

Синтезиране на управляващ блок в цифрова схема за формиране на сигнали със синусоидална ШИМ

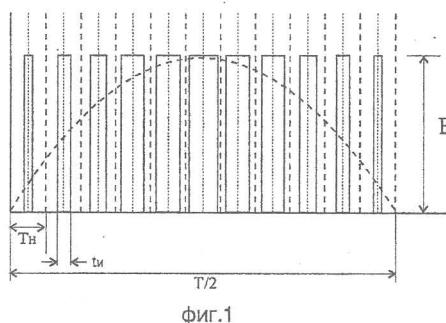
Петър Горанов, Марин Христов, Мариана Горанова, Антоний Трифонов
Технически университет - София

Abstract—The objective of the article is synthesizing of a control unit (CU), which is necessary for the correct functioning of a monolithic analog-digital integrated circuit to provide sinusoidal pulse-width modulated (PWM) output signals. The CU can be modelled with the high level hardware description language VHDL and subsequently the translation to the schematic level can be done with the system for automated design SYNOPSYS, which provides schematic editor, VHDL simulator and digital synthesizer. The used modern methods, tools and programmes allow an algorithmic, functional, structural-logical and topological design of the custom integrated circuit so that the received results satisfy completely the required technical specifications.

1. Увод

В много специализирани системи за контрол и управление, в телекомуникациите, при компютърна обработка на данни, в енергетиката, при задвижване на асинхронни двигатели се налага резервиране на захранващата мрежа и синтезиране на напрежение със синусоидална форма. Най-разпространеният начин за постигане на тази цел е използването на сигнали с широчинно-импулсна модулация (ШИМ) по синусоидален закон, които управляват силови електронни преобразуватели.

Еднополярният вариант на сигнал със синусоидална ШИМ е показан



фиг.1

на фиг. 1. Всеки полупериод е съставен от серия импулси с фиксирана честота $f_H = 1/T_H$, която е много по-висока от честотата $F = 1/T$ на синтезираната синусоида. Ако изменението на продължителността на импулсите става по синусоидален закон $t_u/T_H = \mu \sin(\Omega t)$, се получава $u_H = \mu \cdot E \cdot \sin(\Omega t)$, където $\mu = 0 \div 1$ е дълбочината на модулацията, а $\Omega = 2\pi/T$. Следователно, ефективната стойност на изходното напрежение може да се променя с изменение на коефициента μ .

2. Блокова схема за синтезиране на ШИМ сигнали

Синусоидални ШИМ сигнали могат да се синтезират по различни



ФИГ. 2

в блоков вид на фиг. 2.

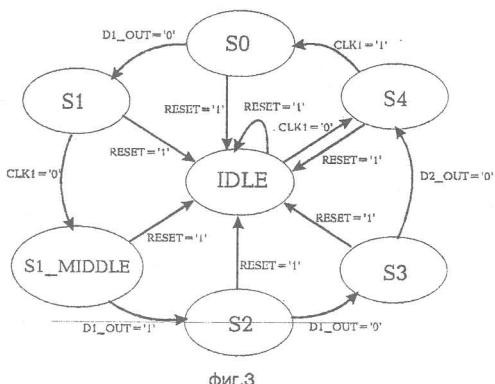
3. Проектиране на управляващ блок

От съществено значение за правилното функциониране на схемата е навременното превключване и коректно предаване на сигнали от един блок към друг. Тези контролни функции се изпълняват от управляващия блок (УБ). УБ е обхванат от обратна връзка, поради което той представлява последователна схема. Тъй като изходите зависят както от състоянието така и от стойностите на входните сигнали, то УБ се представя като автомат на Mealy (фиг. 3). Превключването от едно състояние в друго се осъществява при настъпване на дадено събитие. Под събитие се разбира проверката за това дали даден входен сигнал е

изменил свое то ниво.

Схемата на УБ е синхронна. Тя се тактува от входния сигнал CLK, който управлява всички операции в системата. Състоянието IDLE трае от момента на активиране на $\text{RESET} = '1'$. Ако в този момент $\text{CLK1} = '0'$, то автоматът преминава в състояние S4, в противен случай той ще стои в състоянието IDLE. Коректната работа на автомата се осигурява след първия нарастващ

