

Лабораторна робота № 4

Системи числення, моделювання суматорів

Мета роботи: отримання навичок запису чисел в різних системах числення, проектування перетворювачів кодів, комбінаційних суматорів, удосконалення навичок поведінкового опису апаратури на мові Verilog .

Зміст :

Вступ	1
Запис чисел	1
Додавання чисел	3
Суматор із прискореним переносом	4
Завдання до виконання	5
Вимоги до оформлення звіту	6
Література 6 _	6

Вступ

Мова моделювання Verilog HDL дозволяє представляти числа в кількох системах числення . Всередині схем числа обробляються у двійковій формі, однак, ввід та вивід таких схем часто здійснюється за допомогою десяткових чисел. У цій лабораторній роботі розглядаються різні представлення чисел, метод перетворення чисел з однієї форми до іншої.

Запис чисел

У Verilog сигнали можуть приймати значення із 4-значного алфавіту:

- а) 0: логічний нуль, або хибна величина;
- б) 1: логічна одиниця, або істинна величина;
- в) x : невідоме значення;
- г) z : високоімпедансний стан.

Величина z на вході схеми зазвичай інтерпретується як невідома величина x . у загальному випадку, у мові Verilog розрізняються символи нижнього та верхнього регістру, проте для запису чисел можуть рівнозначно використовуватись символи обох реєстрів.

У Verilog визначено три типи констант:

- 1) цілі;
- 2) дійсні;
- 3) рядки.

Символ підкреслювання () можна використовувати для запису цілих та дійсних констант для покращення візуального сприйняття, наприклад, для відділення груп розрядів. Символ підкреслення не може бути першим чи останнім символом константи.

Цілі константи можна записувати у звичній десятковій формі або з явним зазначенням основи числення. Наприклад, у звичній десятковій формі можна записати:

```
15
-32
```

Хоча для запису цих констант вистачило б 6 двійкових розрядів ($+15_{10} = 01_1111_2$, $-32_{10} = 10_0000$), проте використання констант у звичайному десятковому записі призводить до використання 32-бітної апаратури.

Для явного позначення основи числення використовують такий синтаксис:

```
[size] 'baze value
```

де **size** визначає кількість двійкових розрядів, **value** записується послідовністю цифр, допустимих у заданій системі числення, а **baze** власне визначає основу числення за допомогою символів:

- o або O - вісімкова ;
- b або B – двійкова;
- d або D – десяткова;
- h або H – шістнадцяткова .

value має бути беззнаковою величиною. Приклади:

wire [4:0] 5'O37	5-бітовий вісімковий запис
reg [3:0] 4'B1x_01	4-бітовий двійковий
wire [3:0] 4'd-4	помилка, не може бути від'ємною
wire [11:0] 7'Hx	7-бітовий, x розширюється до xxxxxxx

Якщо вказано такий розмір константи, що її діапазон перевищує вказану величину, то старші розряди зліва заповнюються нулями, крім випадків, коли старший зазначений розряд дорівнює z або x , тоді розряди зліва заповнюються відповідно z або x . Якщо вказана величина перевищує діапазон значень, що визначаються розміром, то старші розряди усикаються. Якщо ж розмір константи опущено, то за замовчуванням використовується 32-бітовий запис.

Додавання чисел

При додаванні пари однорозрядних чисел отриманий результат може потребувати для зберігання двох розрядів, наприклад $1_2 + 1_2 = 10_2$, тобто крім суми може виникнути перенос у наступний старший розряд. Якщо багаторозрядний суматор будувати каскадним включенням однорозрядних суматорів, то в кожному розряді слід буде додавати по 3 однорозрядні числа: два доданки та перенос із попереднього розряду. Проте результат такого додавання також можна зберігати за допомогою двох розрядів: $1_2 + 1_2 + 1_2 = 11_2$. Ця операція реалізується за допомогою повного суматора:

