4\, p_3 p_2 g_1 + p_4\, p_3 p_2 p_1 c_0$$

Dla wyznaczenia przeniesienia c_i brak konieczności znajomości wyniku z pozycji wcześniejszej c_{i-1} – możliwość równoległej dla wszystkich pozycji sumatora generacji przeniesień jako sumy iloczynów opartych na sygnałach dostępnych na wejściu.

Sumator z przeniesieniami równoległymi

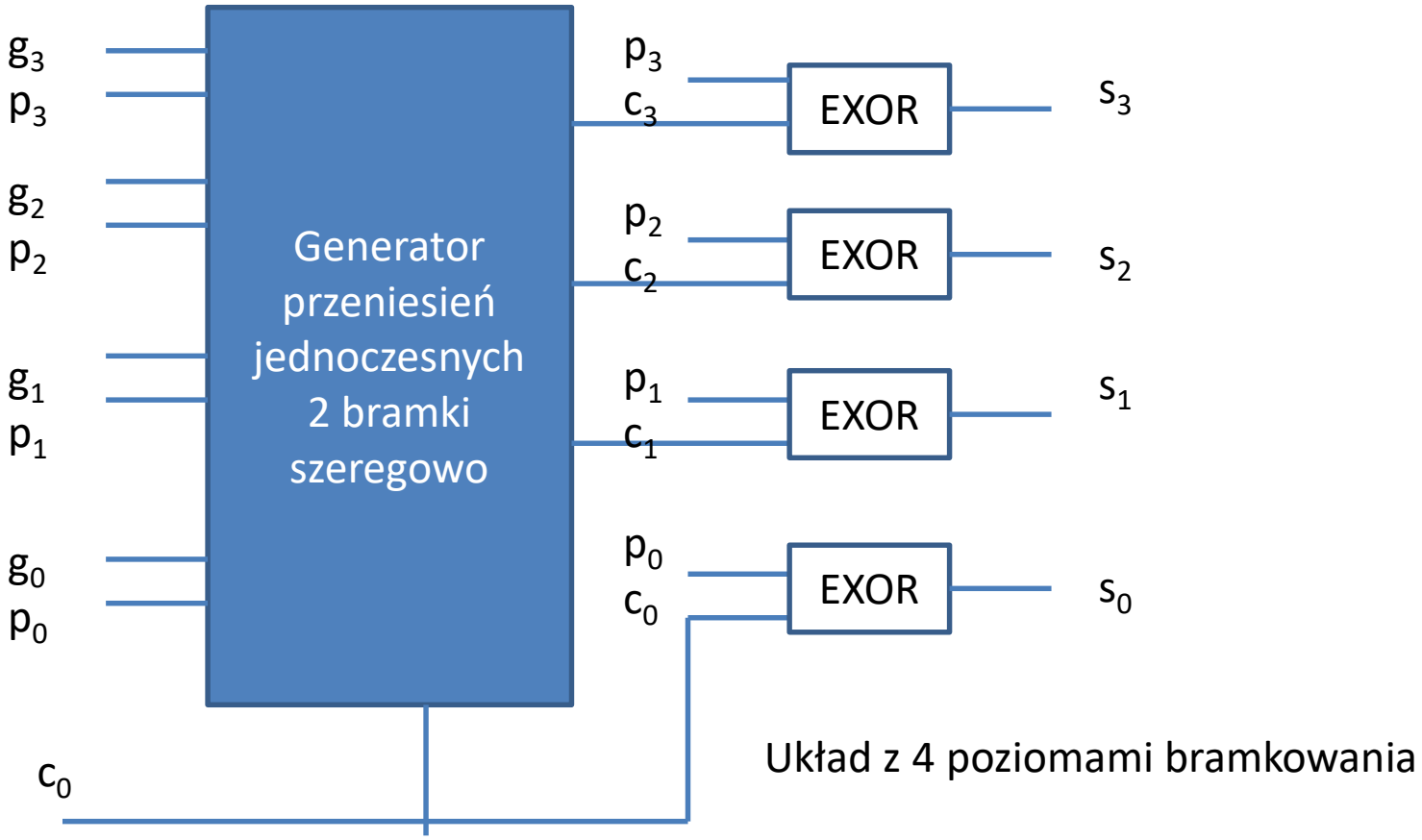
Wykorzystanie podanych równań pozwala na uzyskanie wyniku –

sumy na 4 bitach po przejściu 1+2+1=4 poziomów bramkowania (PB) - 4 czasy propagacji (4 tp)

