

# МОДЕЛИРАНЕ И ОПТИМИЗАЦИЯ НА ПОСЛЕДОВАТЕЛНОСТНИ ЛОГИЧЕСКИ СХЕМИ

Валентина Стоянова Кукенска\*

Иван Симеонов Симеонов\*\*

Технически университет - Габрово

Катедра "Компютърни системи и технологии"

e-mail: \*vally@tugab.bg \*\*isim@tugab.bg

*Koukenska V., Simeonov I., Modeling and Optimization of Sequence Logical Schemes. What is shown in the present paper is a method for modeling and optimization of sequent schemes through the help of specialized software of the firm SYNOPSIS. Structural and behavioral models of counters on the basis of triggers. The models are described by the means of the hardware description language Very High Speed Integrated Circuit Hardware Description Language (VHDL). The testing of the models and the optimization are realized by the specialized system for automated design Synopsis FPGA Express.*

## ВЪВЕДЕНИЕ

Съвременните електронни системи са сложни системи състоящи се от множество компоненти. При автоматизираното проектиране на такива системи се търсят и прилагат подходи за намаляване на времето за проектиране и сложността на задачата. Един такъв подход е свързан с принципите на декомпозиция и йерархия.

Декомпозицията се налага като обща техника за намаляване на сложността на проектите и задачите при разработване на апаратните и програмни части на системите.

Обикновено разделянето на системите се осъществява по начин, който обособява функционално независими блокове. Например цифровите системи са изградени от голям брой логически елементи. За целите на проектирането е полезно да се отделят части от системата които съдържат комбинационни логически схеми (Combinational Logic Circuits, CLC) и последователностни логически схеми (Sequential Logic Circuits, SLC, Finite State Machines, FMS, sequential machines). Разделянето на системите на съставни части подпомага процеса на проектиране.

Комбинационните схеми не съдържат обратни връзки. Те не въвеждат нови характеристики, но могат да бъдат използвани за създаването на полезни гравивни блокове.

Текущото състояние на последователностните схеми се запомня от елементите памет. Следващото им състояние е функция на входовете  $X$  и на текущото състояние. Състоянието на изходите  $Z$  е определя от текущото състояние.

Наличието на обратна връзка в архитектурата на последователностните схеми налага използването на тригери от типа flip-flop за запомнящи елементи.

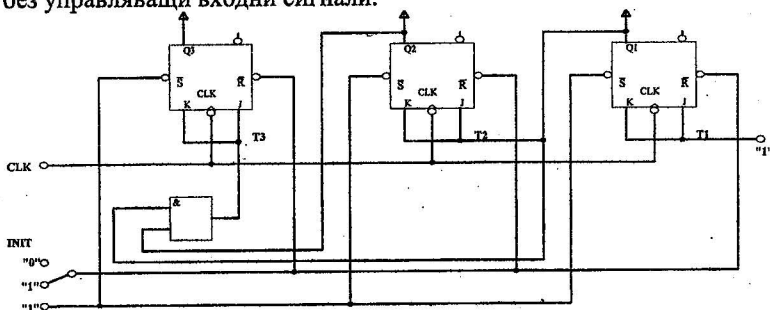
В този доклад е показан един подход за моделиране и оптимизация на последователни схеми с помощта на специализиран софтуер на фирмата SYNOPSIS. Построени са структурни и поведенчески модели на броячи на базата на тригери. Моделите са описани с помощта на средствата на езика за хардуерно описание Very High Speed Integrated Circuit Hardware Description Language (VHDL).

Тестването на моделите и оптимизацията са реализирани със специализираната система за автоматизирано проектиране Synopsys FPGA Express.

## СТРУКТУРА НА БРОЯЧА

Броячите са последователни схеми (ПС), регистриращи постъпващите им на входа сигнали. Често се наричат ПС от броячен тип. Като градивни елементи на брояча се използват елементарни автомати (тригери). Ако броячът е съставен от  $n$  на брой тригери, броят на състоянията е  $2^n$  и се определя като коефициент или модул на броене. Обикновено последният се отбелязва с  $K$ . За броячи се използват Т, JK, D тригери. RS тригерите не се използват поради нецелесъобразност.

На фиг. 1 е представена структурна схема на три-разряден сумиращ брояч с коефициент на броене  $K=8$ . Реализиран е на базата на Т тригери. В кодираната таблица 1 е зададена програмата за работа на броячната схема. Тя работи без управляващи входни сигнали.



