

# РЕАЛИЗАЦИЯ НА АЛГОРИТЪМ ЗА ЦИФРОВА ФИЛТРАЦИЯ ЗА ПРАВОЪГЪЛНИ СИГНАЛИ

Благомир Росенов Дончев, Стефан Мирославов Зарев  
Технически Университет – София, ECAD лаборатория  
donchev@ecad.vmei.acad.bg

*Donchev B.R., Zarev S.M., Implementation of a digital filtering algorithm for rectangular signals. The objective of this paper is to demonstrate a particular implementation of filtering of a periodic rectangular pulse signal with a finite spectrum. As a result of the implementation, we got a functional system, which filters a rectangular pulse signal under a noise impact. The requirement with respect to the noise is that the noise be preliminary determined. An advantage of the method is the fact that the amplitude/phase and phase/frequency pulse spectrum recovers with a neglecting small error based on two preliminary chosen and not noised spectral signal constituents. A "Top-Down" design method was carried out which reduced considerably the developing time. As a basis of the physical implementation, a Xilinx Virtex II FPGA architecture was chosen. The design was developed in ECAD laboratory.*

## 1. ВЪВЕДЕНИЕ

В теорията се използват два класа математични представяния на сигналите – в координатната област и в честотната област. При конкретно теоретично изследване, ефективността на двете представяния е различна. Изборът на единия или на другия клас се определя от спецификата на случая. За анализа на процеса филтрация е по-подходящо представянето на сигналите да бъде в честотната област.

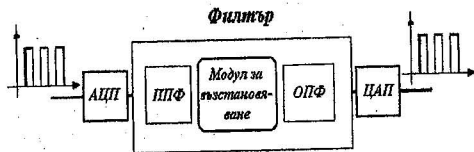
Един от проблемите в радиотехниката – възстановяването на периодичен импулсен сигнал, може да се реши чрез екстраполация и интерполация на спектъра на сигнала. В настоящата разработка се разглежда един частен случай, който се отнася до филтрация на периодичен **правоъгълен** импулсен сигнал с краен спектър, в зашумена област. Математическото описание на сигнала е направено чрез формулата за апроксимация на правоъгълния импулс [1].

$$S(n\Delta t) = S((n - mj)\Delta t) \quad (1)$$

Изискване на метода е шума насложен в спектъра на импулса да не е случаен характер, т.е. трябва предварително да е определена коя част от спектъра е под влияние на шума. Предимство на метода, е че позволява пълното

възстановяване на амплитудно-честотния и фазово-честотния спектър на импулса,

от информацията взета за две произволни не зашумени хармонични съставки. От тази гледна точка този метод може да послужи не само като филтър, но и като формирова̀тел на право̀гълни импулси.



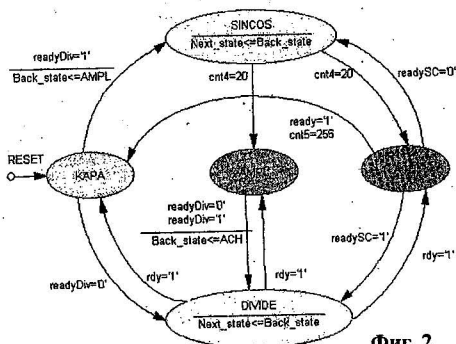
Фиг.1

Блоквата схема на завършена система за филтрация на право̀гълни импулси ще има следния вида показан на фиг.1.

## 2. ПРИНЦИП НА РАБОТА

Проектираният модул има 4 входа: Вход за тактов сигнал **CLK**, вход за **RESET** и два 16 битови входа за стойностите на първи и втори хармоник. На изхода е реализиран интерфейс към външен модул за обратно преобразуване по Фурие, като на всеки активен фронт на сигнала **DATA**, на шините **Amplitude** и **Phase** са подадени 24 битови стойности, съответно за амплитудно и фазово-честотния спектър.

Работата на устройството се управлява с помощта на краен автомат показан на фиг.2



Фиг. 2

Първото състояние което заема крайният автомат е **KAPA**. При първоначалното влизане се взимат входните стойности на първите две хармонични съставки, подготвят се за делене и **NEXT\_STATE** присвоява **DIVIDE**, а **BACK\_STATE** – **KAPA**. След като е извършено деленето, отново се връщаме в състояние **KAPA**, където полученото съотношение се сравнява с предварително записаните в таблица константи, за да се избере на какъв коефициент  $\chi$

отговаря. След като сигналът **kapa\_sig** получи правилната стойност, трябва да преминем в състояние **AMPL**. За да изчислим амплитудата ще ни трябва резултата

от синусовата функция от формула (6.1.2) и затова ще използваме състояние **SINCOS**, като на **BACK\_STATE** ще присвоим **AMPL**. На сигнала **control** се подава '1', за да стартира процесът **FastControl**.