- wyznaczenie G,P – 1 tp,
- wyznaczenie C - 2 tp (iloczyn i suma),
- wyznaczenie wyniku - 1 tp (bramka exor)

Czas propagacji czterobitowego sumatora z przeniesieniem propagowanym szeregowo to 4*2PB=8 PB.

Sumator z przeniesieniami równoległymi



Układ z 4 poziomami bramkowania

Sumator z przeniesieniami równoległymi 16 bitów

Problem: duża liczba wejść bloku wyznaczającego przeniesienie dla starszych bitów liczby.

Rozwiązanie: równoczesne lokalne przetwarzanie grup bitów liczby i wykorzystanie do otrzymania przeniesienia z bieżącej grupy sygnału przeniesienia wyznaczonego dla młodszej grupy.

Rozważmy sumator 16 bitowy - 4 grupy po 4 bity:

Generator przeniesień wewnątrz grupy:

- Warunek generacji grupy 4 bitów $G_g=G_4+G_3P_4+G_2P_4P_3+G_1P_4P_3P_2$ - (czas 3 t_p)
- Warunek propagacji grupy 4 bitów $P_g=P_4P_3P_2P_1$ (czas 2 t_p)
- Generacja przeniesień na podstawie wzorów poprzedniej strony

Generator przeniesienia grupy (użyty dla 4 grup 4 bitowych) :

- Przeniesienie z grupy $C_g=G_g+P_g\ C_{g-1}$

Czasy: t_p – czas przejścia sygnału przez jeden poziomu bramkowania (PB)

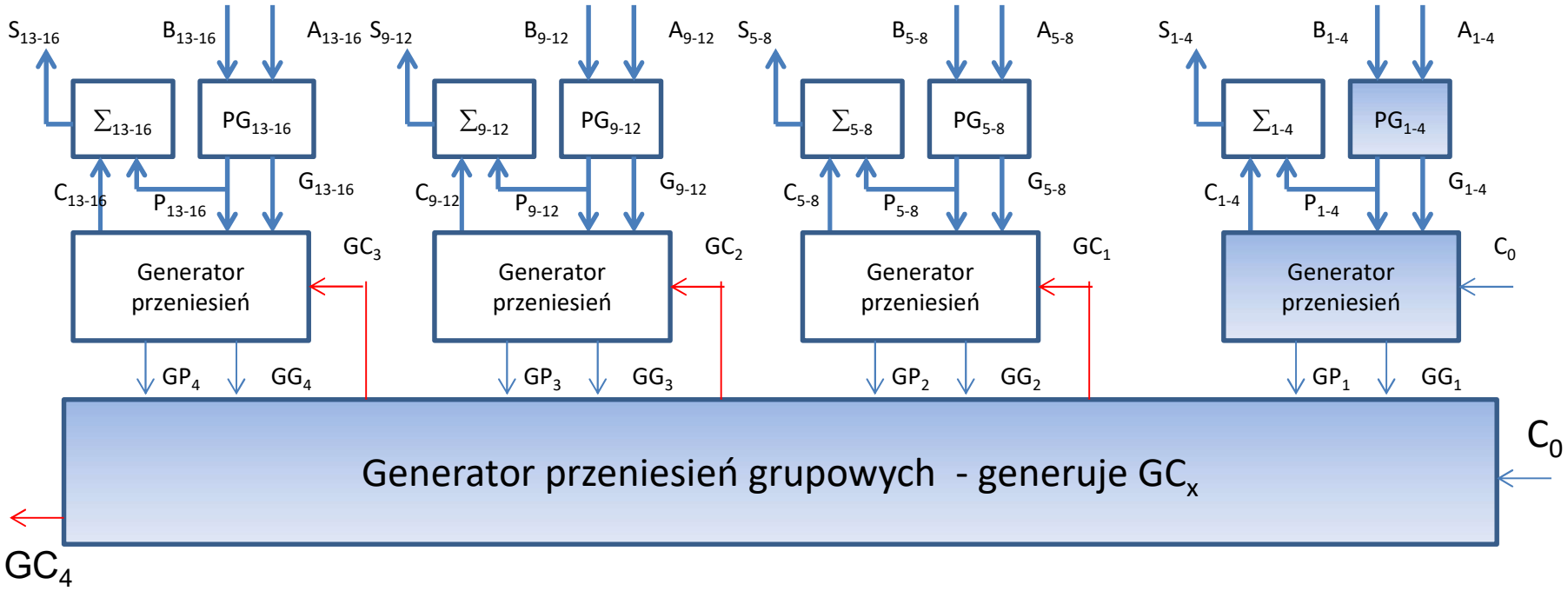
1. Generacja sygnałów G i P (dla grup) jest realizowana równolegle dla każdej z grup - 3 t_p
2. Wyznaczenie sygnału przeniesienia grupy to dodatkowo 2 t_p na grupę (grupowy generator przeniesienia).
3. Powtórzenie kroku 2 dla kolejnych 3 grup.

