

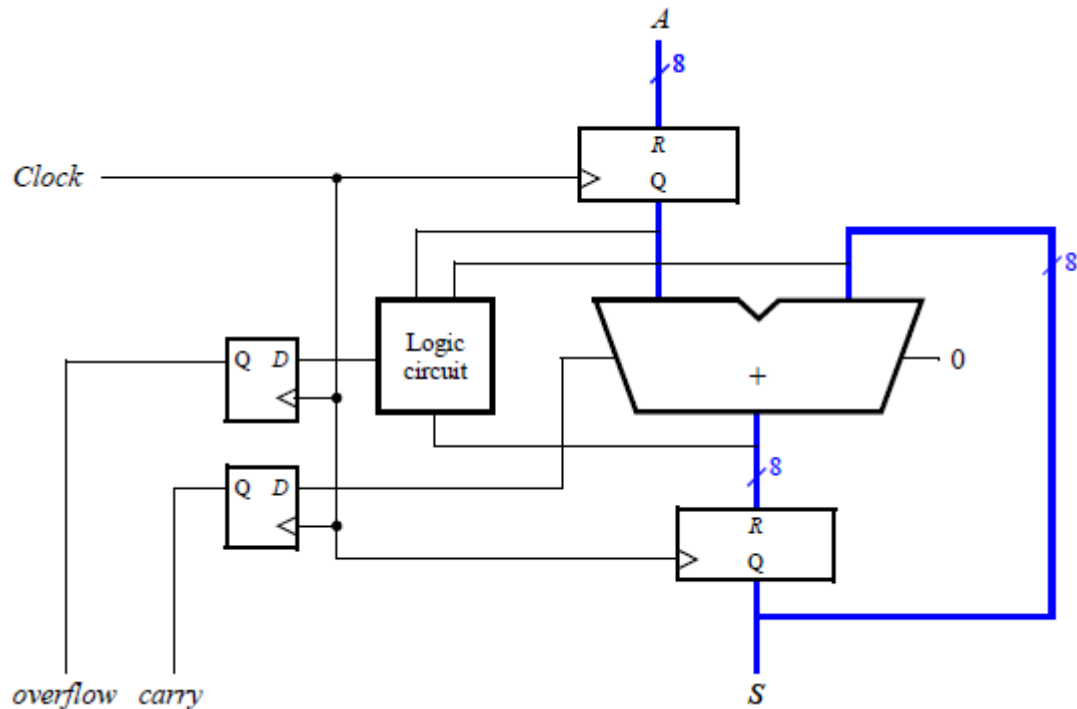
Lab. 7. Sumatory, ALU i multiplikatory

Wymagana wiedza

podstawy budowy sumatorów, ALU i multiplikatorów.

Wykonanie

1. Utwórz projekt accum_N_bits akumulatora N-bitowego, którego struktura jest pokazana na Rys. 1.

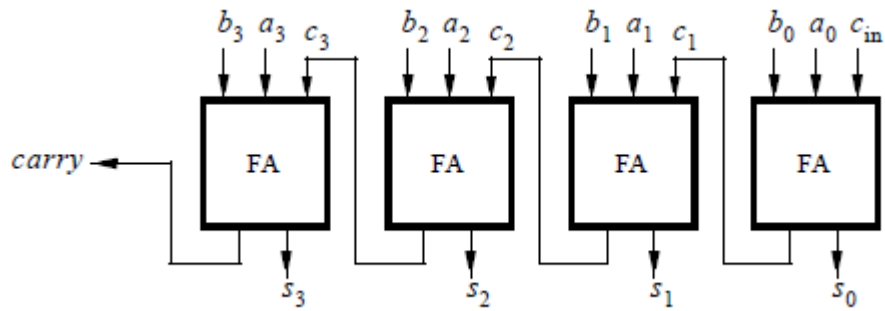


Ryc.1. Schemat ośmiobitowego akumulatora

z następującymi wyprowadzeniami:

Wyprowadzenia	Piny
A	SW7-0, HEX3-2
clk	KEY1
aclr	KEY0
S	LEDR7-0, HEX1-0
carry	LEDR8
overflow	LEDR9

Aby zaimplementować moduł sumujący, użyj układu sumatora z przeniesieniami (rys. 2).