Влизайки в състояние **AMPL**, вече имаме стойността на знаменателя във формула [2].

$$A = \frac{C_2\pi}{\sin(2\chi\pi)} \quad (2)$$

Изчисляваме числителя във формулата и преминаваме в състояние **DIVIDE**. След като се върнем в **AMPL**, с вече изчислената амплитуда, преди да преминем в **ACH**, ще преминем отново през **DIVIDE**, за да подготвим предварително една константа от формула [3] – **ampl\_div\_PI** =  $2A/\pi$ , която ще ни трябва за изчислението на АЧХ. **NEXT\_STATE** присвоява **DIVIDE**, а **BACK\_STATE** – **ACH**.

Зависимостите които се изчисляват в състояние **ACH** са:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (3)$$

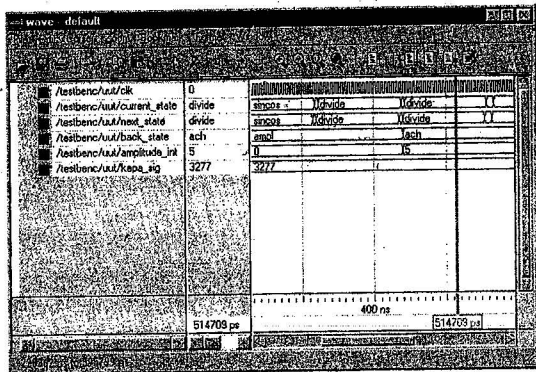
, където  $X$  и  $Y$  са координатите на вектора, а  $\theta$  е ъгълът на изменение. В първия етап се изчислява числителят във формулата. За целта **ACH** се обръща  $n$  пъти към състояние **SINCOS** и натрува получените стойности от синусовата функция в масив от регистри. Аргументите на функцията се взимат от предварително изчислен в паралелния процес **Fast** регистър. След като бъдат изчислени всички стойности на синуса се вдига флаг **ready** в единица. Във втория етап се изчисляват стойностите на амплитудно-честотния спектър, като за целта натрупаните числителите във формулата и знаменателя се разделят, преминавайки през състояние **DIVIDE**. След приключване на всички изчисления се преминава отново в състояние **KAPA**.

**Състояние SINCOS**. Служи за управление на включения към проекта модул за изчисляване на синус и косинус. При влизане в това състояние, трябва предварително да сме присвоили на сигнала **THETA** стойността на ъгъла, за който искаме да бъдат изчислени синус и косинус. На сигнала **ena** се присвоява '1', което запуска компонента и след 20 такта на шините **sinus** и **cosinus** получаваме изчислените стойности. Сигналят **ena** присвоява '0' и се вдига флаг **readySC** в единица. **NEXT\_STATE** присвоява стойността на **BACK\_STATE**.

**Състояние DIVIDE**. Служи за управление на включения към проекта модул за изчисляване на частното на две числа. Такъв модул е необходим, защото синтезаторите поддържат делене на числа, които са само от степените на двойката. При влизане в това състояние, трябва предварително да сме присвоили на сигналите **delimo** и **delitel**, стойностите, на които търсим частното. На сигнала **St** се присвоява '1', което запуска компонента. Когато сигналят **rdy** присвои

стойност '1', означава, че деленето е завършило и резултатът е присвоен на шината **rezultat**. Сигналят **St** се връща в '0' и се вдига флаг **readyDIV** в единица. **NEXT\_STATE** присвоява стойността на **BACK\_STATE**.

### 3. АНАЛИЗ НА РЕЗУЛТАТИТЕ И ГРЕШКАТА



Фиг. 3

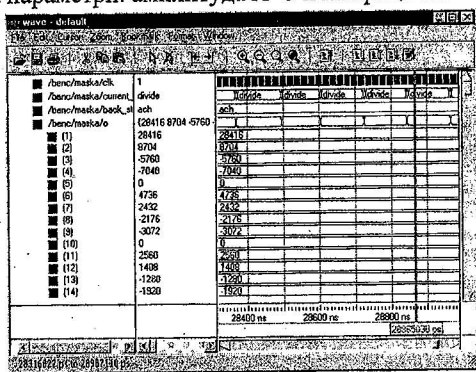
Като входни данни за модула са подадени стойностите на първите две хармонични съставки, съответно 3 и 1. Целта е да се проследи процесът на възстановяване на честотния спектър от проектирания модул, като получените резултати ще се сравнят с тези от теоретичния модел.

На фиг. 3 последните два изобразени сигнала са **amplitude\_int** и **ka\_sig**.

Както може да се види те заемат стойности 5 и 3277. Стойността на **ka\_sig** е мащабирана с коефициент  $2^{13}$ . Ако разделим 3277 на  $2^{13}$ , получаваме 0,4. Обработваният с MathLab импулс има параметри: амплитуда  $A=5$  и коефициент на запълване  $\chi = 0,4$ . Може да се каже, че модулет възстановява коректно параметрите на импулса.

На фиг.4 са изобразени стойностите на първите 15 дискрета от възстановения амплитудно-честотен спектър (сигналят "o"). Стойностите са мащабирани с коефициент 10000.

