

Лабораторна робота №2

Синхронні лічильники

2.1 Практичне застосування досліджуваних схем	1
2.2 Теоретична частина	1
2.2.1 Умовне позначення і логіка роботи синхронного лічильника	1
2.2.2 Синхронний Т-тригер	3
2.2.3 Схема і принцип роботи синхронного лічильника	5
2.2.4 Опис синхронного лічильника на мові Verilog	7
2.2.5 Опис синхронного лічильника з входом завантаження на мові Verilog	9
2.2.6 Цифро-аналоговий перетворювач (ЦАП) на резисторах	10
2.2.7 Генератор пилкоподібної напруги на базі лічильника і ЦАП	13
2.2.8 Опис ROM пам'яті на мові Verilog	13
2.2.9 Генератор синусоїдального сигналу	13
2.2.10 Генератор синусоїдального сигналу з керованою частотою (NCO)	14
2.2.11 Тестбенч на мові Verilog (initial процеси, цикли, затримки)	14
2.3 Практична частина	14
2.3.1 Симуляція проектів в ModelSim	14
2.3.2 Налаштування синтезу пам'яті в Quartus Prime	14
2.3.3 Цифро-аналоговий перетворювач у налагоджувальних платах з VGA	14
2.3.4 Завдання на лабораторну роботу	14
2.4 Контрольні запитання	15
2.5 Перелік посилань	15

2.1 Практичне застосування досліджуваних схем

Синхронні лічильники імпульсів широко використовують для ділення частоти на довільне значення, зокрема в таймерах. Будь який таймер побудований на базі лічильника. В мікроконтролерах таймери використовують для формування часових інтервалів. Наприклад, у перериванні таймеру з частотою 1 КГц може виконуватись перемикання процесів операційної системи. Або по перериванню від таймеру з певною періодичністю процесор може виходити з режиму сну (зі зниженим енергоспоживанням), зчитувати дані від сенсорів і передавати їх в мережу, а потім знову переходити в режим сну до наступного переривання від таймеру.

Загалом ділення частоти одна з найактуальніших задач, оскільки в цифрових мікросхемах часто необхідно виконувати різні операції з різною періодичністю. Цього можна досягнувши діленням опорної вхідної частоти на різні значення і виконуючи операції з прив'язкою до одержаних частот.

Лічильники імпульсів використовують для широтно-імпульсної модуляції, яку ми розглянемо в одній з наступних лабораторних робіт. Також синхронний лічильник є основним компонентом генератора імпульсів з цифровим керуванням (Numerically Controlled Oscillator, NCO), за допомогою якого можна генерувати аналогові сигнали, або реалізувати частотну модуляцію, зокрема для радіо передачі цифрових даних. Створення аналогових сигналів з використанням NCO називається Прямим Цифровим Синтезом (Direct Digital Synthesis, DDS) і застосовується в усіх сучасних генераторах сигналів.

2.2 Теоретична частина

2.2.1 Умовне позначення і логіка роботи синхронного лічильника

Умовне графічне позначення синхронного лічильника наведено на рис.2.1.

Як видно з рис.2.1. типовий синхронний лічильник має однорозрядні входи синхронізації та асинхронного скидання (в даному випадку це входи `i_clk` та `i_rst_n`) і багаторозрядний вихід, що відображає вміст лічильника (в даному випадку це вихід `o_counter`).

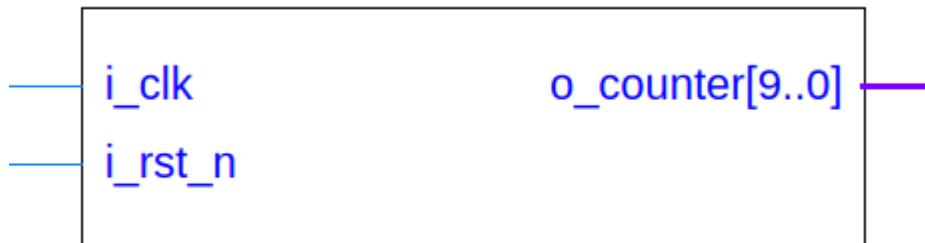


Рис.2.1 - Умовне графічне позначення синхронного лічильника з входом асинхронного зкидання

Основна функція лічильника імпульсів - рахувати імпульси, що надходять на вхід синхронізації. Для лічильників, що рахують вгору, після кожного активного фронту на вході синхронізації вміст лічильника збільшується на 1. Для лічильників, що рахують вниз, після кожного активного фронту на вході синхронізації вміст лічильника зменшується на 1.

Лічильники вміст яких представляється двійковим числом називають двійковими лічильниками. Двійкові лічильники найбільш розповсюджені в цифровій схемотехніці. Окрім двійкових лічильників є ще лічильники у позиційному коді (one-hot counters) і лічильники у коді Грея (Gray code counters). Далі в цій лабораторній роботі будемо розглядати лише двійкові лічильники. Лічильник у позиційному коді розглянемо в наступній лабораторній роботі.