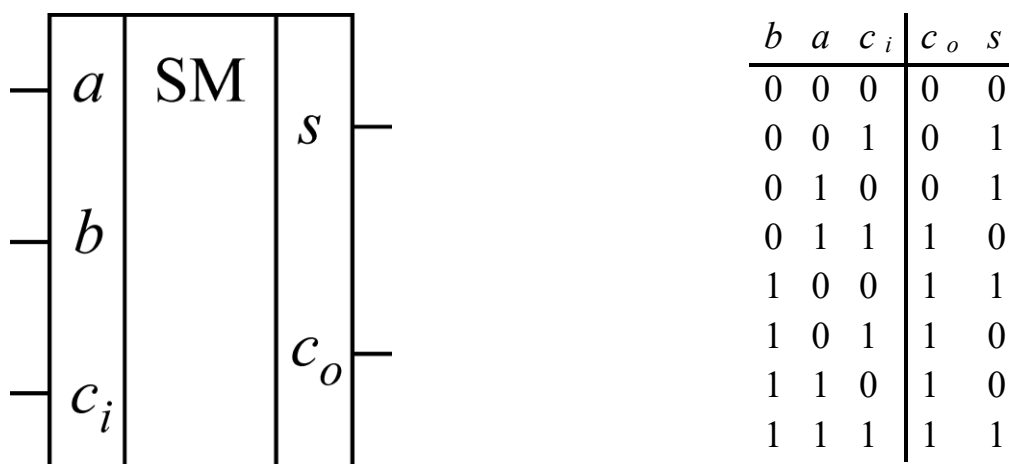


Рис. 1. Символ повного суматора та його таблиця істинності

Найпростіший спосіб створення багаторозрядного суматора — каскадне з'єднання однорозрядних суматорів. При цьому в найгіршому випадку сигнал переносу може поширюватися від молодшого до старшого розряду через кожен

суматор, тому таку схему називають схемою із послідовним переносом (англ. ripple carry).

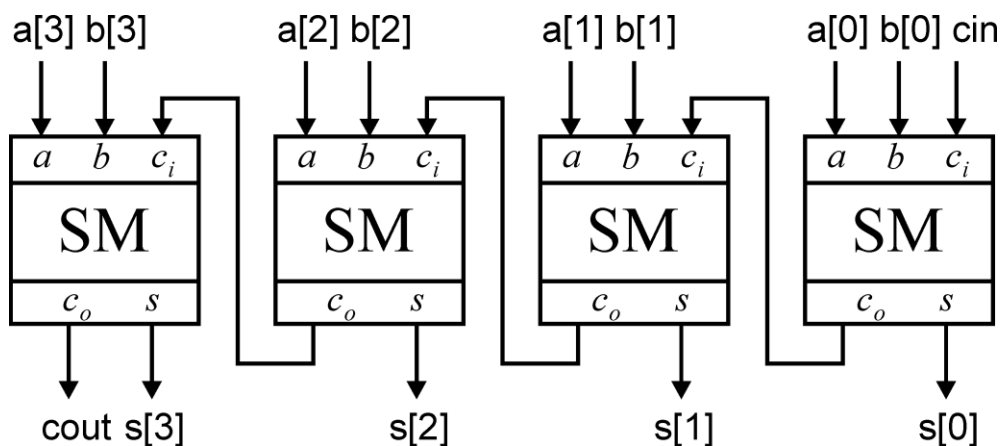


Рис. 0. 2. Чотирирозрядний суматор із послідовним переносом

Суматор із прискореним переносом

Через поширення сигналу переносу через усю схему суматора, додавання багаторозрядних чисел може відбуватися повільно. Для його прискорення використовують суматор із паралельним переносом (англ. carry look ahead). Ця схема працює з використанням сигналів P та G для кожного біта, виконуючи передачу сигналу переносу від попереднього розряду, якщо один із сигналів P або G дорівнює 1, генерацію сигналу переносу, якщо обидва сигнали P та G дорівнюють одиниці, або скидаючи сигнал переносу в нуль, якщо обидва сигнали P та G дорівнюють нулю. На основі сигналів P та G обчислюються сигнали переносу для всіх розрядів:

$$P_i = A_i \oplus B_i;$$

$$G_i = A_i B_i;$$

$$C_{i+1} = G_i + P_i C_i.$$

Прискорення обчислення відбувається за рахунок того, що всі сигнали переносів обчислюються одночасно, паралельно, незалежно від положення розряду.