Резултатите от MathLab за първите 14 дискрета, умножени с 10000 и в целочислен вид са показани в таблица 1:

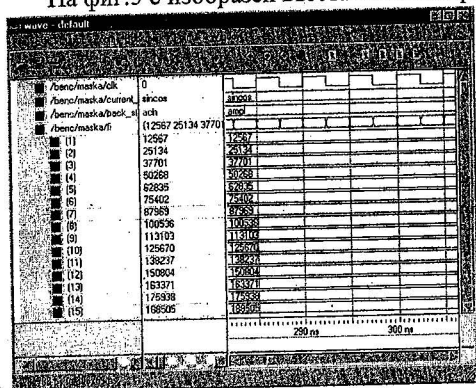


Фиг.4

Таблица 1

O(1)	O(2)	O(3)	O(4)	O(5)	O(6)	O(7)
29971	10098	-5454	-7805	-935	4679	3370
O(8)	O(9)	O(10)	O(11)	O(12)	O(13)	O(14)
-1522	-3532	-924	2324	2206	594	-2262

На фиг.5 е изобразен възстановеният фазов спектър.



Фиг.5

Резултатите от MathLab за първите 14 дискрета, умножени с 10000 и в целочислен вид са показани в таблица 2.

Грешките от възстановяването на двата честотни спектъра са анализирани с продукта MathLab. Изчислени са средноквадратичните грешки от резултатите, дадени от симулациите на схемата и спектрите на идеален импулс със същите параметри. Опитно е установено, че грешките имат нормално разпределение в целия честотен спектър, т.е. няма грешки от

нагруване. Напълно възстановени са основните параметри на правоъгълния импулс: амплитуда  $A = 5$  и  $\chi = 0,4$ . Амплитудно-честотният спектър се възстановява със средноквадратична грешка от 2,4%, а фазо-честотният спектър с грешка от 1,5%. Тези грешки не се отразяват във формата на възстановения импулс във времевата област.

Таблица 2

Fi(1)	Fi (2)	Fi (3)	Fi (4)	Fi (5)	Fi (6)	Fi (7)
12566	25133	37699	50266	62832	75398	87965
Fi (8)	Fi (9)	Fi (10)	Fi (11)	Fi (12)	Fi (13)	Fi (14)
100531	113098	125664	138230	150797	163363	175930

Резултатите от синтеза и имплементацията на проектирания модул са дадени в приложението. За физическа реализация е използвана FPGA програмируема матрица на фирмата Xilinx (XC2V250-SFG256). Тактовата честота, на която може да работи избрания чип е  $F_{\text{system}}=400 \text{ MHz}$ .

Проектираният модул извършва пълното възстановяване на амплитудно и фазово-честотния спектър на един правоъгълен импулс за 12650 такта. От това следва, че честотата на правоъгълни сигнали които могат да се възстановяват е 31KHz ( $400\text{MHz}/12650$ ). Условие на използвания метод е входният аналогов сигнал да бъде дискретизиран 256 пъти за един период. Следователно честотата на дискретизация  $f_d$  на използвания АЦП ще бъде 8MHz ( $256 \times 31\text{KHz}$ ).

Според критерия на Найкуист [4], имаме

$$F_{\max} \leq \frac{f_d}{2} = \frac{8\text{MHz}}{2} = 4\text{MHz} (4)$$

, където  $F_{\max}$  е най-високата честота в спектъра на входния правоъгълен сигнал. Следователно в спектъра на входните правоъгълните импулси след дискретизацията остават първите 128 (4MHz/31KHz) хармоника, което е напълно достатъчно за коректното им възстановяване.

#### 4. ЗАКЛЮЧЕНИЕ:

Целта на представената разработка е да представи хардуерна реализация на теоретичен модел за възстановяване честотния спектър на сигнали.

За реализация на хардуерната част е избрана програмируема логика. Това е в следствие на всички предимства, които тя предлага, като възможност за усъвършенстване на дизайна и последващо имплементиране на нови алгоритми.

Избраният метод за проектиране е "top-down", позволяващ функционалните алгоритми да се опишат на абстрактно поведенческо ниво с езика за хардуерно проектиране VHDL.

Разработения модул позволява възстановяването на целия честотен спектър и параметри на правоъгълен импулс от стойностите на първите две негови хармонични съставки. Намира приложение в области, където имаме силно зашумени бавно изменящи се сигнали.

#### 5. ИЗПОЛЗВАНА ЛИТЕРАТУРА:

1. Димитров Д. Ръководство по основи на радио-съобщителната техника, София, Техника, 1985
2. Фердинандов Е., Сигнали и Системи част 1, София, Сиела, 1999
3. Стоянов Г., Теоретични основи на съобщителната техника, София, Техника, 1993
4. Василева Т., Автоматизация на проектирането на специализирани интегрални схеми, София, Техника, 1997
5. Василева Т., Ръководство по машинно проектиране на интегрални схеми и електронни възли, София, Техника, 1997.
6. Xilinx, Virtex-II Product Datasheet