

Sumator

Projekt i symulacja sumatora

Cel ćwiczenia

Ćwiczenie ma na celu zapoznanie studenta z hierarchiczną strukturą projektu. Zastosowano podejście “bottom up” dla projektu 4-bitowego sumatora. W pierwszym etapie projektowany jest jednobitowy sumator na podstawie schematu tranzystorowego. Następnie tworzony jest jego symbol, który posłuży do projektu sumatora 4-bitowego (Ripple Carry Adder). Na obu etapach poprawność układu weryfikowana jest za pomocą symulatora ELDO.

Sumator może zostać zaimplementowany według różnych strategii. Funkcja logiczna opisująca sumator jest postaci:

C-Carry in

$$\begin{aligned} \text{(SOP)} \quad \text{SUM} &= ABC + AB'C' + A'B'C + A'BC' & (A' = \text{not } A) \\ \text{CARRY} &= AB + AC + BC = AB + C(A + B) \end{aligned}$$

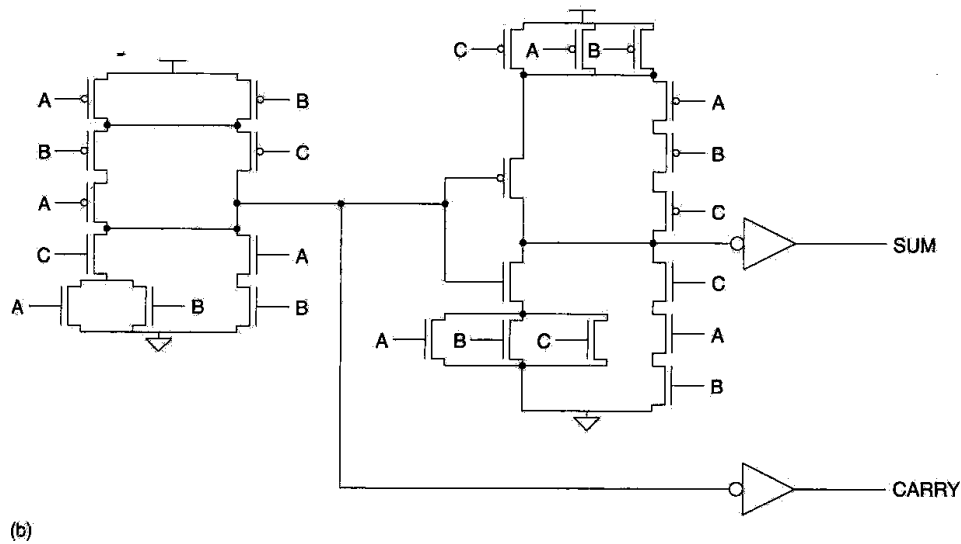
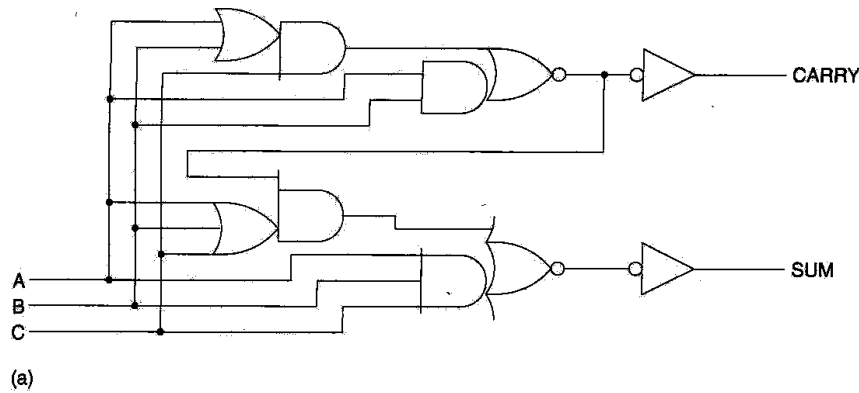
Postać SOP (Sum-Of-Product) może zostać zaimplementowana bezpośrednio z użyciem bramek NAND dla iloczynów oraz bramki OR i dodatkowego inwertera. Realizacja taka nie jest jednak optymalna ze względu na efektywność i prostotę rozwiązania. Jednym z problemów jest 4 wejściowy NAND, który dla poprawnego działania wymaga dużych rozmiarów tranzystorów. Problem ten można łatwo ominąć rozdzielając iloczyny 4 czynnikowe na 2 czynnikowe, co pozwala na użycie 2 wejściowych bramek NAND. Realizowaną funkcję można więc zapisać w postaci:

$$\text{SUM} = C(AB + A'B') + C'(AB' + A'B) = A \text{ xor } B \text{ xor } C$$

3 wejściowa bramka XOR może być wykorzystana do zaimplementowania funkcji SUM. Uwzględniając dodatkowe inwertery całość implementacji zawiera 32 tranzystory. Kolejną modyfikację wyrażenia, polegającą na wykorzystaniu funkcji CARRY do obliczania SUM przedstawiono poniżej:

$$\text{SUM} = ABC + (A + B + C)\text{CARRY}' \quad (\text{CARRY}' = \text{not CARRY})$$

Implementacja sprzętowa tego wyrażenia pozwala zmniejszyć liczbę tranzystorów do 28 co pokazano na poniższym rysunku.