При подачі активного логічного рівня на вхід асинхронного зкидання, значення лічильника скидається в 0. Активним рівнем асинхронного зкидання у багатьох лічильниках є логічний нуль. Однак будуючи схему лічильника у схемному редакторі, або описуючи таку схему на мові Verilog ви можете за власним бажанням визначити активний логічний рівень входу асинхронного зкидання.

У деяких лічильників також може бути присутній вхід синхронного зкидання. Якщо на вхід синхронного зкидання подано активний логічний рівень, після надходження найближчого активного фронту на тактовий вхід вміст такого лічильника зкинеться в 0.

Лічильник розрядністю N біт може рахувати від 0 до $2^N - 1$. Якщо N -розрядний лічильник, що рахує вгору, містить число $2^N - 1$, наступний імпульс на тактовому вході

приведе до переповнення лічильника (counter overflow) і появи в ньому значення 0. Іншими словами, у такого лічильника буде 2^N стійких станів (від 0 до $2^N - 1$). Кількість станів, які може приймати лічильник називається його модулем рахунку М. Для деяких лічильників можна визначити їх модуль рахунку за допомогою окремого входу. Така можливість часто використовується в керованих подільниках частоти, які розглянемо в наступній лабораторній роботі.

За допомогою лічильника з модулем рахунку М можна створити подільник частоти на М.

У деяких лічильниках присутній вхід завантаження вмісту лічильника, за допомогою якого можна завантажити в лічильник певне значення, з якого буде продовжуватись рахунок.

2.2.2 Синхронний Т-тригер

Перед вивченням схеми синхронного двійкового лічильника необхідно розібратися з синхронним по фронту Т-тригером, умовне графічне позначення якого наведено на рис.2.2.

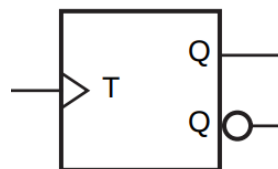


Рис.2.2 - Умовне графічне позначення Т-тригера синхронного по передньому фронту

Найпростіший Т-тригер має вхід синхронізації Т, на який подається сигнал тактової частоти. На виході Q присутнє значення тригера (0 або 1).

По кожному активному фронту на вході синхронізації вміст Т-тригера інвертується - рис.2.3.



Рис.2.3 - Часова діаграма зміни сигналів в Т-тригері синхронному по передньому фронту

Як видно з рис.2.3 найпростіший Т-тригер працює, як подільник частоти на 2, оскільки період імпульсного сигналу на виході Q в два рази більший ніж період сигналу на вході T (відповідно, частота в 2 рази нижча).

Найпростіший Т-тригер можна побудувати на базі синхронного по фронту D-тригера - рис.2.4.

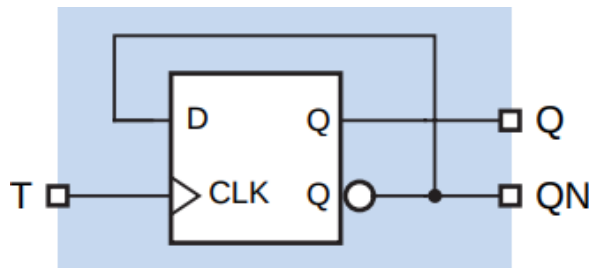


Рис.2.4 - Схема Т-тригера синхронного по передньому фронту

Як видно з рис.2.4 значення інверсного виходу тригера подається на вхід даних D тригера. Якщо тригер зберігає значення $Q=1$, на його інверсному виході QN буде присутній 0 і цей 0 буде подано на вхід даних. Отже по наступному активному фронту цей 0 буде записано в тригер і на виході Q з'явиться 0.

Також існує Т-тригер з входом дозволу роботи EN. Умовне графічне позначення такого тригера наведено на рис.2.5, а часові діаграми роботи - на рис.2.6.

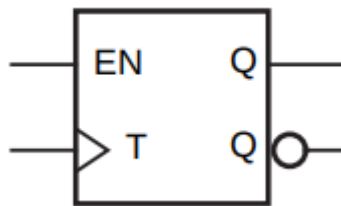


Рис.2.5 - Умовне графічне позначення Т-тригера синхронного по передньому фронту з входом дозволу EN (Enable)

В такому Т-тригері значення на виході Q інвертується по активному фронту сигналу синхронізації на вході T лише за умови, що на вході EN присутній активний логічний рівень. Якщо на вході EN не активний логічний рівень - після надходження активного фронту стан тригера не змінюється. Це добре видно з рис.2.6.

Зазвичай у входа EN активний високий рівень.