Ryc. 2. Czterobitowy sumator z przeniesieniem

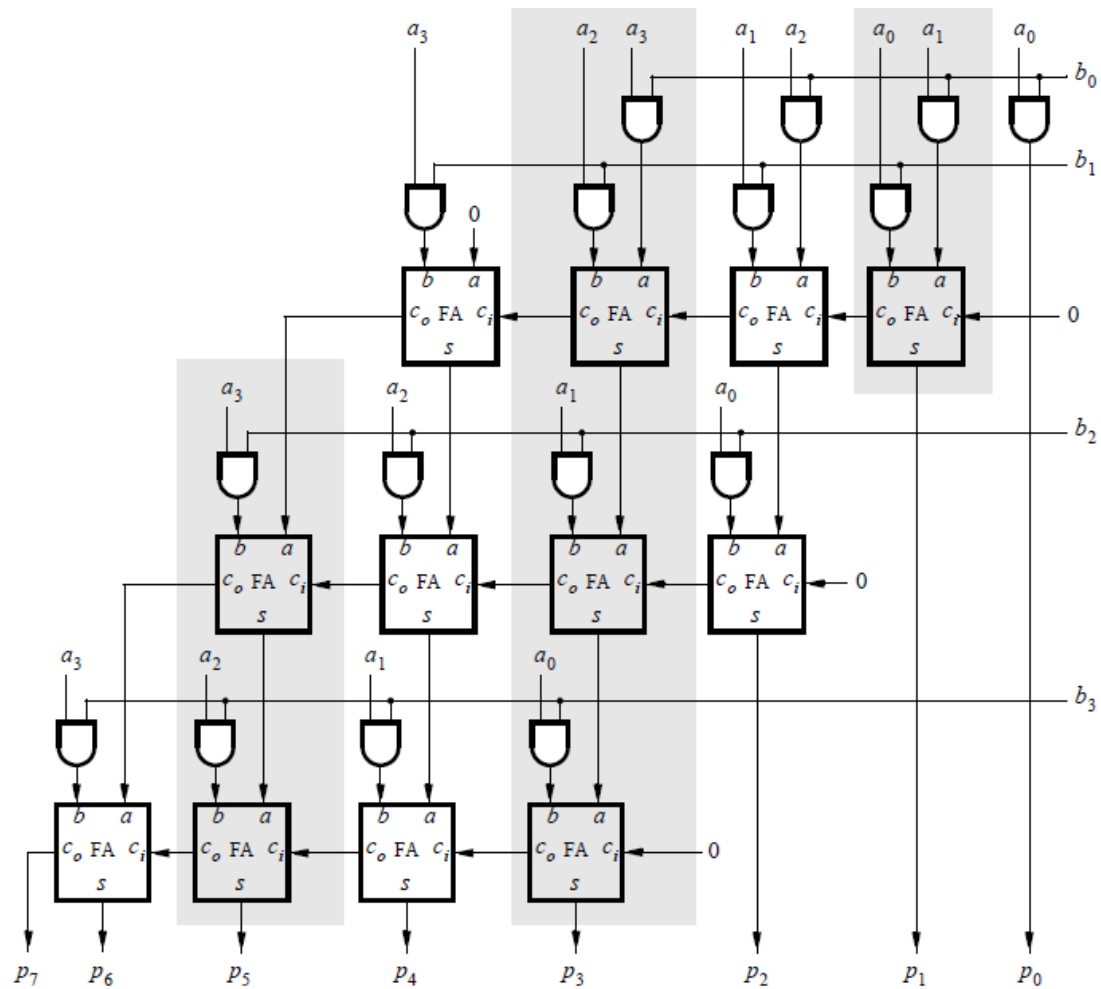
W drugiej wersji projektu użyj operatora przypisania

```
wire [N-1:0] A, B, sum;
assign {carry, sum} = A + B;
```

Porównaj dwa projekty pod kątem kosztów realizacji i prędkości, a także wyniki ich syntezy w **RTL Viewer** i w **Technology Map Viewer**.