Czas propagacji dla liczby 16 bitowej:

- Sumator z przeniesieniami propagowanymi równolegle $3t_p + 4*2t_p = 11\ t_p$ (dla n bitów n/2+3)
- Sumator z przeniesieniami propagowanymi szeregowo $16 * 2t_p = 32\ t_p$

Wartości zaznaczone na zielono wyznaczane są równolegle (bazują na tych samych sygnałach)

Grupowy sumator z przeniesieniami równoległymi



- Kolorem niebieskim oznaczono ścieżkę krytyczną w układzie.
- Blok $\sum$ sprowadza się do 1 poziomu bramek EXOR, a generatory przeniesień i generator przeniesień grupowych pracują wg podanych wcześniej wzorów funkcji.

Kombinacyjny układ mnożący

- 2 liczby N bitowe
- Wymagane N*(N-1) sumatorów 1 bitowych

Algorytm mnożenia - przykład

				1	**1**	**0**	**1**	**A**
			*	1	1	1	1	B
				1	1	0	1	A*b0
			1	1	0	1		A*b1
			1	0	1	1	1	Suma bitów
			1	0	0			Przeniesienie
		1	1	0	1			A*b2
		1	1	0	0			Suma bitów
		1	0	1				Przeniesienie
	1	1	0	1				A*b3
	1	1	1	0				Suma bitów
	1	0	1					Przeniesienie
1	1	0	0	0	0	1	1	Suma liczb

Mnożenie, sumowanie, sumowanie bitów + sumowanie liczb

Algorytm mnożenia – koncepcja

				a_3	a_2	a_1	a_0	
			X	b_3	b_2	b_1	b_0	
				$a_3\ b_0$	$a_2\ b_0$	$a_1\ b_0$	$a_0\ b_0$	* 1 bit
		+	$a_3\ b_1$	$a_2\ b_1$	$a_1\ b_1$	$a_0\ b_1$		* 2 bit
			$a_3\ b_1$	s_{31}	s_{21}	s_{11}		
			c_{31}	c_{21}	c_{11}			
	+	$a_3\ b_2$	$a_2\ b_2$	$a_1\ b_2$	$a_0\ b_2$			* 3 bit
		$a_3\ b_2$	s_{42}	s_{32}	s_{22}			
		c_{42}	c_{32}	c_{22}				
+	$a_3\ b_3$	$a_2\ b_3$	$a_1\ b_3$	$a_0\ b_3$				* 4 bit
	$a_3\ b_3$	s_{53}	s_{43}	s_{33}				
+	c_{53}	c_{43}	c_{33}					
s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0	

Sumowanie jest operacją rozłączną i może być realizowane z przesunięciem czasowym, zatem przeniesienie może być uwzględniane nie koniecznie na bieżącym, lecz w kolejnych etapach sumowania – oprócz ostatniego etapu sumowania.