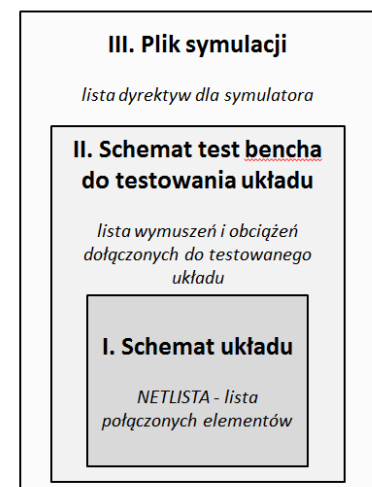
W przypadku optymalizacji układu za względu na szybkość działania korzystniejsze jest wykorzystywanie bramek o mniejszej ilości wejść. Gdy sumator jest wykorzystany do budowy struktury x-bit ripple-carry adder, krytyczną ścieżką jest opóźnienie sygnału CARRY. Istnieje wiele metod pozwalających na minimalizację tego opóźnienia. W ćwiczeniu zajmiemy się wyznaczeniem wielkości tego opóźnienia dla proponowanej realizacji.

Przebieg ćwiczenia

W tym ćwiczeniu wykorzystamy strukturę projektu, którą opanowaliśmy podczas laboratorium drugiego. Ma ona postać taką, jak na rysunku obok. Na początek zdefiniujemy schemat sumatora 1-bit. Schemat ten stanowi netlistę, dlatego zostanie on umieszczony w pliku pierwszego poziomu tj. *I. Schemat układu*.

📌 **I. Posłuchaj wskazówek prowadzącego związanych z definiowaniem schematu, w szczególności jego interfejsu oraz napięć polaryzacji.**

1. Do zdefiniowania podukładu użyj poniższego fragmentu kodu:



```

1  * Sumator netlist
2  .subckt cell_name interface
3
4  ***** Carry unit
5  *PMOS array
6  M1 ...
7  M2 ...
8
9  *NMOS array
10 M6 ...
11 M7 ...
12
13 *INV
14 M11 ...
15 M12 ...
16
17 ***** Sum unit
18 *PMOS array
19 M13 ...
20 M14 ...
21
22 *NMOS array
23 M20 ...
24 M21 ...
25
26 *INV
27 M27 ...
28 M28 ...
29
30 .ends

```

W projekcie wykorzystaj tranzystory NMOS i PMOS o parametrach z poniższej tabeli.

Parametry tranzystorów		
Property name	NMOS	PMOS
ASIM_MODEL	NCH	PCH
L	65 nm	65 nm
Subtype	N	P
W	200 nm	940

2. Po narysowaniu schematu zdefiniuj w pliku drugiego poziomu test bench, w którym będziesz mógł przetestować Twój sumator według koncepcji opracowanej w przygotowaniu do ćwiczenia. Dołącz w test benchu plik z netlistą sumatora:

```

1  .INCLUDE /home/centos/.../sumator/sumator.spi

```

Osadzenia sumatora dokonaj za pomocą poniższej dyrektywy:

```

1  xSUMATOR1 interface SUMATOR

```

Odpowiednio dołącz potrzebne źródła typu PULSE (wymuszenia) i zasilanie DC. Pamiętaj o odpowiednim podłączeniu zasilania (VDD, GROUND). Do wyjść sumatora dołącz

kondensatory o wartościach 500 fF.

🎓 II. Posłuchaj wskazówek prowadzącego na temat metody testowania układu sumatora i definiowania test bencha.

3. Sprawdź poprawność działania sumatora wykonując symulacje zaproponowane w pkt 2 przygotowania do ćwiczenia. Wykorzystaj funkcję PULSE do zrealizowania wymaganych wymuszeń. Po uzyskaniu wyników na monitorze poproś prowadzącego o potwierdzenie w tabeli 1. Na podstawie uzyskanych wykresów uzupełnij tabelę 2.

Dalsza część instrukcji poświęcona jest badaniu sumatora 4-bit.

🎓 III. Posłuchaj wskazówek prowadzącego na temat sposobu budowy i testowania sumatora 4-bitowego.

Architektura sumatora 4-bitowego tj. sposób połączenia podukładów sumatorów 1-bitowych może być zdefiniowana w pliku test bencha lub osobnym pliku .spi.

4. W celu wyznaczenia ścieżki krytycznej (najdłuższego opóźnienia) wykonaj odpowiednie symulacje analogowe zgodnie z propozycjami z pkt 3. przygotowania do ćwiczenia. Kombinację sygnałów wejściowych, dla której testować będziesz sumator, zapisz w tabeli 3.

