

Лабораторна робота 4

Дискримінатор

1 Мета:

Мета роботи одержати знання і практичний досвід в розробці дискримінатора для сучасних спеціалізованих ЕОМ, одержати досвід програмування і відлагоджування програм на мові VHDL.

2 Теоретичні відомості

Дискримінатор – це кінцевий блок демодуляторів цифрових даних, які використовуються в керуючих контурах промислових установок, радіопристроїв і т.і. Часто його називають детектором. Згідно з законом модуляції сигналу, розрізняються амплітудні (АМ), фазові (ФМ), амплітудно-фазові (АМ-ФМ), квадратурно-амплітудні (QAM), частотні (ЧМ) дискримінатори.

АМ – дискримінатор

АМ – дискримінатор звичайно називається як АМ – демодулятор або детектор. АМ – сигнал може бути випрямлений без необхідності складного демодулятора. Наприклад, сигнал може пропускатися через квадратичний детектор, вихідний сигнал якого повторює амплітудну криву вхідного смугового сигналу.

Коли вхідний сигнал – аналітичний, тобто дорівнює $I + jQ$, тоді вихідний сигнал АМ – дискримінатора – це амплітуда комплексного вектору:

$$A = \sqrt{I^2 + Q^2}.$$

Але функція квадратного корня доволі складна для обчислень. Тому безпосереднє обчислення вектора амплітуди реалізується в дискримінаторах рідко. Але в багатьох випадках вихідний сигнал досить обчислити з похибкою близько 5%. Французський механік Понселе запропонував в 1828 р. найкращу лінійну апроксимацію цієї формули.

Якщо $I > 0, Q > 0, I > Q$, тоді $A \approx 0.9605 * I + 0.3978 * Q$ і

якщо $I > 2Q$, тоді $A \approx 0.9859 * I + 0.2327 * Q$.

В першій формулі одержується апроксимація з похибкою менше ніж 4%, а в другій – з похибкою менше ніж 1.4%.

Коефіцієнти можна представити наступним двійковим кодом:

$0.9605 \approx 0.11110110$ і

$0.3978 \approx 0.0110110$,

і вони представляються наступними сумами дробів:

$0.11110110 = 1 - 1/16 + 1/64 + 1/128$ і $0.0110110 = 1/4 + 1/8 + 1/64 + 1/128$.

Операція множення на ці коефіцієнти може бути замінена на суму зсунутих операндів $|I|$ і $|Q|$, де $|I|$ – абсолютне значення доповнюючого коду I . Результуюча формула апроксимації модуля вектора виглядає як наступна:

$$A \approx |I| - 1/16 * |I| + 1/64 * |I| + 1/128 * |I| + 1/4 * |Q| + 1/8 * |Q| + 1/64 * |Q| + 1/128 * |Q|,$$

Функція обчислення амплітуди комплексного вектора описується на VHDL як наступна:

```
function MAGN_APPROX(A,B: integer) return integer
  variable Re,Im,Tmp,Magn: integer;
begin
  Re:=ABS(A);
  Im:=ABS(B);
  if Re<Im then
    Tmp:=Re; Re:=Im;Im:=Tmp;
  end if;
  Magn:=Re - Re/16+Re/64+Re/128+ Im/4 +Im/8+Im/64+Im/128;
  return Magn;
end function;
```

Спочатку обчислюється абсолютне значення операндів, тоді вони сортуються, щоб потрапити в першого октант прямокутної системи координат, тобто щоб задовольняти $|A| > |B|$. Тоді обчислюється функція Понселе.

Частотний дискримінатор

Існує кілька способів щоб демодулювати ЧМ-сигнал. Найбільш загальний спосіб – використати фільтр, АЧХ якого зменшує амплітуди одних частот відносно інших, і до виходу якого під'єднано АМ-демодулятор. Якщо фільтр має лінійну АЧХ, то результируючий вихідний сигнал буде пропорційний частоті вхідного сигналу. Інший спосіб – це заведення сигналу в контур фазового зворотнього зв'язку і використання похибки слідкування за фазою як демодульований сигнал.