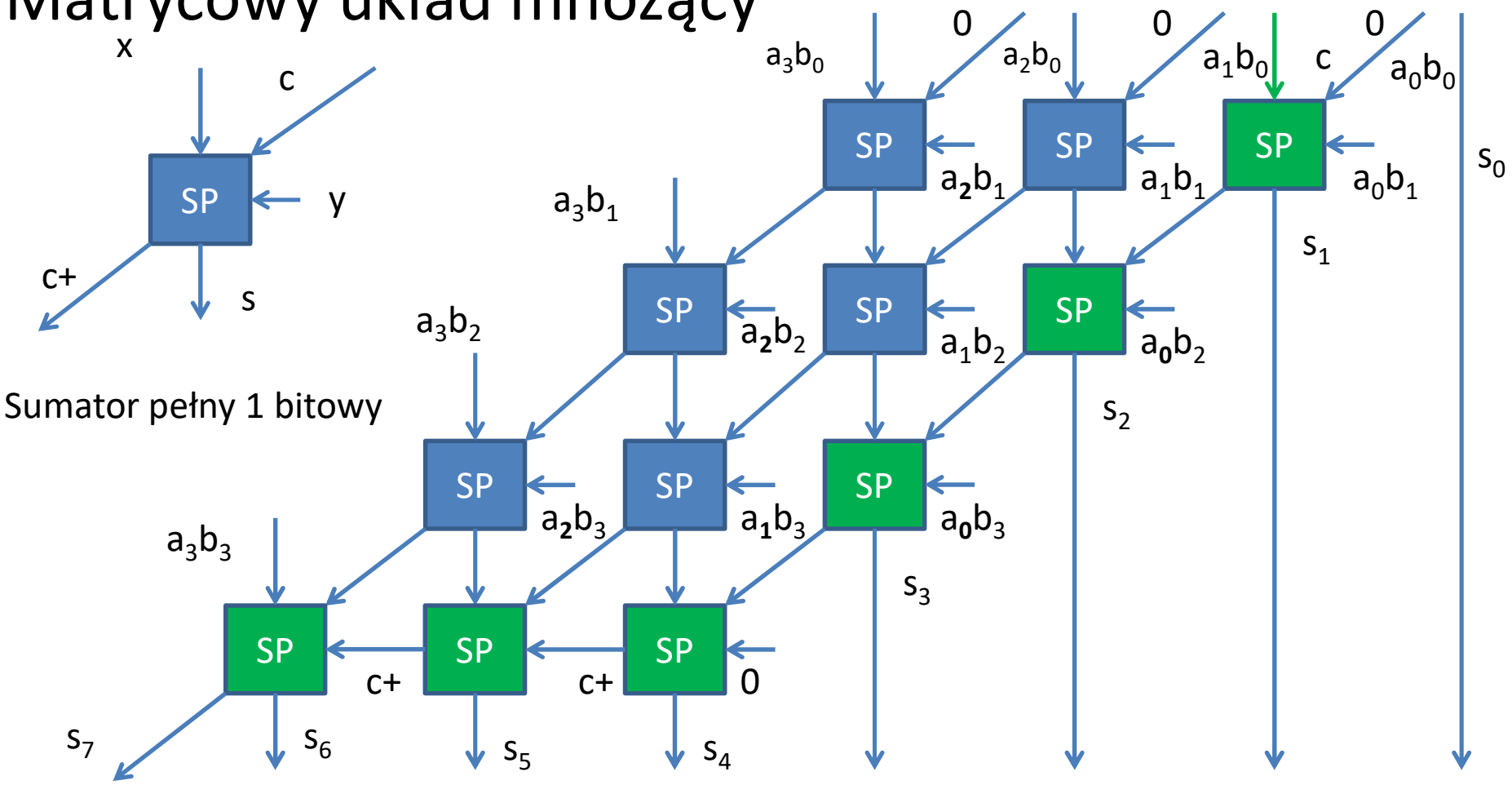
Algorytm mnożenia - przebieg

				a_3	a_2	a_1	a_0
			X	b_3	b_2	b_1	b_0
				$a_3\ b_0$	$a_2\ b_0$	$a_1\ b_0$	$a_0\ b_0$
		+	$a_3\ b_1$	$a_2\ b_1$	$a_1\ b_1$	$a_0\ b_1$	
			$a_3\ b_1$	s_{31}	s_{21}	s_{11}	
			c_{31}	c_{21}	c_{11}		
	+	$a_3\ b_2$	$a_2\ b_2$	$a_1\ b_2$	$a_0\ b_2$		
		$a_3\ b_2$	s_{42}	s_{32}	s_{22}		
		c_{42}	c_{32}	c_{22}			
+	$a_3\ b_3$	$a_2\ b_3$	$a_1\ b_3$	$a_0\ b_3$			
	$a_3\ b_3$	s_{53}	s_{43}	s_{33}			
+	c_{53}	c_{43}	c_{33}				
s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0

