

Разпределена система за управление на стъпков двигател

Димитър С. Тянев, Александър И. Иванов, Драгомир В. Янев, Стефка И. Попова

Резюме: Представено е едно ново апаратно-програмно решение на система за управление на стъпков двигател с разпределени функции и ресурси, който е изпълнителен орган в разрязващия агрегат на ролетно печатаща машина. Устройството представлява ниво в разпределена компютърна система за цялостно on-line управление на печатащата машина. Проектирането и реализацията на устройството е изпълнено със средства и интегрални елементи на фирма Xilinx. Управлението замества морално остаряла електромеханична система за разрязване, като постига изключително висока точност.

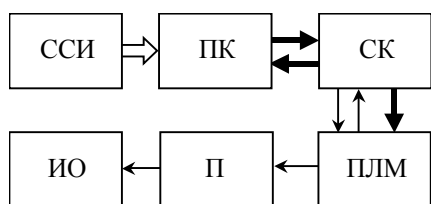
Distributed system for stepper motor control

Dimitar S. Tyanev, Aleksandar I. Ivanov, Dragomir V. Yanev, Stefka I. Popova

Abstract: In the document is presented the technical implementation of a control device for a stepper motor, which is the actuator in the slitting unit of a roll printing machine. The device is a part of a distributed computer system for online control of the printing machine. The design and implementation of the device is done using the tools and programmable devices of Xilinx. This type of control replaces the outdated electro-mechanical slitting control system, thus achieving higher level of precision.

1. Увод

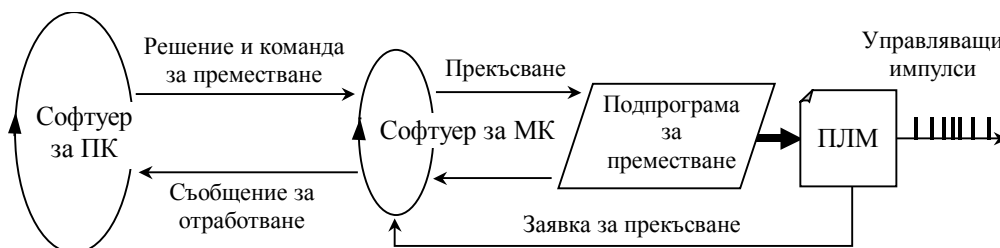
В настоящата работа се предлага едно ново апаратно-програмно решение на система за управление на стъпков двигател с разпределени функции и ресурси, характеризираща се с това, че реализира управляващия алгоритъм хардуерно, на базата на програмируема логическа матрица. Общата структура, в състава на която се намира подсистемата за управление на стъпковия двигател, е показана на фигура 1.



Фиг. 1 Общ вид на подсистемата

Тя съдържа система за сбор на информация (ССИ); персонален компютър (ПК); специализиран контролер (СК); програмируема логическа матрица (ПЛМ), в която е реализирано устройството за управление на стъпковия двигател (УУСД); преобразувател (П) и изпълнителен орган (ИО). Изпълнителният орган, който се има предвид, е двуфазен реверсивен стъпков двигател тип PK264A2-SG7.2, с предавателно отношение 7,2:1, който се задвижва чрез преобразувател тип CSD2120-T.

Общата схема на алгоритъма, по който функционира представената на фигура 1 система, е показана на фигура 2.



Фиг. 2 Общ вид на софтуера на подсистемата

Както ПК така и МК представляват многофункционални апаратно-програмни системи, в които задачата за управление на стъпковия двигател се решава при необходимост в режим на прекъсване.

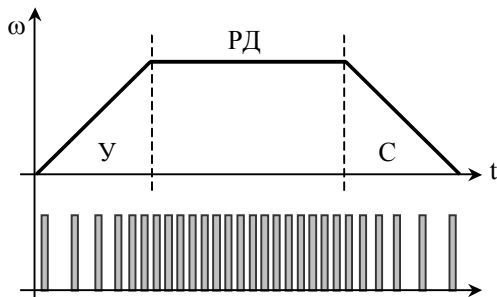
2. Синтез на управляващия алгоритъм

Тъй като бързодействието за отработване на отклонението в работния орган на системата е желано качество, за управлението на двигателя е избран линеен закон за скоростта ω , който съдържа етап на ускорение (У), етап на равномерно движение (РД) и етап на спиране (С):

Ъгловата скорост ω и честотата f на управляващите импулси са линейно зависими:

$$\omega = \alpha \cdot f, \quad (1)$$

където α е означен ъгълът на завъртане на ротора на двигателя. Тъй като ъгловата стъпка на завъртане s е константа ($s=0,25^\circ$), следва че при управление манипулираната величина е честотата. Реалният сигнал за управление представлява импулсната последователност, показана на фигура 3.