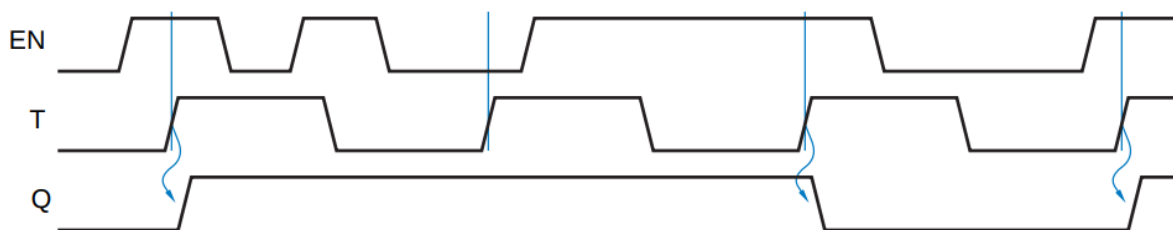


Рис.2.6 - Часова діаграма зміни сигналів в синхронному по передньому фронту Т-тригері з входом дозволу EN (Enable)

Схема Т-тригера синхронного по передньому фронту з входом дозволу EN наведена на рис.2.7.

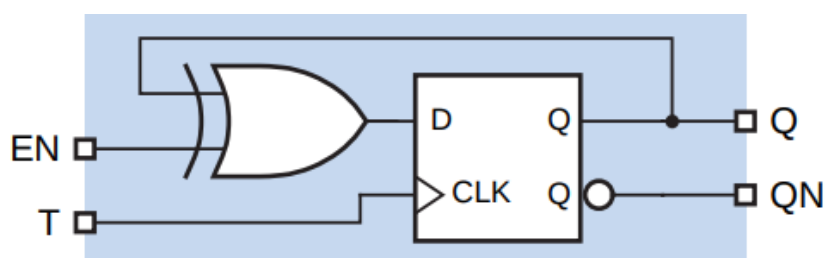


Рис.2.7 - Схема Т-тригера синхронного по передньому фронту з входом дозволу EN

2.2.3 Схема і принцип роботи синхронного лічильника

Розглянемо часову діаграму лічильника, що рахує вгору - рис.8.

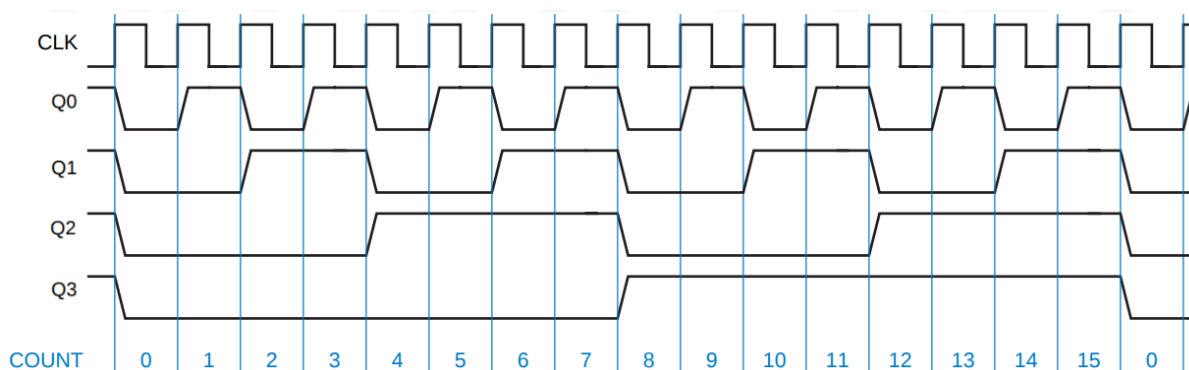


Рис.2.8 - Часова діаграма зміни сигналів в синхронному лічильнику, що рахує вгору

З діаграми на рис.2.8 видно, що кожен наступний розряд інвертується в момент надходження вхідного сигналу синхронізації CLK, за умови, що всі попередні розряди приймають значення одиниць.

Враховуючи наведене вище спостереження не складно побудувати схему синхронного лічильника, що рахує вгору, на базі синхронних по фронту Т-тригерів з входом дозволу - рис.2.9.