State Diagram



Фиг.3

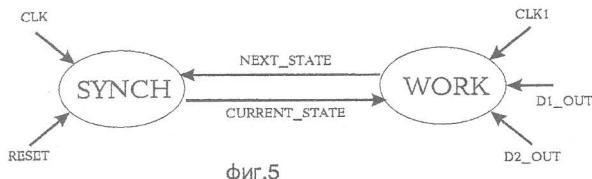
architecture BEHAVIORAL of FSM is

```
type STATE_TYPE is (S0,S1,S1_middle,S2,S3,S4,IDLE);
signal CURRENT_STATE, NEXT_STATE: STATE_TYPE;
begin
WORK:process(CURRENT_STATE,CLK1,D1_OUT,D2_OUT)
begin
    NEXT_STATE <= CURRENT_STATE;
    case CURRENT_STATE is
when S0 =>
if D1_OUT='0' then
    EN_G3_C1 <='0';--
    NEN_G3_C1 <='0';
    NEN_G2_C1 <='0';--
    EN_G3_C2 <='1';
    NEN_G3_C2 <='0';--
    NEXT_STATE <= S1;
else
    NEXT_STATE <= S0;
end if;
when S1 =>
if CLK1='0' then
    EN_G3_C1 <='0';--
    NEN_G3_C1 <='0';--
    NEN_G2_C1 <='1';
    EN_G3_C2 <='0';
    NEN_G3_C2 <='1';
    NEXT_STATE <= S1_middle;
else
    NEXT_STATE <= S1;
end if;
when S1_middle =>
if D1_OUT='1' then
    EN_G3_C1 <='0';--
    NEN_G3_C1 <='0';
    NEN_G2_C1 <='1';
    EN_G3_C2 <='0';
    NEN_G3_C2 <='1';
    NEXT_STATE <= S2;
else
    NEXT_STATE <= S1_middle;
end if;
when S2 =>
if D1_OUT='0' then
    EN_G3_C1 <='1';
    NEN_G3_C1 <='0';--
    NEN_G2_C1 <='0';
    EN_G3_C2 <='0';--
    NEN_G3_C2 <='1';
    NEXT_STATE <= S3;
else
    NEXT_STATE <= IDLE;
end if;
when S3 =>
if D2_OUT='0' then
    EN_G3_C1 <='1';--
    NEN_G3_C1 <='0';--
    NEN_G2_C1 <='0';--
    EN_G3_C2 <='0';--
    NEN_G3_C2 <='0';
    NEXT_STATE <= S4;
else
    NEXT_STATE <= S3;
end if;
when S4 =>
if CLK1='1' then
    EN_G3_C1 <='0';
    NEN_G3_C1 <='1';
    NEN_G2_C1 <='0';--
    EN_G3_C2 <='0';--
    NEN_G3_C2 <='0';
    NEXT_STATE <= S0;
else
    NEXT_STATE <= S4;
end if;
when IDLE =>
if CLK1='0' then
    EN_G3_C1 <='1';--
    NEN_G3_C1 <='0';--
    NEN_G2_C1 <='0';--
    EN_G3_C2 <='0';--
    NEN_G3_C2 <='0';
    NEXT_STATE <= S4;
else
    NEXT_STATE <= IDLE;
end if;
end case;
end process;
```

```
SYNCH: process
begin
    wait until CLK'event and CLK = '1';
    if RESET='1' then
        CURRENT_STATE <= IDLE;
    else
        CURRENT_STATE <= NEXT_STATE;
    end if;
end process;
end BEHAVIORAL;
```

ФИГ.4

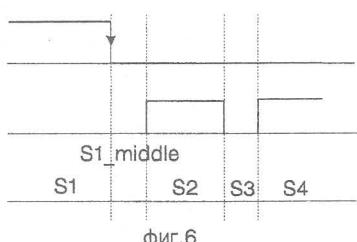
(преден) фронт на CLK1 и той преминава в състояние S0. Състоянието S1 се въвежда от S0, когато D1_OUT='0'. От S1 автоматът превключва в S1_middle при CLK1='0'. Състоянието S1_middle е междинно. То трае един такт на CLK и ни гарантира, че сигналът D1_OUT ще се види на високо ниво. Последното събитие (D1_OUT='1') превключва автомата в състояние S2. В противен случай автоматът преминава към S3. Състоянието S4 се въвежда от S3, когато D2_OUT='0'.



фиг.5

процеса SYNCH и WORK и сигналите, които управляват процесите. На всеки нарастващ фронт на CLK се активира процесът SYNCH, който от своя страна изработва сигналът CURRENT_STATE. Той активира процеса WORK като автоматът остава в същото състояние, в което е бил, ако междувременно не е настъпило дадено събитие. Същественото в случая е, че автоматът преминава в друго състояние, само ако е настъпило точно определено събитие за даденото състояние. Това се вижда от диаграмата (фиг. 6).

Процесите SYNCH и WORK са структурирани чрез различни методи.



фиг.6

ство на създадената архитектура с един синхронизиращ сигнал е и фактът, че е без значение кога ще се появи фронтът на дадено събитие. Ако дадено събитие изпревари друго, това няма да наруши правилната работа на УБ, тъй като се проверява единствено нивото на сигнала.

4. Синтезиране на управляващия блок

От описанието на фиг. 4 е синтезирана цифрова схема чрез системния продукт SYNOPSYS, която е показана на фиг. 7. При кодирането на езика VHDL са спазени следните изисквания:

VHDL моделът на крайния автомат включва два процеса (фиг. 5), които си предават информация един на друг. VHDL описанието е представено на фиг. 4. Изобразени са двата

WORK се активира, когато се случи дадено събитие от обявения му списък. SYNCH се активира само тогава, когато се случи събитието - преден фронт на сигнала CLK. Този механизъм на работа осигурява само един сигнал, по който се извършва синхронизацията и това е общият тактов сигнал на системата CLK. Друго много важно предим-

ство на създадената архитектура с един синхронизиращ сигнал е и фактът, че е без значение кога ще се появи фронтът на дадено събитие. Ако дадено събитие изпревари друго, това няма да наруши правилната работа на УБ, тъй като се проверява единствено нивото на сигнала.