Фиг. 3 Закон на скоростта

Продължителността на правоъгълните импулси е постоянна и е ограничена от преобразувателя до $5[\mu s]$. При смяна на посоката на движение на двигателя, управлението следва да осигури след последния и преди първия управляващи импулси, празен интервал с минимална продължителност от $100[\mu s]$. Максималната скорост на въртене на двигателя е $250[Rpm]$. За постигане на тази скорост импулсите трябва да следват с минимален период, който определя максималната честота, както следва:

$$f_{max} = \frac{360^\circ}{s} \cdot 250 \cdot \frac{1}{60} = 6000 [Hz], \quad (2)$$

Така за периода се получава:

$$T_{min} = \frac{1}{6000} = 167 [\mu s]. \quad (3)$$

За да се постигне линейност при нарастване на скоростта, генерирането на управляващите импулси трябва да се извършва променливо закъснение. Всяко следващо закъснение трябва да е такова, че постиганата скорост на въртене на двигателя, да следва профила на ускорението възможно най-точно, както е показано на фигура 3. При достигане на максималната скорост честотата на управляващите импулси остава постоянна, а при спиране те следват един след друг с такива закъснения, с каквито съответно са се подавали по време на ускорителния етап, но в обратен ред.

2. Изчисляване на данните за управление

Времето закъснение δt , с което се появява всеки импулс, управлява скоростта, т.е. колкото по-малко е то, толкова по-голяма е скоростта. Движението на стъпковия двигател е образувано от дискретни стъпки, а зърнеността на това движение се определя от честотата на таймер. Параметрите на таймера и механичното движение са свързани чрез следната зависимост:

$$\delta t = c \cdot t_t = \frac{c}{f_t}, \quad (4)$$

където t_t и f_t са съответно периодът и честотата на таймера, а c е означен междустъпковият интервал в брой негови периоди. За да генерираме реално поредица от импулси е необходима базова честота, чрез която да изразим дължината на интервалите между стъпките. Този таймер е реализиран в програмируемата логическа матрица ПЛМ, поради което изчислението на елементите на масива от периоди за управляващите импулси, следва да се извърши спрямо честотата, на която работи таймерът – f_t .

За да се осъществи управлението на двигателя по зададения закон, е необходимо да се изчислят закъсненията на последователните управляващи импулси,

които като елементи на един едномерен масив ще представляват данновата основа, върху която устройството в ПЛМ ще генерира тези импулси. Преместването S , което извършва двигателят за времето t_n , за което достига максималната скорост, се определя според закона:

$$S = 0,5 \cdot \dot{\omega} \cdot t_n^2, \quad (5)$$

където $\dot{\omega}$ е означено допустимото за системата на двигателя максимално ускорение.

Изминатият път се изразява още така:

$$S = \alpha \cdot n, \quad (6)$$

т.е. това е ъгълът на завъртане при една стъпка умножен с броя на стъпките. Приравнявайки (5) и (6) определяме броя на стъпките за времето на ускорение t_n :

$$n = \frac{\dot{\omega} \cdot t_n^2}{2 \cdot \alpha}. \quad (7)$$

След като е известен броят на стъпките за достигане на максималната скорост, следва да се определят стойностите на последователните закъснения. Можем да изразим времето за извършване на k на брой стъпки по горната формула така:

$$t_k = \sqrt{\frac{2 \cdot k \cdot \alpha}{\dot{\omega}}}. \quad (8)$$

Интервалът между две последователни стъпки се изразява чрез формулата:

$$T_k = t_{k+1} - t_k = \sqrt{\frac{(k+1) \cdot 2 \cdot \alpha}{\dot{\omega}}} - \sqrt{\frac{k \cdot 2 \cdot \alpha}{\dot{\omega}}}. \quad (9)$$

За да намерим времето за изпълнение на първата стъпка, което се явява първият период T_0 на импулсната поредица, заместваем k с нула:

$$T_0 = \sqrt{\frac{2 \cdot \alpha}{\dot{\omega}}} - 0 = \sqrt{\frac{2 \cdot \alpha}{\dot{\omega}}}.$$