Розглянемо сигнал, частота якого модульована двійковим сигналом (BFSK – binary frequency shift keying). Тоді частота сигналу дорівнює $f = f_0 - \Delta f/2$, коли символ 0 приймається, і $f = f_0 + \Delta f/2$, коли символ 1. де $\Delta f T_C = 1$, T_C символ період. Тому дискримінатор повинен вимірювати різницю $f - f_0$. Розглянемо в k -th clock такт ми маємо виборки вектора сигнал $A_{k-1} = I_{k-1} + jQ_{k-1} = |A| \exp(2\pi f t)$, і в наступний такт він $A_k = I_k + jQ_k = |A| \exp(2\pi f t + \phi)$. Тоді

$$\sin((2\pi f t) + \phi) - \sin(2\pi f t) = \sin(\phi) = (Q_k I_{k-1} - I_k Q_{k-1}) / |A|,$$

Тут кут $\phi = \Delta f T_S / 2$ пропорційний $f - f_0$, де T_S частота дискретизації. Тоді шуканий зсув частоти з частотним масштабним коефіцієнтом K може бути знайдений як

$$\Delta f = K \phi = K \arcsin(\sin(\phi)) \approx K \sin(\phi) = K(Q_k I_{k-1} - I_k Q_{k-1}) / |A|.$$

Коли амплітуда сигналу стабільна, наприклад, коли вона підстроюється відповідно зворотнього зв'язку системи автоматичного регулювання підсилення, тоді формула може бути зпрощена:

$$\Delta f \approx K'(Q_k I_{k-1} - I_k Q_{k-1}).$$

Тут масштабний коефіцієнт K' може бути взятий до уваги коли приймається рішення, який передається символ. Тому він представлений в деякому діапазоні значень і не потребує операції множення.

Фазовий дискримінатор

Фаза комплексного вектора звичайно обчислюється як $\theta = \varphi \pm \arctg(Q/I)$, де $\varphi = 0 \text{ or } \pm \pi/2$ в залежності від квадранту кута. Як арктангенс, так і ділення – це доволі складна операція, яка потребує великий об'єм апаратури об'єм при обчисленнях в реальному часі.

Волдер був перший, хто запропонував розраховувати фазу, використовуючи свій алгоритм CORDIC в 1950-х роках. Більш того, цей алгоритм обчислює амплітуду вектора паралельно з обчисленням кута. Його синонім – алгоритм “цифра за цифрою” означає, що з кожним кроком алгоритму одержується одна точна цифра результату. При цьому базова операція алгоритму використовує тільки додаванням, віднімання і зсуви.

Нехай ми хочемо повернути вектор (X, Y) на кут θ . Тоді координати нового вектору (X_{new}, Y_{new}) дорівнюють:

$$\begin{aligned} X_{new} &= X * \cos(\theta) - Y * \sin(\theta) \\ Y_{new} &= Y * \cos(\theta) + X * \sin(\theta) \end{aligned}$$

Або після ділення на $\cos(\theta)$:

$$\begin{aligned} X_{new} / \cos(\theta) &= X - Y * \tg(\theta) \\ Y_{new} / \cos(\theta) &= Y + X * \tg(\theta) \end{aligned}$$

Можна розділити кут θ на невеликі елементарні кути, такі що тангенси цих кутів завжди дорівнюватимуть ступеню 2. Тоді одержимо наступні рівняння повороту на елементарний кут $\arctg(2^{-i})$ з накопиченням всіх таких кутів в суму θ_i :

$$\begin{aligned} X_{i+1} &= P_i * (X_i - Y_i 2^{-i}); \\ Y_{i+1} &= P_i * (Y_i + X_i 2^{-i}); \\ \theta_{i+1} &= \theta_i + \arctg(2^{-i}). \end{aligned}$$

Тут константа $\arctg(2^{-i})$ може бути обчислена заздалегідь. Коефіцієнт P_i може бути видалений з рівняння, а остаточний результат може бути зкоррегований з його урахуванням. Якщо ми перемножимо всі P_i , то одержимо постійну величину, яка дорівнює 0.607...

Такий алгоритм також годиться, щоб обчислити функцію $\arctg(Q/I)$. Для цього початковий вектор задається як $(Y_0, X_0) = (Q, I)$. Тоді в алгоритмі виконуються n ітерацій:

$$\begin{aligned} \text{sign } \xi_i &= \text{sign } Y_i; \\ Y_{i+1} &= Y_i - \xi_i X_i 2^{-i}; \\ X_{i+1} &= X_i + \xi_i Y_i 2^{-i}; \\ \theta_{i+1} &= \theta_i + \xi_i \arctg(2^{-i}). \end{aligned}$$

Тут $\text{sign } \xi_i = 1$ коли $Y_i > 0$, -1 коли $Y_i < 0$ і 0 коли $Y_i = 0$.

Після n кроків одержимо результати $\theta_n = \arctg(Q/I)$ і $X_n = 1/P * \sqrt{I^2 + Q^2}$. Слід відмітити, що для n кроків n правильну бітс результат обчислений. для цей feature таких алгоритм часто називається як «цифра за цифрою» алгоритм.

Слід відмітити що I розглянемо як $I > 0$. Коли $I < 0$ його абсолютне значення повинні обчислений, і результуючого кут повинні правильну як $\pm\pi - \theta_n$.