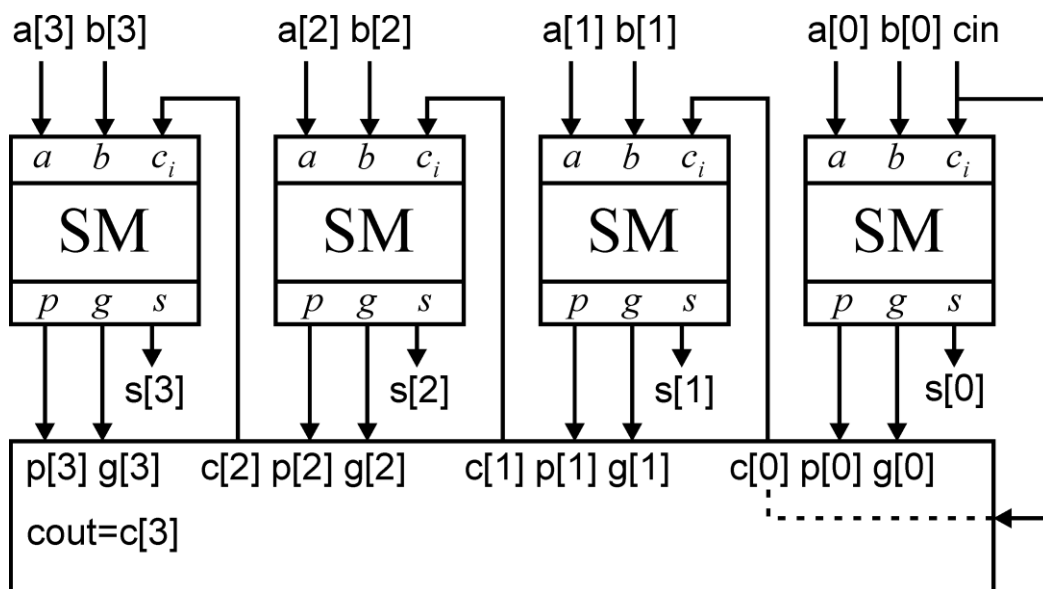


Рис. 0. 1. Суматор із прискоренням (паралельним переносом)

Завдання до виконання

1. [Макет] Побудувати дешифратор 7-сегментного індикатора та перевірити його роботу на симуляторі та на макеті.
2. Побудувати модуль повного суматора, використовуючи модель потоків даних (dataflow). Входами модуля є однорозрядні додатки a , b та однорозрядний сигнал перенесення із попереднього розряду cin .
3. Перевірити роботу повного суматора на симуляторі.
4. Побудувати модуль 4 розрядного суматора з послідовним перенесенням. За основу взяти модулі з попереднього завдання.
5. Перевірити роботу 4-розрядного суматора на симуляторі.
6. [Макет] Перевірити роботу 4-розрядного суматора на макеті. Для цього сигнали сформувати за допомогою:
 - a. a – перемикачів SW7-SW4,
 - b. b – перемикачів SW3-SW0, cout to LED7
 - c. cin - кнопки BTNU ,
 - d. s – вивести на світлодіоди LED 3- LED 0 ,
 - e. $cout$ - вивести на LED7 .

7. *Побудувати двійково-десятковий суматор. Інтерпретувати вхідні аргументи як задані у кодї 8-4-2-1. Перевірити роботу суматора на симуляторі.
8. [Макет] Перевірити роботу суматора на макеті. Сигнал суми вивести на правий семисегментний індикатор за допомогою раніше розробленого дешифратора.

Вимоги до оформлення звіту

1. Протокол оформлюється кожним студентом окремо.
2. Протокол має містити:
 - a. Титульний лист.
 - b. Завдання за варіантом .
 - c. Лістинги модуля та перевірного стенду до нього.
 - d. Результат та моделювання – вивід у консоль та часові діаграми сигналів.
 - e. Висновки по роботі.
3. Захист робіт і проводиться кожним студентом особисто.

Література

1. Дж.Ф. Вейкерлі. Проектування цифрових пристроїв.
2. В.І. Зубчук, В.П. Сігорський, О.М. Шкуро. Довідник з цифрової схемотехніки.