фиг. 1. Структурна схема на три-разряден сумиращ брояч

Таблица 1

$Q_3^t$	$Q_2^t$	$Q_1^t$	$Q_3^{t+1}$	$Q_2^{t+1}$	$Q_1^{t+1}$	$T_3$	$T_2$	$T_1$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Възбудителните функции  $T_i$  се получават след минимизация с помощта на карти на Вейч. От анализа на възбудителните функции може да се направи извода: в сумиращ двоичен брояч възбудителната функция на входа на всеки тригер тип Т представлява логическо произведение от сигналите от единичните изходи на всички по-младши тригери. На входа на най-младшия тригер се подава константа единица "1". Следователно един  $n$ -разреден сумиращ двоичен брояч може да се опише чрез система от възбудителни функции:

$$(1) \quad T_1 = 1;$$

$$(2) \quad T_i = \bigwedge_{j=1}^{i-1} Q_j, \quad i = 2, 3, \dots, n.$$

Броячът от фиг. 1 брой тактуваните импулси. Променя вътрешното си състояние на всеки тактов импулс (CLK).

Ако броячът трябва да брой не тактуваните импулси, а сигналите  $x$  на отделен броячен вход, зависимостите (1) и (2) добиват вида (3) и (4).

$$(3) \quad T_1 = x;$$

$$(4) \quad T_i = \bigwedge_{j=1}^{i-1} Q_j, \quad i = 2, 3, \dots, n$$

На  $\bar{R}$  входовете се подава сигнал константа нула "0" за нулиране на всички тригери. За да се превключват тригерите на входовете  $\bar{R}$  и  $\bar{S}$  сигналът трябва да бъде константа единица "1". Тъй като входът  $T_1$  е свързан към логическа единица "1", схемата ще се превключва на всеки входен тактов импулс (CLK), като при това ще променя състоянията си от 0 до 7.

## МОДЕЛИРАНЕ И ОПТИМИЗАЦИЯ

Процесът на проектиране на електронни схеми, устройства и системи се реализира на няколко етапа. На всеки от тях се решават определени задачи. Задачите за моделиране и изследване са свързани с почти всички етапи на проектиране.

Самият процес на проектиране е свързан с три типа описания на проекта: поведенческо, структурно и физическо.

Поведенческото описание специфицира системата по нейното действие без да е необходимо да се знае от какви блокове ще се реализира.

Структурното описание представя структурата на схемата. То отразява отделните блокове, както и връзките между тях.

Физическото описание е свързано с реализацията на самата структура така че да се осигури необходимото поведение на схемата.

Обект на разработката са структурния и поведенчески модел получени при описание на схема на брояч. Те са представени със средствата на езика за хардуерно описание VHDL.

```
entity COUNTER3 is
port (CLK: in bit;
      RESET: in bit;
```

```

COUNT: out integer range 0 to 7);
end COUNTER3;
architecture ARCH_COUNTER of COUNTER3 is
signal COUNT_tmp: integer range 0 to 7;
begin
  process
  begin
    wait until (CLK'event and CLK='1');
    if RESET='1' or COUNT_tmp=7 then
      COUNT_tmp <=0;
    else COUNT_tmp<= COUNT_tmp+1;
    end if;
  end process;
COUNT<= COUNT_tmp;
end ARCH_COUNTER;

```

Тестването на моделите и оптимизацията са реализирани със специализираната система за автоматизирано проектиране Synopsys FPGA Express.

FPGA Express е инструментална програмна среда за синтез и оптимизация на цифрови схеми. С нея може да се създава и оптимизира FPGA нетлист от VHDL и Verilog кодове. Изходните данни могат да бъдат записани и във форматите EDIF и XNF.

За самата оптимизация е необходимо да се избере устройство. След това да се определят и посочат необходимите параметри като работна честота, производител, тип на елемента, фамилия на готовите чипове.

Системата реализира самата оптимизация, прави анализ и генерира изходен код. Последната стъпка на оптимизацията е създаване на FPGA нетлист файлове.

За схемата от фиг. 1 е извършена оптимизация за Speed и Area с две различни цели устройства на фирмата Xilinx и Altera.