В дискримінаторі необхідно відобразити повний кут $\pm\pi$ в діапазон ± 1 . Тому константи арктангенса запам'ятовуються відповідно. Ці константи для 16-бітових даних представлені в наступній таблиці 1:

Таблиця 1

i	$\arctg(2^{-i}) / \pi$	i	$\arctg(2^{-i}) / \pi$
0	0,0100 0000 0000 0000 0	8	0,0000 0000 0101 0001 1
1	0,0010 0101 1100 1000 0	9	0,0000 0000 0010 1000 1
2	0,0001 0011 1111 0110 1	10	0,0000 0000 0001 0100 0
3	0,0000 1010 0010 0010 0	11	0,0000 0000 0000 1010 0
4	0,0000 0101 0001 0110 0	12	0,0000 0000 0000 0101 0
5	0,0000 0010 1000 1011 1	13	0,0000 0000 0000 0010 1
6	0,0000 0001 0100 0110 0	14	0,0000 0000 0000 0001 0
7	0,0000 0000 1010 0011 0	15	0,0000 0000 0000 0000 1

3. Приклад розробки дискримінатора

Розглянемо приклад розробки дискримінатора, який знаходить фазу сигналу за алгоритмом «цифра за цифрою». Початкові дані – 10-розрядні, вихідна фаза має розрядність 6.

Через те, що старший біт результату кодує положення кута в 1 або 4 квадранті, чи в 2 або 3 квадранті, ми повинні виконати $6-1 = 5$ ітерацій алгоритму. В цих ітераціях виконується 5 додавань кутів. Тому коефіцієнти кутів повинні мати розрядність $6 + \lceil \log_2 5 \rceil = 9$.

Модель дискримінатора описується в наступній VHDL-програмі:

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_SIGNED.all;
entity CORDIC5 port( CLK : in STD_LOGIC;
                    RST : in STD_LOGIC;
                    EI : in STD_LOGIC;           --вхід дозволу
                    I : in STD_LOGIC_VECTOR(9 downto 0);
                    Q : in STD_LOGIC_VECTOR(9 downto 0);
                    FI : out STD_LOGIC_VECTOR(5 downto 0) );
end CORDIC5;
architecture BEH CORDIC5
    signal qi,qa,q1,q2,q3,q4,q5:STD_LOGIC_VECTOR(10 downto 0);
    signal ii,ia,i1,i2,i3,i4,i5:STD_LOGIC_VECTOR(10 downto 0);
    signal sq,si,s1,s2,s3,s4,s5:STD_LOGIC;-- sign Q
    signal f1,f2,f3,f4,f5: STD_LOGIC_VECTOR(8 downto 0);
begin
    RGI:process(CLK,RST) begin --   регістри вхідних даних
        if RST='1' then
            ii<=(others=>'0');
            qi<=(others=>'0');
        elsif CLK='1' and CLK'event then
            if EI='1' then
                ii<=I(9)&I;
                qi<=Q(9)&Q;
            end if;
        end if;
    end process;
    -- тепер ми одержимо абсолютне значення I
    ia<=abs(ii);
    sq<=qi(9);           --знаки даних
    si<=ii(9);
    -- перша ітерація
    sq<=qi(9);
    q1<= qi - ia when sq='0' else
            qi + ia;
    i1<= ia + qi when sq='0' else
            ia - qi;
    f1<= "001000000" when sq='0' else --кут 45 grad.
            -"001000000";
    -- друга iteration
    s1<=q1(9);
    q2<= q1 - SHR(i1,"001") when s1='0' else
            q1 + SHR(i1,"001");
    i2<=i1 + SHR(q1,"001") when s1='0' else
            i1 - SHR(q1,"001");
    f2<= f1 + "000100101" when s1='0' else --кут 22,5 grad.
            f1 - "000100101";
    -- 3-я iteration
    s2<=q2(9);
    q3<= q2 - SHR(i1,"010") when s2='0' else
            q2 + SHR(i1,"010");
    i3<=i2 + SHR(q1,"010") when s2='0' else
            i2 - SHR(q1,"010");
    f3<= f2 + "000010011" when s2='0' else --кут 11 grad.
            f2 - "000010011";
    -- 4-a iteration
    s3<=q3(9);
    q4<= q3 - SHR(i1,"011") when s3='0' else
            q3 + SHR(i1,"011");
    i4<=i3 + SHR(q1,"011") when s3='0' else

```

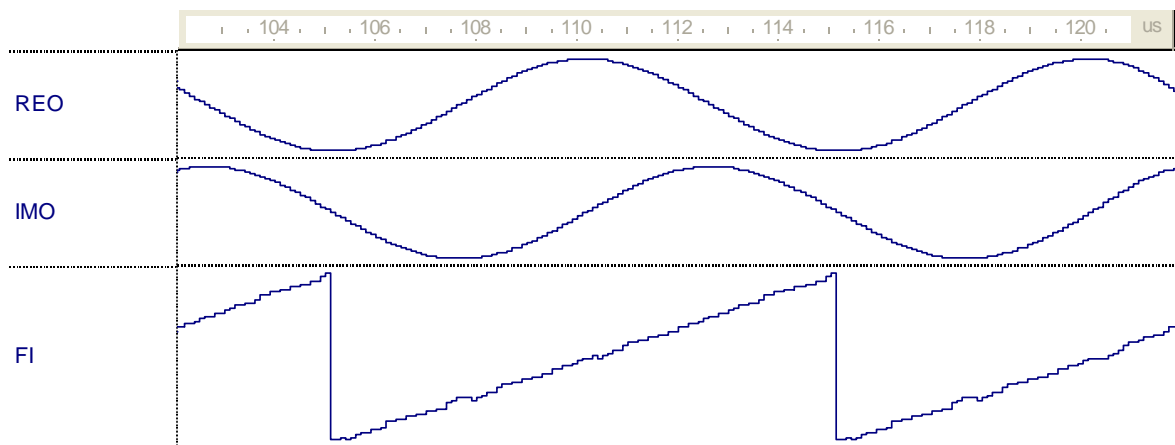
```