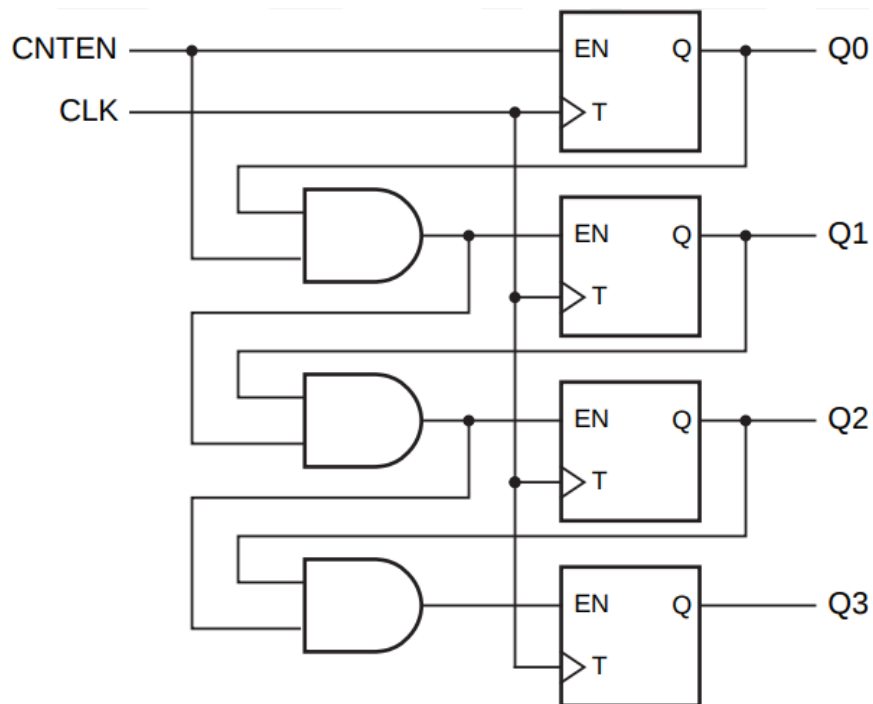


Рис.2.9 - Схема синхронного 4-х розрядного лічильника вгору з послідовним формуванням сигналу EN

Як видно з рис.2.9, на вході EN кожного Т-тригера буде присутня лог. 1 лише за умови, що на виходах Q всіх попередніх розрядів будуть одиниці. Таку схему нескладно масштабувати на більшу розрядність. При цьому, затримка формування сигналу EN для старшого розряду лічильника (в даному випадку Q3) буде дорівнювати сумі затримок 3-х лог. елементів AND.

Можна зменшити затримку формування сигналів EN використовуючи схему з рис.2.10.

Зверніть увагу на вхід CNTEN, що дозволяє рахунок лічильника. У разі, якщо $CNTEN = 0$, на входах EN тригерів присутній 0 і стан тригерів після приходу активного фронту не змінюється. Якщо $CNTEN = 1$, лічильник починає лічити імпульси вхідного сигналу синхронізації.

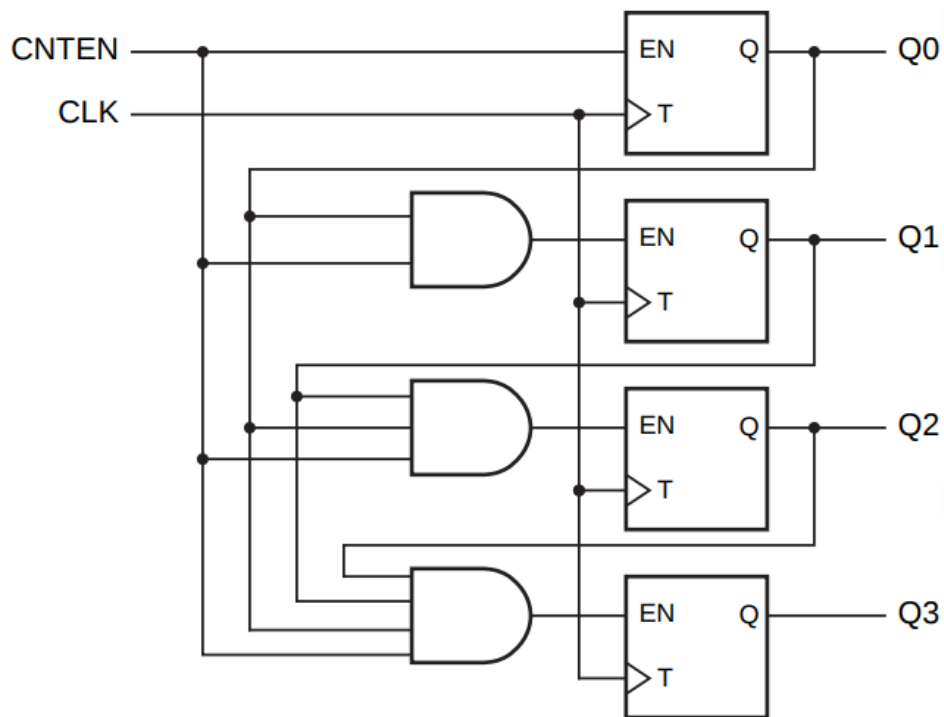


Рис.2.10 - Схема синхронного 4-х розрядного лічильника вгору з паралельним формуванням сигналу EN

Зверніть увагу, що частота імпульсного сигналу з рис.2.8 у кожному наступному розряді вдвічі менша, ніж у попередньому розряді. Саме тому звичайний лічильник можна використовувати для ділення частоти на 2^N .

Для побудови лічильника, що рахує вниз на входи лог. елементів AND необхідно подавати дані не з прямих виходів Q, а з інверсних виходів NQ (або інвертовані значення Q за допомогою інверторів). За необхідності можна створити лічильник з керованим напрямком рахунку (вгору/вниз), якщо обирати між прямим і інвертованим значенням Q за допомогою мультимплексора.

2.2.4 Опис синхронного лічильника на мові Verilog

Шаблон опису на мові Verilog для синхронного лічильника, що рахує вгору наведено в лістингу 2.1.