Iloczyn czynnika i najmłodszego bitu mnożnej T_{PAND}
Iloczyn czynnika i 2 bitu
Wynik częściowy (bez przeniesienia) $+T_{PSUM}$
Dodanie przeniesienia
Iloczyn czynnika i 3 bitu
Wynik częściowy (bez przeniesienia) $+T_{PSUM}$
Dodanie przeniesienia
Iloczyn czynnika i 4 bitu
Wynik częściowy (bez przeniesienia) $+T_{PSUM}$
SUMAWANIE Z PROPAGACJĄ PRZENIESIENIA
Wynik ostateczny $+ 3T_{PSUM}$

Dla liczby 4 bitowej 4 etapy sumowania, 3 etapy sumowania (na 3 bitach) bez propagacji przeniesienia, przeniesienie dodawane jako składnik sumy kolejnego etapu, 4 etap – sumowanie z propagacją przeniesienia. Czas wyznaczania iloczynu liczb N bitowych $T_{PAND} + (N-1+N-1) * T_{PSUM}$

Matrycowy układ mnożący



T_{p_sp} – propagacja sumatora pełnego

T_{p_b} – propagacja bramki

Czas propagacji układu jest równy T_{p_b} +6* T_{p_sp}

Na zielono oznaczono elementy na ścieżce krytycznej czasu propagacji

Sekwencyjny układ mnożący

2 liczby N bitowe

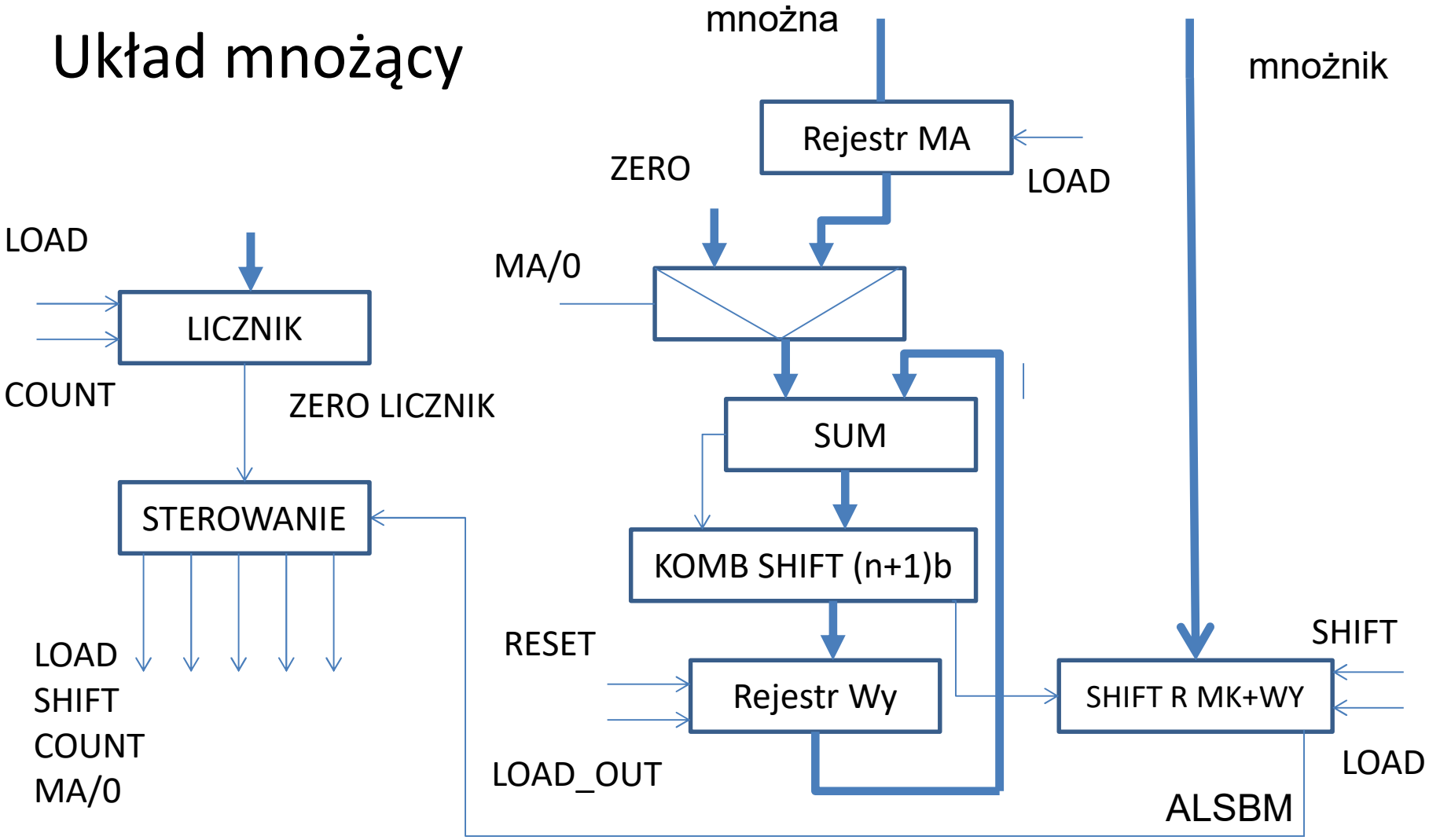
N sumatorów 1 bitowych

Mnożenie liczb w NKB (bez znaku)

1101	Mnożna	13*13=169
1101	Mnożnik	
1101		wynik = 13 – mnożna razy pierwszy bit mnożnika =1
01101		przesunięcie
01101		wynik bez zmian – mnożna razy drugi bit mnożnika =0
001101		przesunięcie
1101		dodanie mnożnej razy 3 bit mnożnika =1 (+52)
1000001		wynik 13* 5= 65
1000001		przesunięcie