        i3 - SHR(q1,"011");
f4<= f3 + "000001010" when s3='0' else      --кут 5,5 grad.
        f3 - "000001010";
-- 5-a iteration
s4<=q4(9);
q5<= q4 - SHR(i1,"100") when s4='0' else
q4 + SHR(i1,"100");
i5<=i4 + SHR(q1,"100") when s4='0' else
        i4 - SHR(q1,"100");
f5<="010000000" when ia(10 downto 4)=0 and sq='0' else--кут 90 grad.
        "110000000" when ia(10 downto 4)=0 and sq='1' else--кут-90 grad.
        f4 + "000000100" when s4='0' else      --кут 2,7 grad.
        f4 - "000000100";

process(CLK,RST) begin      --      -- Регістри Результату
    if RST='1' then
        FI<=(others=>'0');
    elsif CLK='1' and CLK'event then
        if si='0' then
            FI<=f5(8 downto 3);      -- кутів      +-90 grad
        else
            if sq='0' then
                FI<="011111"-f5(8 downto 3)+1;-- кутів+90+180 grad
            else
                FI<="100001"-f5(8 downto 3)-1; -- кут-90-180 grad
            end if;
        end if;
    end if;
end if;
end process;

end BEH;

```



4. Виконання лабораторної роботи

Дискримінатор повинен розроблятися і тестуватися як у вищеприказаному прикладі .

Кожен варіант роботи вибирається за номером залікової книжки студента. Розглянемо 3 останні цифри a_2, a_1, a_0 , номеру залікової книжки. Тоді номер варіанту

$$N = 100a_2 + 10a_1 + a_0 = 2^9b_9 + 2^8b_8 + 2^7b_7 + 2^6b_6 + 2^5b_5 + 2^4b_4 + 2^3b_3 + 2^2b_2 + 2^1b_1 + b_0,$$

де b_i – це біти номеру N в двійковому представленні.

Параметри дискримінатора: розрядність вхідного сигналу N_i і розрядність вихідного сигналу N_o вибираються з Таблиці 2.

Таблиця 2

b_2, b_1, b_0	N_i	N_o
000	10	4
001	10	5
010	12	6
011	12	7
100	14	4
101	14	5
110	16	6
111	16	7

Тип дискримінатора і режим тестування задані в Таблиці 3

Таблиця 2

b_3, b_0	Тип дискримінатора	Генератор тестового сигналу
00	АМ Понселе	Генератор з лабораторної роботи 1 з обмеженням виходного сигналу*
01	ЧМ	Генератор стенду для іспитів з лабораторної роботи 3
10	ФМ цифра за цифрою	Генератор з лабораторної роботи 1
11	АМ цифра за цифрою	Генератор з лабораторної роботи 1 з обмеженням виходного сигналу*

* Обмеження сигналу X означає його обробку за оператором:

$Y \leq X_m$ when $X > X_m$, X when $X > -X_m$ else $-X_m$;

де X_m – поріг обмеження.

5. Звіт про лабораторну роботу

Звіт про лабораторну роботу повинен вміщувати:

- мету роботи,
- опис дискримінатора,
- тексти VHDL,
- графіки тестування,
- висновки.

Література

1. Отнес Р., Эноксон Л. Прикладной анализ временных рядов. –М.:Мир. –1982. – 428 с.
2. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. –М.:Мир. –1978. – 848 с.