2. Zmodyfikuj projekt z pierwszej części, aby można było dodawać i odejmować liczby. Aby to zrobić, dodaj wejście `add_sub` do projektu. Przy `add_sub = 0`, układ musi dodawać liczby do wartości sumy `S`, przy `add_sub = 1`, układ musi odjąć liczby od wartości sumy `S`. Przetestować układ. Sprawdź, czy sygnały **carry** i **overflow** są ustawiane podczas dodawania i odejmowania liczb.

3. Utwórz projekt `array_multiplier_4_bits` mnożnika macierzowego 4-bitowego o strukturze jak na rys 3.



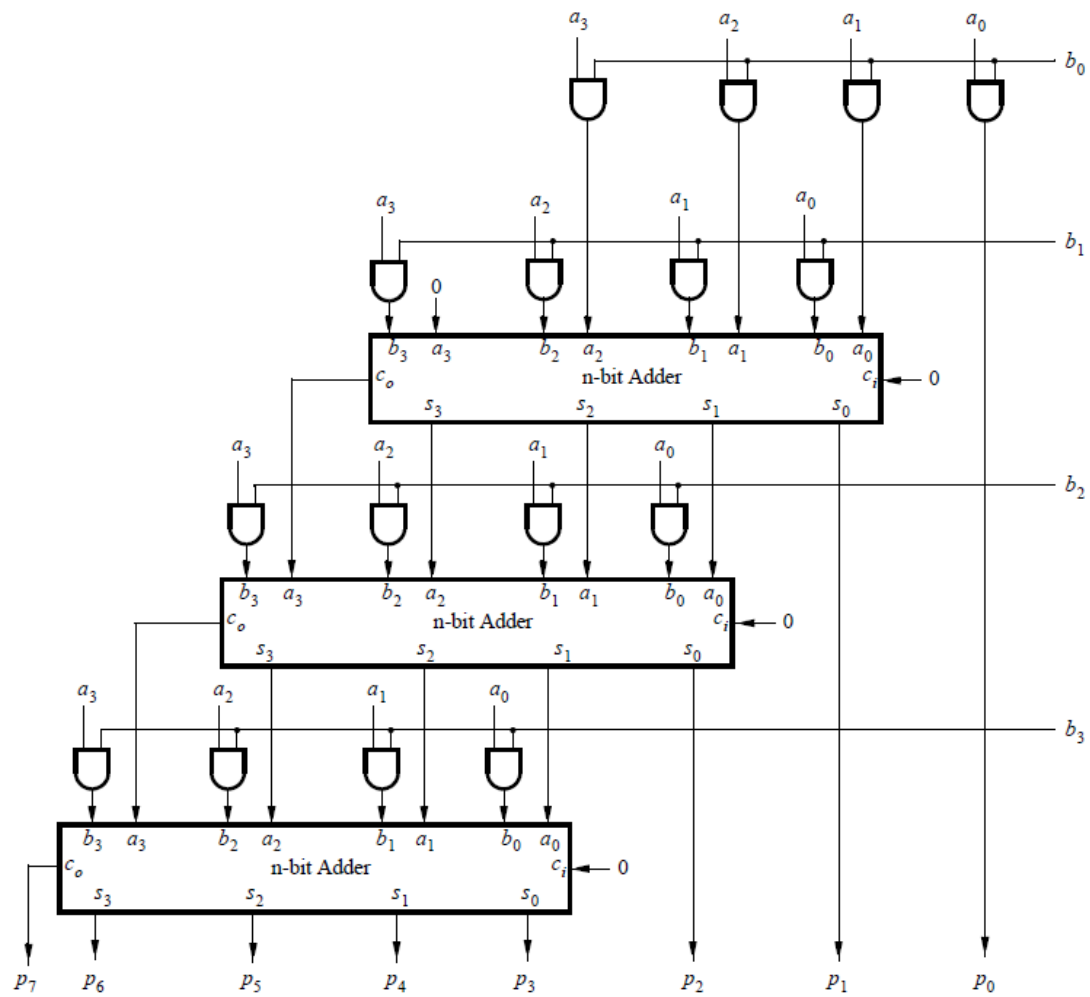
Rys. 3. Schemat mnożnika macierzowego

z następującymi wyprowadzeniami:

Wyprowadzenia	Piny
A[3:0]	SW7-4, HEX2
B[3:0]	SW3-0, HEX0
P[7:0]	HEX5-4

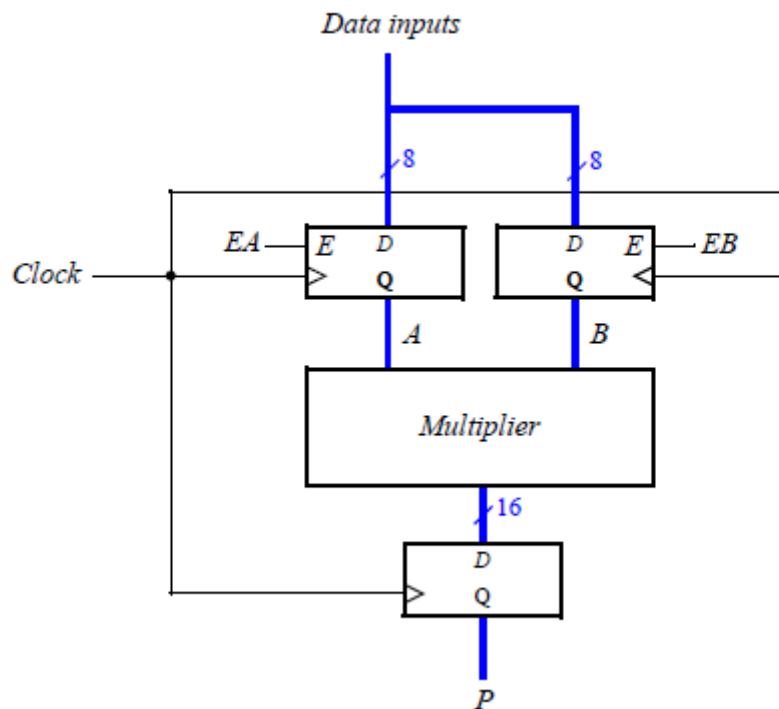
Wykonaj symulację funkcjonalną obwodu, a następnie przetestuj ją na płycie.

4. Utwórz projekt multiplier_N_bits mnożnika N-bitowego o strukturze jak na rys. 4.



Ryc. 4. Realizacja mnożnika macierzowego za pomocą n-bitowych sumatorów

Tutaj, sumatory N-bitowe są używane do tworzenia częściowych produktów. Wynik jest tworzony przez dodanie przesuniętych wartości częściowych produktów. Zastosuj 8-bitowy układ mnożnika N-bitowego o strukturze jak na rys. 5.



Rys. 5. Schemat mnożnika z buforem

z następującymi wyprowadzeniami:

Wyprowadzenia	Piny
Data[7:0]	SW7-0
clk	KEY1
aclr	KEY0
EA	SW9
EB	SW8
P[15:0]	HEX3-0
A	LEDR[7:0] при SW9=1
B	LEDR[7:0] при SW8=1

Wykonaj symulację funkcjonalną obwodu, a następnie przetestuj ją na płycie.

Wskazówka: Aby zrealizować projekt, utwórz następujące moduły:

register_N_bits_ena_aclr - rejestr dla N bitów z włączoną synchronizacją (ena) i asynchronicznym resetem (aclr)

adder_N_bits - sumator dla N bitów z wejściem przeniesienia (cin) i wyjściem przeniesienia (cout)

multiplier_N_bits # (parametr $N = 8$)

(input [N-1: 0] A, B, output [2 * N: 0] P); - Mnożnik N-bitowy, który implementuje strukturę z rys. 4.

W kodzie mnożnika użyj bloków generujących (generate), aby wygenerować częściowe produkty, częściowe sumy i wynik.

5. Użyj innej metody mnożenia 8-bitowych liczb binarnych (wybierz z literatury). Porównaj dwie metody względem kosztu realizacji i prędkości.