1101		dodanie mnożnej razy 4 bit mnożnika =1 (+124)
10101001		wynik 13*13=169

Wynik składa się z części modyfikowanej i stałej. Część stała to bity najmłodsze po jednym dla każdego kroku sumowania. Część modyfikowalna to bity starsze. Do części modyfikowalnej (starsze bity wyniku dotychczasowych sumowań) dodajemy mnożną, wartość dodanych bitów jest zależna od tego na jakim etapie (na których pozycjach) zostaną dodane – każdy kolejny krok to dodanie liczby 2x większej (w przykładzie (13, 52,104)

Układ mnożący



Sygnał SHIFT=LOAD OUT – ŁADUJE NOWĄ WARTOŚĆ – PRZESUNIĘTY (AWYNIK +MA LUB AWYNIK +0)
NA POCZĄTKU LICZNIK ŁADUJE wartość N-1
REJESTR SHIFT R ZAWIERA MŁODSZE BITY WYNIKU I STARSZE BITY MNOŻNIKA
ALSBM AKTUALNY NAJMNIEJ ZNACZĄCY BIT MNOŻNIKA

Sekwencyjny układ mnożący - uwagi

- Licznik kroków – liczba bitów-1
- Sterowanie na podstawie kolejnego bitu mnożnika podejmuje decyzje o dodaniu mnożnej lub 0 (początkowa wartość wyniku równa zero aby X*0=0).
- Zapis wyniku dodawania powoduje przesunięcie wyniku o jeden bit – brak konieczności użycia rejestru przesuwnego i kroku przesuwania dla generacji powiększonego dwukrotnie wyniku sumowania.
- Sterowanie realizuje zadania: zerowanie układu, sprawdzanie warunku stopu, sterowanie multiplekserem, sterowanie zapisem, sygnalizacja końca.
- Sterowanie układem wykonawczym omawiano na ćwiczeniach z PTC