Як видно з лістингу 2.1. по передньому фронту `i_clk`, якщо сигнал `i_rst_n` не активний, до значення `o_counter` додається 1 і результат суми записується у багаторозрядну змінну `o_counter` типу `reg`. Оскільки така логіка роботи відповідає

лічильнику, САПР синтезує зазначений шаблон коду в лічильник. Розрядність лічильника визначається розрядністю змінної o_counter.

Аналогічним чином можна описати лічильник, що рахує вниз, однак замість оператора “+” необхідно використовувати оператор “-”.

Лістинг 2.1 - Шаблон опису синхронного лічильника вгору на мові Verilog

```
module counter(i_clk, i_rst_n, o_counter);  
  
    input          i_clk;  
    input          i_rst_n;  
    output reg [9:0] o_counter;  
  
    always @(posedge i_clk, negedge i_rst_n) begin  
        if(~i_rst_n) begin  
            o_counter <= 0;  
        end else begin  
            o_counter <= o_counter + 1'b1;  
        end  
    end  
  
endmodule
```

Результат синтезу такого вихідного коду в Quartus Prime наведений на рис.2.11. Зазначений результат синтезу можна переглянути самостійно в RTL Viewer. Інформація по користуванню RTL Viewer наведена в лабораторній роботі Lab0.

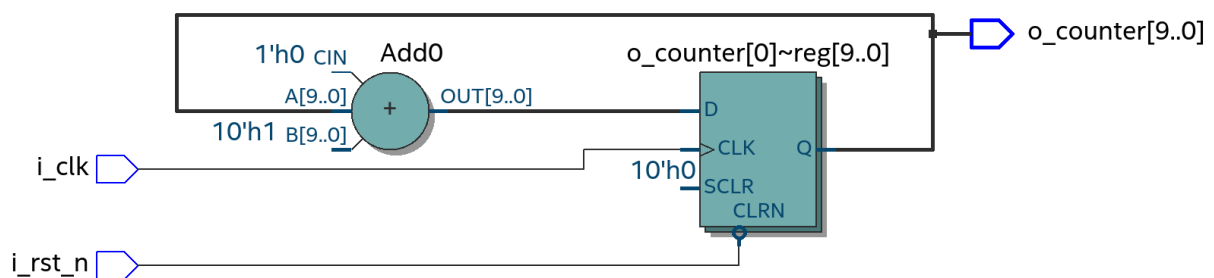


Рис.2.11 - Результат синтезу синхронного лічильника з лістингу 2.1 в RTL Viewer

Схема з рис.2.11 є ще одним варіантом реалізації синхронного двійкового лічильника, що використовує паралельний регістр та суматор, який додає до вмісту регістру одиницю і результат суми по активному фронту сигналу синхронізації записується в регістр.

2.2.5 Опис синхронного лічильника з входом завантаження на мові Verilog

Шаблон опису на мові Verilog для лічильника вгору із завантаженням наведено в лістингу 2.2.

Лістинг 2.2 - Шаблон опису лічильника вгору із завантаженням на мові Verilog

```
module counter(i_clk, i_rst_n, i_load, i_data, o_counter);

input          i_clk;
input          i_rst_n;
input          i_load;
input [9:0]    i_data;
output reg [9:0] o_counter;

always @(posedge i_clk, negedge i_rst_n) begin
    if(~i_rst_n) begin
        o_counter <= 0;
    end else begin
        if (i_load)
            o_counter <= i_data;
        else
            o_counter <= o_counter + 1'b1;
    end
end

endmodule
```

Результат синтезу лістингу 2.2 можна переглянути на рис.2.12. Як видно, схема дуже схожа на лічильник з рис.2.11, однак на рис.2.12 присутній додатковий мультиплексор, за допомогою якого можна підключити вхід даних регістру або до входу *i_data*, або до виходу суматора.

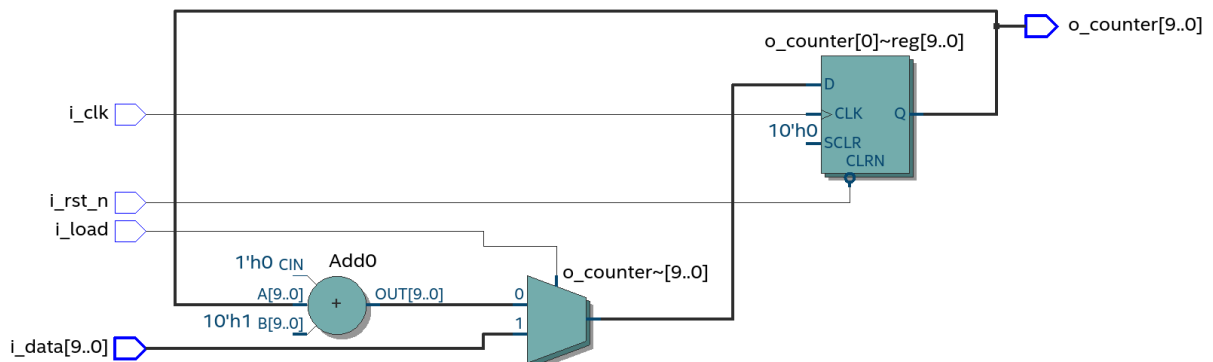


Рис.2.12 - Результат синтезу лічильника з лістингу 2.2 в RTL Viewer

На рис.2.12 мультиплексор має однорозрядний вхід адреси і два багаторозрядних входи з номерами 0 та 1. Вхід адреси мультиплексора підключений до входу i_load .