Dołączenie do wejścia logicznego „0” w symulatorze analogowym jest równoznaczne z podłączeniem tego wejścia do masy (GROUND); logiczne „1” oznacza dołączenie napięcia zasilania (VDD). Można również wykorzystać wymuszenia typu PWL lub PULSE. Odczytane z wykresów wyniki wpisz do tabeli 4 w karcie ćwiczenia. Poproś prowadzącego o potwierdzenie poprawności wyników uzyskanych na monitorze w karcie ćwiczeń w tabeli 4.

5. Wykonaj symulacje pozwalające wyznaczyć parametry wymienione w tabeli 5. Otrzymane wyniki wpisz do tabeli.

Karta Ćwiczenia 3. Projekt i symulacja sumatora

Imię i Nazwisko: Grupa Semestr

Dzień tygodnia i godzina:

Data wykonania ćwiczenia: Podpis prowadzącego:

I. Przygotowanie do ćwiczenia

Wykonanie „przygotowania do ćwiczenia” jest warunkiem koniecznym dopuszczenia do ćwiczenia i powinno być przygotowane w domu.

Do niniejszej kartki dołącz (zszyte zszywaczem) pisane odręcznie (zabronione jest pisanie na komputerze) odpowiedzi na następujące zagadnienia:

1. Narysuj schemat pełnego sumatora (full adder) za pomocą bramek logicznych. Napisz funkcje logiczne dla wyjść SUM i CARRYout.
2. Zaproponuj i narysuj przebiegi sygnałów wejściowych A,B,Cin pozwalające przetestować poprawność działania sumatora. Narysowane przebiegi fali prostokątnej powinny mieć wartości amplitudy 0 i 1.2 V, a czas trwania jednego bitu powinien wynosić 50 ns. Cały przebieg powinien więc zmieścić się w czasie 400 ns±czas narastania i opadania.
Jaką analizę należy wykonać, aby sprawdzić poprawność działania sumatora na poziomie tranzystorów? Podaj parametry analizy, a w szczególności czas jej trwania.
Zaproponuj analizę i przebiegi do wyznaczenia czasów, które należy wpisać w tabeli 2.
3. Narysuj symbol sumatora, a następnie użyj go do narysowania schematu sumatora 4 bitowego typu „Ripple-Carry Adder” (połączenie łańcuchowe sumatorów).
4. Określ ścieżkę krytyczną dla tego typu sumatora 4-bitowego i podaj kombinację wejściową (przebiegi wejściowe), które pozwolą wyznaczyć maksymalny czas propagacji.
5. Zakładając, że najdłuższe opóźnienie na jednym sumatorze (Cin do Cout) wynosi 10 ms i wejściowymi liczbami są $(xxxx)_2$ oraz $(yyyy)_2$ zaproponowane przez Ciebie w pkt 4 podaj wartości (0/1) na wyjściach sumatora w chwilach 9 ms, 19 ms, 29 ms, 39 ms, 49 ms i 59 ms.

Czas	Cout	S3	S2	S1	S0
9 ms					
19 ms					
29 ms					
39 ms					
49 ms					
59 ms					

Zaproponuj rodzaj symulacji i narysuj przebiegi wejściowe dla weryfikacji powyższej tabeli dla sumatora 4-bitowego.

II. Wyniki symulacji

Tabela 1.

Potwierdzenie przez prowadzącego poprawności wykresów na monitorze	podpis prowadzącego
--	---------------------

Tabela 2.

	<i>Parametr</i>	<i>Wartość</i>	<i>Jednostka</i>
Z wejścia A do wyjścia CARRY	<i>Opóźnienie high-to-low t_{df}</i>		
	<i>Opóźnienie low-to-high t_{dr}</i>		
	<i>Opóźnienie t_d</i>		
	<i>Czas opadania t_f</i>		
	<i>Czas narastania t_r</i>		
Z wejścia C do wyjścia CARRY	<i>Opóźnienie t_d</i>		
Z wejścia A do wyjścia SUM	<i>Opóźnienie Delay time t_d</i>		

Tabela 3.

A3A2A1A0 C0	
B3B2B1B0	
CARRY S3S2S1S0	

Tabela 4.

Czas	Cout	S3	S2	S1	S0
Potwierdzenie przez prowadzącego zgodności wpisów z wykresami na monitorze				podpis prowadzącego	

Tabela 5.

	<i>Parametr</i>	<i>Wartość</i>	<i>Jednostka</i>
Z wejścia A0 do wyjścia CARRY	<i>Opóźnienie high-to-low t_{df}</i>		
	<i>Opóźnienie low-to-high t_{dr}</i>		
	<i>Opóźnienie t_d</i>		
	<i>Czas opadania t_f</i>		
	<i>Czas narastania t_r</i>		
Z wejścia C0 do wyjścia CARRY	<i>Opóźnienie t_d</i>		
Z wejścia A0 do wyjścia SUM0	<i>Opóźnienie Delay time t_d</i>		