

СПОСІБ ТЕСТУВАННЯ ЦИФРОВОГО КАНАЛУ ЗВ'ЯЗКУ

Коцержинський Б. О., д. т. н., професор

КПІ ім. Ігоря Сікорського, м. Київ, Україна

Розглядається такий спосіб тестування цифрового каналу зв'язку, коли у канал у безперервному режимі посиляється заданий тестовий пакет, який складається із преамбули та псевдовипадкової бітової послідовності. Прийнятий бітовий пакет порівнюється з оригіналом пакету для визначення кількості незбігів у пакетах, щоб оцінити якість каналу.

Доцільно мати два пакети: короткий для швидкої перевірки функціонування каналу та довгий для оцінки якості каналу. Реалізація цього способу на сигнальних процесорах [1] потребує багатократного використання пам'яті та послідовної програмної обробки даних, що зменшує швидкість передачі даних. ПЛІС-и можуть забезпечити паралельну обробку даних, тому доцільно дослідити можливість реалізації цього способу на ПЛІС-ах (бажано у реальному часі при швидкостях порядку 100 Мбіт/с). Короткий тестовий пакет складається із 8-розрядної преамбули та 15-бітової псевдовипадкової послідовності (БПП), яка може бути створена генератором на основі 5-розрядного зсувного регістра [2] за таким алгоритмом: перший розряд створює вихідний біт, розраховується сума за модулем 2 значень першого та другого розрядів, виконується зсув розрядів регістра праворуч, сума заміняє п'ятий розряд. Моделювання у системі програмування C++ підтвердило ефективність алгоритму. Код преамбули був вибраний довільно за умови відсутності збігів з кодами усередині БПП.

Програмно-апаратна реалізація виведення короткого пакету можлива за одним із розглянутих варіантів. У першому варіанті замість регістра вибираються 5 тригерів. Відповідний програмний код мовою Verilog представлений як реалізація №1 у таблиці 1.

Один із тригерів при ініціалізації повинен отримати значення 1. Циклічний зсув із модифікацією п'ятого тригера забезпечує створення 15-бітової ПП. У даному варіанті нема виведення перед БПП преамбули, що потребує введення лічильників.

У другому варіанті, враховуючи невелику довжину пакету, пропонується бітовий код пакету завантажити у 23-розрядний регістр та виводити його біти безперервно у циклічному режимі зсуву регістра для збереження його вмісту (реалізація №2 у таблиці 1). Для зсуву тимчасового регістру можна використати операцію зсуву ($t_{mp} \leq t_{mp} \gg 1$).

Довгий тестовий пакет складається із 16-бітової преамбули та БПП на 1023 біти, яка може бути створена генератором на основі 11-розрядного зсувного регістра [2] за таким алгоритмом: перший розряд створює вихідний біт, розраховується сума за модулем 2 значень першого та восьмого розрядів, виконується зсув розрядів регістра праворуч, сума заміняє 11-й розряд.

Таблиця 1

Реалізація №1

```
module _shift_top(A,clk,rst,B,C,D,P,OUT);
output A,B,C,D,P,OUT;
input clk,rst;
reg A,B,C,D,P,OUT,T;
always @(posedge rst or posedge clk)
if(rst)
//ініціалізація тригерів
begin
A<=1;
B<=0;
C<=0;
D<=0;
P<=0;
end
else
begin
OUT=A; //виведення біта
T=A^B;
A=B; //зсув
B=C;
C=D;
D=T; //зміна біта
end
endmodule
```

Реалізація №2

```
module shpack_top(clk,so,r);
input clk,r; //вхідні сигнали синхронізації та запуску
output reg so; //вихідний тригер
reg [7:0] temp;
reg T;
reg[22:0] tmp;

always @(posedge clk or posedge r)
if(r)
//ініціалізація регістра кодом пакету
begin
tmp<=23'b10110010100010011010111; //h5944D7;
end
else
//побітне виведення вмісту регістра
begin
so<=tmp[22];
T=tmp[22];
tmp[22:1]<=tmp[21:0]; //циклічний зсув
tmp[0]=T;
end
endmodule
```

Моделювання у системі програмування C++ підтвердило ефективність

алгоритму. Код преамбули був вибраний довільно за умови відсутності збігів з кодами усередині БПП з перевіркою на C++. Програмний код мовою Verilog представлений у таблиці 2. Задіяні два лічильники для виведення 16 бітів преамбули та 1023 біти БПП.

Таблиця 2. Програмна реалізація довгого пакету

<pre>module preaposl_top(clk,r,pout); input clk,r; output reg pout; reg T; reg[15:0] codpre; reg[9:0] tmp; parameter n=16; integer i,k; assign compare=(i==5'b10000); assign compare1=(k==10'b1111111111); always @(posedge r or posedge clk) if(r) begin codpre<=16'b1110101010101011; i=1'b0;</pre>	<pre>k=1'b1; tmp<=10'b00000000001; end else if(!compare1) begin T=(tmp[0]^tmp[7]); pout=tmp[0]; tmp<=tmp>>1; tmp[9]=T; k=k+1; end else begin i=1'b0; k=1'b1; tmp<=10'b00000000001; end endmodule</pre>
--	---

Симуляція пакетів для ПЛІС-ів ALTERA у системі Quartus II була успішною.

Перелік посилань

1. Коцержинський Б.О. Сигнальні процесори ADSP2181x в генераторах періодичних коливань / Коцержинський Б. О., Храновський А. А. // Вісник НТТУ “КПІ” . Сер. Радіотехніка. Радіоапаратобудування. — 2007. — №35. — с. 149–150.

2. Цифровые методы в космической связи [Текст]: Пер. с англ. / Под ред. С. Голомба, В. И. Шляпоберского; Вступ. статья А. П. Мановцева. — М.: Связь, 1969. — 270 с.: ил.

Анотація

Запропоновані метод тестування цифрового каналу зв'язку та структури тестових пакетів та їх реалізація на ПЛІС-ах.

Ключові слова: тестування, канал цифрового зв'язку, бітові послідовності.

Аннотация

Предложены метод оценки качества цифрового канала связи и структуры тестовых пакетов и их реализация на ПЛИС-ах.

Ключевые слова: тестирование, канал цифровой связи, битовые последовательности.

Abstract

The digital radiochannel quality estimation method and test batches structures are proposed with their FPDA realization.

Keywords: quality estimation, digital radiochannel, bit chains.