Якщо однорозрядний вхід адреси приймає значення 0, на вихід мультиплексора передаються дані з входу з номером 0. Якщо однорозрядний вхід адреси приймає значення 1, на вихід мультиплексора передаються дані з входу з номером 1. Більш детально мультиплексор буде розглянуто в одній з наступних лабораторних робіт.

Таким чином, коли вхід i_load приймає значення лог. 1, в регістр, що зберігає стан лічильника, після активного переднього фронту на вході синхронізації i_clk записуються дані з входу i_data . Якщо ж на вході i_load присутній лог. 0, по кожному активному передньому фронту на вході синхронізації вміст лічильника збільшується на 1.

2.2.6 Цифро-аналоговий перетворювач (ЦАП) на резисторах

Даний матеріал необхідний для розуміння наступних розділів лабораторної роботи.

Цифро-аналоговий перетворювач, ЦАП (Digital-to-Analog Converter, DAC) — це пристрій для перетворення цифрового коду в напругу. На цифровий вхід ЦАП подається число розрядністю N , а на виході ЦАП з'являється напруга, що відповідає цифровому коду на вході. Також на ЦАП подається опорна напруга V_{ref} . Напруга на виході ЦАП прямо пропорційна цифровому коду на його вході.

Нулю на цифровому вході ЦАП відповідає нульова напруга на виході. Максимальному цифровому коду на вході N -розрядного ЦАП ($2^N - 1$), відповідає максимально можливе значення напруги на виході, яке називається опорною напругою (Reference Voltage, V_{ref}) і часто дорівнює напрузі живлення.

Формула залежності напруги V_o на виході ЦАП від цифрового N -розрядного коду D на вході та опорної напруги V_{ref} :

$$V_o = \frac{V_{ref}}{2^N - 1} \cdot D$$

Умовне графічне позначення ЦАП наведено на рис.2.13.

Залежність напруги на виході 4-розрядного ЦАП від цифрового коду на його вході наведена на рис.2.14.