- Използване само на един синхронизиращ сигнал;
- Използване само на разрешените за систем инструкции;
- Зададени са атрибути на входните и изходни пинове според изискванията от съседните модули и схеми;
- Използвана е наличната библиотека от примитиви в SYNOPSYS;
- Не са задавани ограничения по отношение на площта или бързодействието на схемата, както и ограничения в използването на библиотечните елементи.

Цифровият синтез се извършва в съответствие със зададените условия и алгоритмично описание. Електрическата схема, която се получава на изхода на синтезатора, може да бъде различна за всеки опит. В случая е важно да бъдат определени точно атрибутите на пиновете и условията на процеса, което гарантира работеща схема.

Предвидена е възможност за симулация на синтезираната електрическа схема с логически симулатор. Възможна е смесена симулация на модули, описани на ниво гейт, и такива с алгоритмично описание в една обща схема.

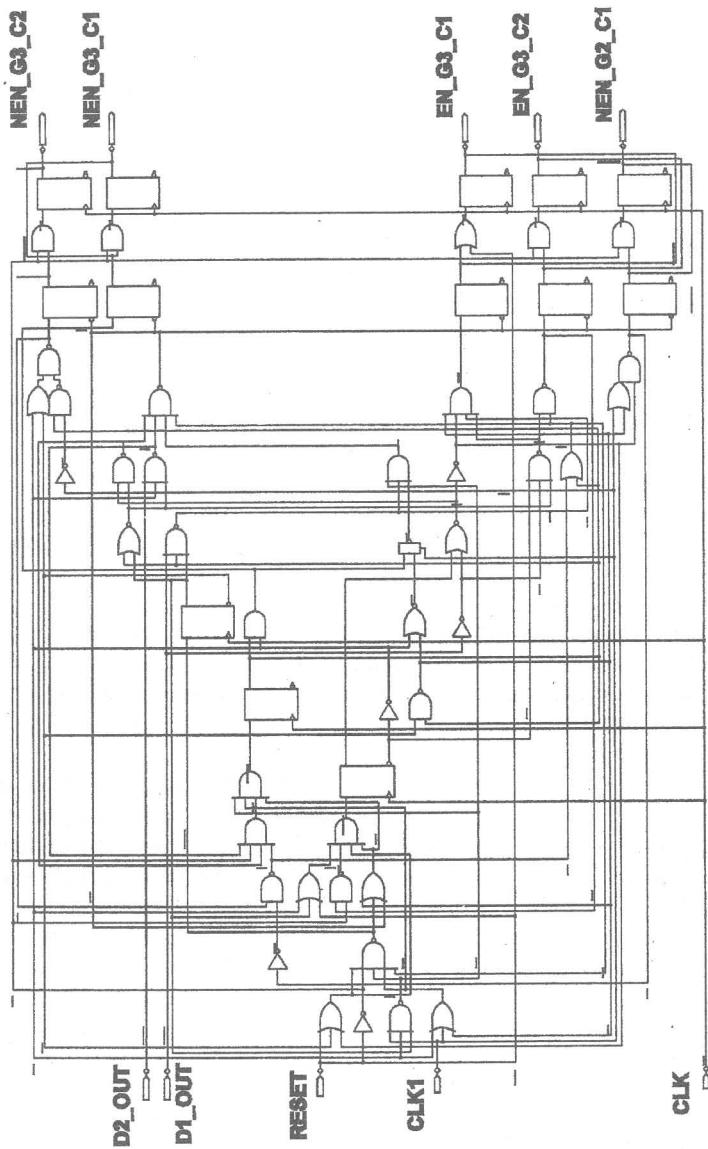
5. Заключение

Системата за автоматизация на инженерния труд SYNOPSYS разрешава гъвкаво, бързо и технически перфектно проектирането на специализирани интегрални схеми. Едновременно с възможността за пълен контрол от страна на проектанта се изисква и компетентност по отношение на техническите решения при проектирането като цяло. Процесът на синтез на управляващия блок е извършен в съответствие с всички изисквания към проектирането на цифрова схема - алгоритмично вярно и оптимизирано VHDL описание; коректно зададени условия за синтеза и ограниченията; цифров синтез с библиотека от цифрови примитиви; проверка за правилна работа на всяко ниво на проектиране.

Преценката на проектанта за качеството на синтезираната схема е основният фактор за крайната оценка на проекта. Разгледаният случай е пример за правилно използване на големите възможности на съвременните системи за автоматизация на проектирането.

6. Литература

1. Berge Y.M., A. Fonkona, S. Maginot. VHDL Designer's Reference, 1992, KLUWER ACADEMIC PUBLISHERS. London.
2. Lipsett R., C. Schaefer, C. Ussery. VHDL: Hardware description and design, 1993, KLUWER ACADEMIC PUBLISHERS. London.
3. Kurup P., P. Fernandes. Finite State Machine Synthesis, 1995, SYNOPSYS Methodology Notes



Фиг.7