Използвана е фамилията XC4000 на фирмата Xilinx. Структурата на всяка клетка от архитектурната матрица съдържа три елемента: матрица от конфигурируем логически блокове, входно-изходни блокове, програмируеми свързващи линии.

Всеки конфигурируем блок съдържа три функционални генератора, два тригера и съответната им управляваща логика. Двата генератора са организирани от справочна таблица и има по четири независими входа. Третият генератор може да реализира произволна булева функция от подадените на входа три променливи.

Всеки конфигурируем логически блок съдържа специална логика за генериране на бърз пренос и заем. Запаметяващите елементи се превключват по фронт на тактовия сигнал.

Трасиращите ресурси са вертикални и хоризонтални. Всеки канал има сегменти с различни дължини и предназначение.

От фирмата Altera е използвана фамилията FLEX 10K.

Получените резултати от оптимизацията на построените модели са в табличен вид .

Практически са изследвани различни времеви съотношения съответстващи на различни коефициенти на запълване на сигнала.

При регистрационна честота 62 MHz за двете фамилии на Xilinx и Altera се получават различни резултати. При това за фамилия XC4000 се получава по-голяма стойност за честотата 158 MHz. Такава честота се получава за фамилията FLEX 10K при регистрационна честота 75 MHz.

При подаване на сигнал по преден фронт от всички портове към устройство RC-N78 се получава закъснение от 2ns (фиг.2), което е много по-малко от предвиденото 16ns. Такъв сигнал се подава и на всички изходни портове. Получава се времезакъснение от 6ns (фиг.2), което също е по-малко от предвиденото.

Signal	Delay (ns)
CLK	13.9
RESET	13.6
Output Slack	9.7

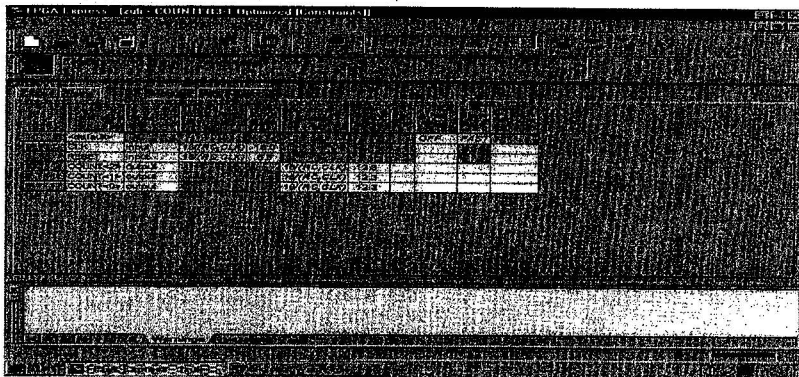
Фиг.2

При оптимизация с целево устройство от фамилията XC4000 на фирмата Xilinx при подаден сигнал на входа CLK за Input Slack се получава 13,9. А при подаване на сигнал RESET се получава 13,6. На изхода за Output Slack се получава една и съща стойност 9,7 (фиг.3).

Signal	Delay (ns)
CLK	13.9
RESET	13.6
Output Slack	9.7

Фиг.3

При оптимизация с целево устройство от фамилията FLEX 10K на фирмата Altera за Input Slack се получава 16,0. А при подаване на сигнал RESET се получава 6,6. На изхода за Output Slack се получава по-висока стойност 12,3. (фиг.4).



Фиг.4

## ЗАКЛЮЧЕНИЕ

Построени са структурни и поведенчески модели на броячи на базата на тригери. Моделите са описани с помощта на средствата на езика за хардуерно описание VHDL.

Изследването и оптимизацията на моделите са реализирани със специализираната система за автоматизирано проектиране. За целта са използвани устройства от фамилията XC4000 на фирмата Xilinx и от фамилията FLEX 10K на фирмата Altera.

Получените резултати показват, че решението с устройствата от фамилията XC4000 са по-добри.

Предложен е алгоритъм, който може да се използва за моделиране и оптимизация на последователности логически схеми.

## ЛИТЕРАТУРА

1. Амстронг Д.Р., Моделирование цифровых систем на языке VHDL., М., Мир, 1992.
2. Гиздарски Е., Проектиране с програмируема логика, Русе, 1998.
3. Даковски Л., Анализ и синтез на логически схеми., София, 1996.
4. Lipsett R., C. Ussery, VHDL: Hardware Description and Design, 1989.