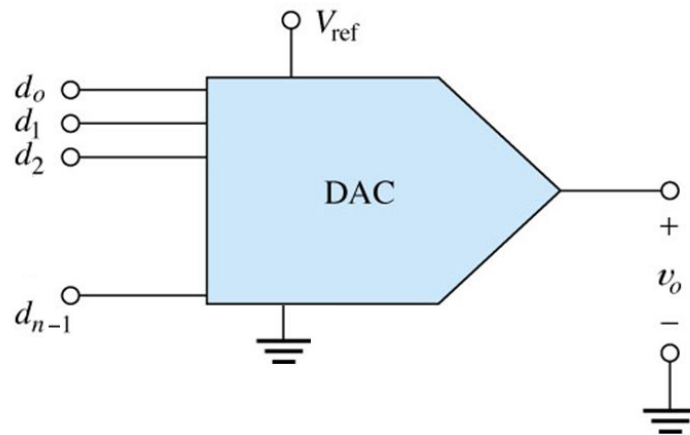


Рис.2.13 - Умовне графічне позначення ЦАП

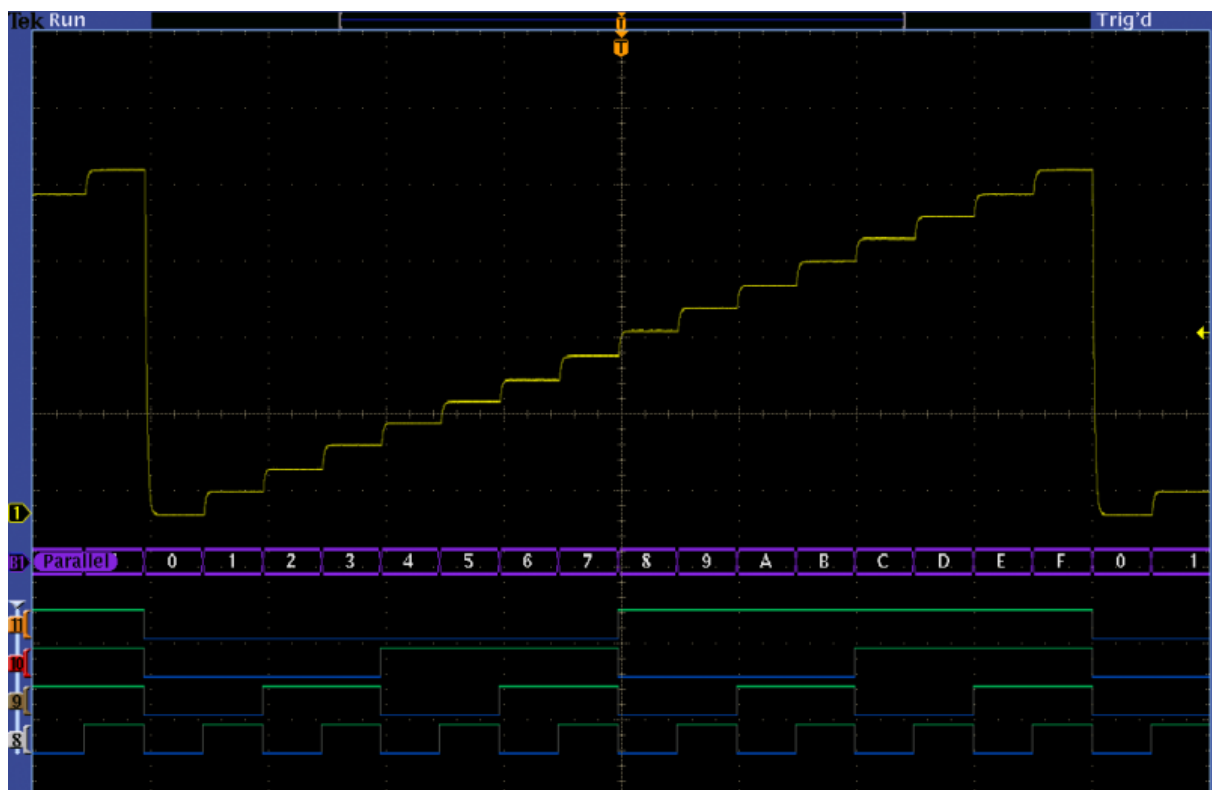


Рис.2.14 - Залежність напруги на виході 4-розрядного ЦАП
від цифрового коду на його вході

Більшість широко використовуваних ЦАП мають розрядність в діапазоні 8 - 16 біт. Однак іноді використовують ЦАП, як дуже низької розрядності (4 біт), так і дуже високої розрядності (24 біт).

Професійні ЦАП випускають у вигляді дискретних мікросхем, однак для багатьох задач підходить ЦАП складений із резисторів.

Існує дві схеми ЦАП на резисторах: R2R ЦАП і ЦАП з ваговими резисторами.

R2R ЦАП дуже розповсюджений і з його конструкцією можна ознайомитись за посиланням [2.1].

Однак в лабораторній роботі ми будемо використовувати ЦАП з ваговими резисторами, оскільки таку конструкцію має 4-розрядний ЦАП інтерфейсу VGA на налагоджувальних платах Intel FPGA, який ми використовуватимемо в генераторах сигналів. Тому розглянемо детальніше ЦАП з ваговими резисторами.

Схема 4-розрядного ЦАП з ваговими резисторами наведена на рис.2.15.

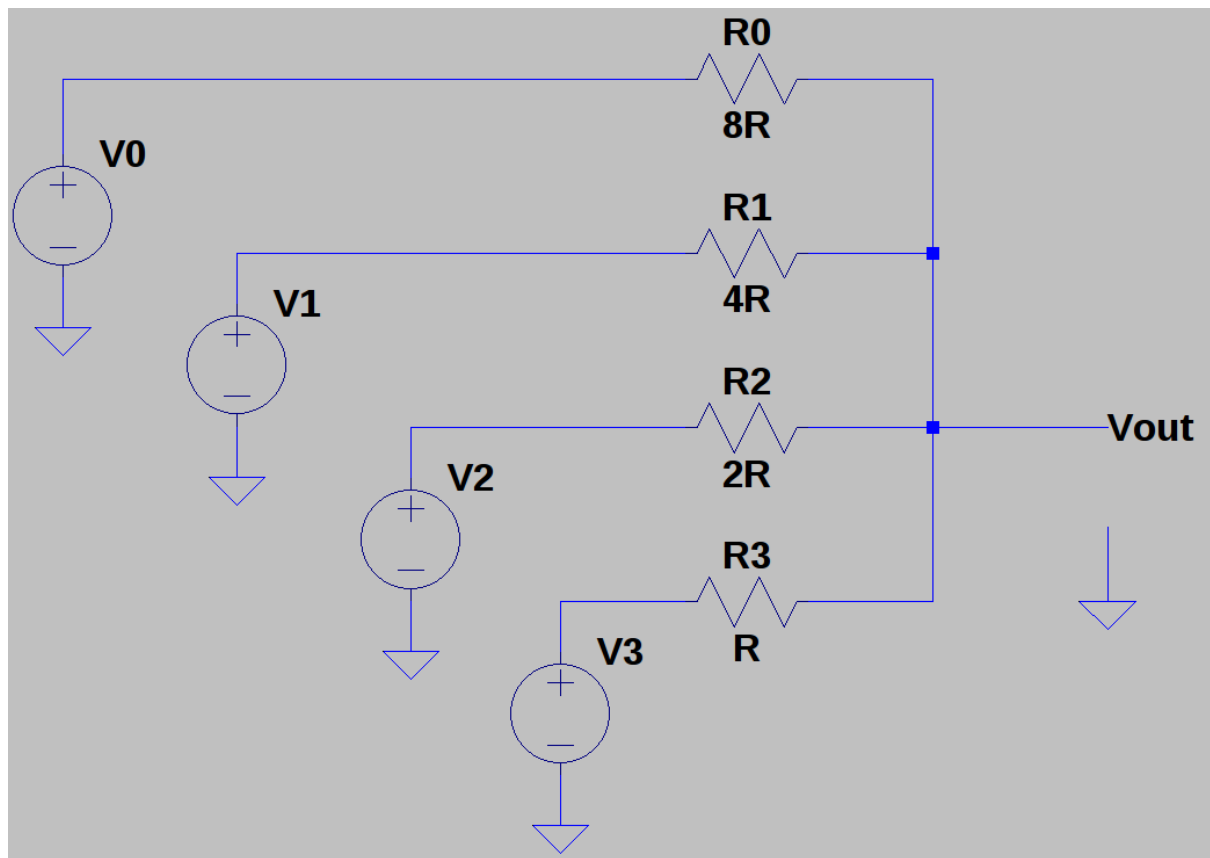


Рис.2.15 - Схема ЦАП з ваговими резисторами

В ЦАП на рис.2.15 вхідний цифровий код задається джерелами напруги, які моделюють логічні рівні. Кожне таке джерело на вході може видавати або лог. 0 (0 вольт), або лог.1. ($V_{ref}=V_{dd}$ вольт, де V_{dd} - напруга живлення). Відповідно напругу кожного джерела з номером i можна представити формулою $V_i = V_{ref} \cdot D_i = V_{dd} \cdot D_i$, де D_i представляє i -тий розряд цифрового коду на вході ЦАП і може приймати значення 0, або 1.

Запишемо перший закон Кірхгофа для струмів через резистори:

$$I_0 + I_1 + I_2 + I_3 = 0 \quad (2.1)$$

З другого закону Кірхгофа напругу на i -му резисторі V_{Ri} можна представити формулою (де V_i - напруга i -го вхідного джерела, а V_{out} - напруга на виході ЦАП):

$$V_{Ri} = V_i - V_{out} \quad (2.2)$$

Підставивши формулу (2.2) в (2.1) для кожного i від 0 до 3 і застосувавши закон Ома, отримаємо:

$$\frac{V_0 - V_{out}}{R_0} + \frac{V_1 - V_{out}}{R_1} + \frac{V_2 - V_{out}}{R_2} + \frac{V_3 - V_{out}}{R_3} = 0 \quad (2.3)$$

Перегрупувавши доданки отримаємо:

$$V_{out} = \frac{\frac{V_0}{R_0} + \frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3}}{\frac{1}{R_0} + \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}} \quad (2.4)$$

Враховуючи, що $R_3 = R$, $R_2 = 2 \cdot R$, $R_1 = 4 \cdot R$, $R_0 = 8 \cdot R$, підставимо ці значення в (2.4) і отримаємо (D - цифровий код на вході ЦАП з розрядністю N біт, D_i - розряди цього цифрового коду):

$$\begin{aligned} V_{out} &= \frac{\frac{V_0}{8R} + \frac{V_1}{4R} + \frac{V_2}{2R} + \frac{V_3}{R}}{\frac{1}{8R} + \frac{1}{4R} + \frac{1}{2R} + \frac{1}{R}} = \frac{V_0 + 2 \cdot V_1 + 4 \cdot V_2 + 8 \cdot V_3}{15} = \\ &= \frac{V_{dd}}{15} \cdot (D_0 + 2 \cdot D_1 + 4 \cdot D_2 + 8 \cdot D_3) = \frac{V_{dd}}{15} \cdot D \end{aligned}$$

2.2.7 Генератор пилкоподібної напруги на базі лічильника і ЦАП

2.2.8 Опис ROM пам'яті на мові Verilog

2.2.9 Генератор синусоїдального сигналу

2.2.10 Генератор синусоїдального сигналу з керованою частотою (NCO)

2.2.11 Тестбенч на мові Verilog (initial процеси, цикли, затримки)

2.3 Практична частина

2.3.1 Симуляція проектів в ModelSim

2.3.2 Налаштування синтезу пам'яті в Quartus Prime

2.3.3 Цифро-аналоговий перетворювач у налагоджувальних платах з VGA

2.3.4 Завдання на лабораторну роботу

Приклади виконання частини завдань можна переглянути за посиланням [[1.9](#)].

1.

2.4 Контрольні запитання

2.5 Перелік посилань

[2.1] “Tutorial: Digital to Analog Conversion – The R-2R DAC”, 2015. [Online].
Available: <https://uk.tek.com/blog/tutorial-digital-analog-conversion---r-2r-dac>

[2.2]