За определяне на всеки следващ период T_k , използваме общата формула, в която след полагане на частта, съответстваща на определеното по-горе за T_0 , получаваме следния вид:

$$T_k = T_0 \cdot (\sqrt{k+1} - \sqrt{k}). \quad (10)$$

Имайки предвид, че T_k е свързано, според (4), с междустъпковия интервал c , получаваме формулите за изчисление на тези интервали, съответно c_0 и c_k :

$$c_0 = \frac{1}{t_1} \sqrt{\frac{2 \cdot \alpha}{\dot{\omega}}} \quad \text{и} \quad c_k = c_0 \cdot (\sqrt{k+1} - \sqrt{k}). \quad (11)$$

Следва да изтъкнем, че между стъпковите интервали следва да се изчисляват с точност до единица, тъй като имат характера на цели числа – те представляват дължината на интервалите в брой цикли на таймера. Изчислението на поредицата коефициенти c_k според получената формула е много бавна процедура, тъй като това се налага да се извърши програмно в архитектурно слабата операционна част на микроконтролера. Ето защо за това изчисление се търси друга по-подходяща форма. Чрез апроксимация на елементарната функция корен втори в (11) с ред на Тейлор, стойността на коефициентите може да бъде получена чрез следната рекурентна формула [1, 2]:

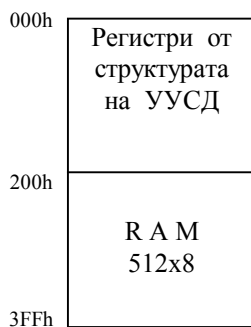
$$c_k = c_{k-1} - \frac{2 \cdot c_{k-1}}{4 \cdot k + 1}. \quad (12)$$

Това изчисление е многократно по-бързо от изчислението (11) с двата квадратни корена, но въвежда грешка 0,44 при $k=1$, която компенсираме като умножаваме c_0 с константата 0,676.

Разполагайки с броя n на стъпките за ускорение, с изчислен първи елемент c_0 и с формулата за всеки следващ елемент c_k , профилът на функцията за управление от фигура 3 е напълно определен.

3. Програмен модел на УУСД

За да поясним кратко алгоритъма за управление и структурата на устройството за управление на стъпковия двигател ще представим програмния модел, който “вижда” микроконтролерът. Последният е свързан с УУСД с 10-битова адресна и 8-битова даннова шини. Така в адресното пространство от 1[КВ] програмният модел заема част от следните области, показани на фигура 4:



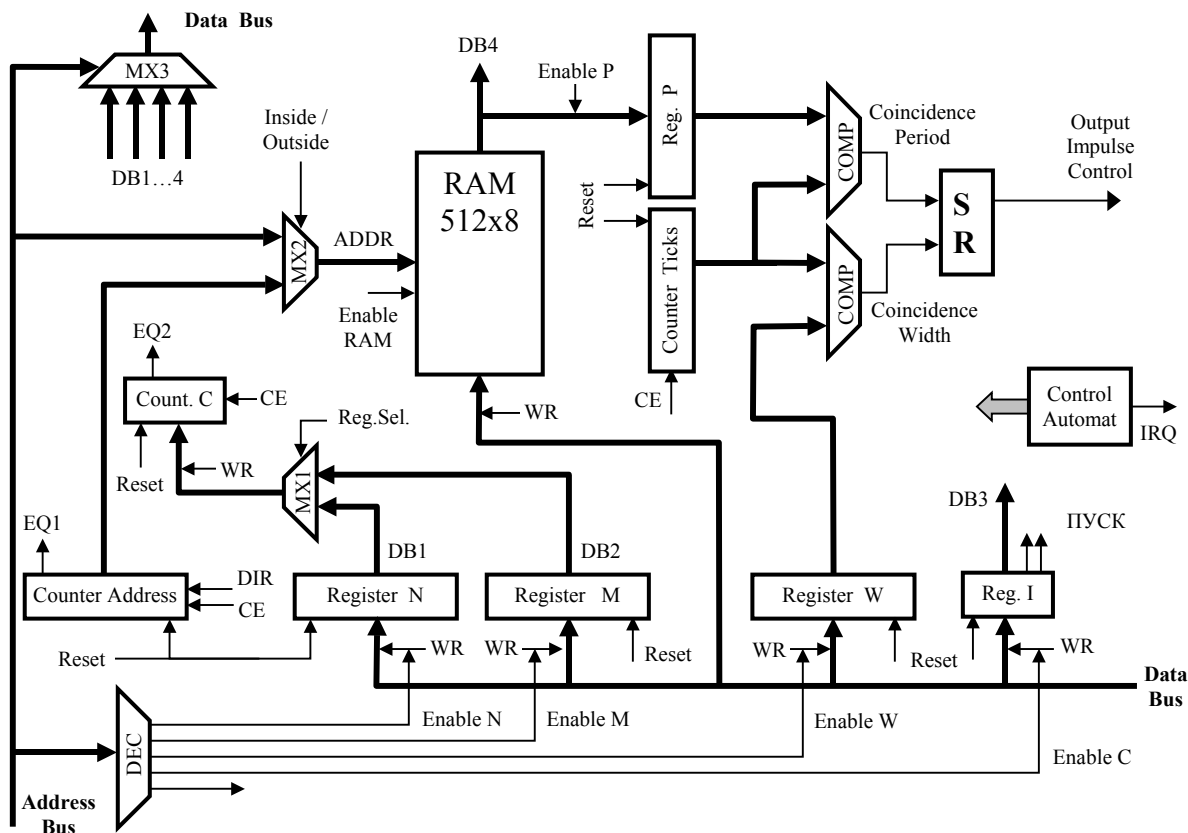
Фиг. 4 Програмен модел на УУСД

Във втората половина на адресното пространство е разположена RAM-памятта на УУСД, в която 512 клетки са предназначени за съхранение на еднобайтовите елементи на масива от стъпковите интервали c_k , изчислени от микроконтролера СК, според формулите (11) и (12). Масивът със стъпковите интервали обаче се записва в тази част на паметта в минимизиран обем по следния начин – реално записаните елементи представляват стъпковите интервали само за етапа У на закона за управление (вижте фигура 3). Това е възможно, защото в етапа

С броят на стъпковите интервали е същият, както в етапа У, а стъпковите интервали в етапа РД съвпадат със стойността на последния стъпков интервал от етапа У.

В първата половина на адресното пространство са разположени операционните регистри на логическата структура на УУСД. Започвайки от адрес 000h, техния ред е следния: регистър на командата (Reg.I) с дължина 1[B]; регистър за брой на стъпките при ускорение Reg.N с дължина 2[B]; регистър за брой на стъпките при равномерно движение Reg.M с дължина 4[B]; регистър за фактор на деление Reg.FD с дължина 1[B]; регистър за ширина на импулса Reg.W с дължина 1[B]; контролен регистър Reg.C с дължина 4[B]. Регистрите Reg.FD и Reg.C не са показани в логическата структура, тъй като тяхното предназначение е да осигурят универсалност на устройството за управление, по отношение на техническите параметри на различни от споменатия тип стъпкови двигатели.

Логическата структура на реализираното УУСД е представена на фигура 5.



Фиг. 5 Логическа структура на УУСД

4. Алгоритъм за управление

При включване на захранването системата се установява в изходно състояние. В това състояние адресната шина на RAM-паметта е включена чрез MX2 към адресната шина на микроконтролера. Така регистрите и паметта в УУСД са видими за микроконтролера, който го инициализира със следните константи:

- Фактор на деление. Това е цяло 8-битово число, което характеризира времевите параметри на използвания тип стъпков двигател. Съдържа се в Reg.FD ;
- Ширина на управляващия импулс, според изискванията на използвания тип стъпков двигател. Представлява цяло 8-битово число в регистър Reg.W.

Посочените константи се записват в УУСД еднократно от микроконтролера.

Нов сеанс за управление на стъпковия двигател започва по инициатива на по-високото ниво в разпределената система, т.е. със запис на всички необходими нови данни в регистрите на УУСД и в RAM-паметта. Подготвителната процедура изпълнява записите в следната последователност:

1. Запис в регистър Reg.N на броя стъпки в етапа за ускорение (цяло 2-байтово число) ;
2. Запис в регистър Reg.M на броя стъпки в етапа за равномерно движение (цяло 4-байтово число) ;
3. Запис в RAM-паметта на едномерния масив от еднобайтови цели числа, съответстващи на междустъпкови интервали c_k , изчислени от микроконтролера ;
4. Запис в регистъра на командата Reg.I на флага за посока на движението (бит №1) ;
5. Реализация на пауза при смяна на посоката на движение на стъпковия двигател, която е с продължителност не по-малка от 100[μ s] за посочения тип стъпков двигател (СД).
6. Запис в регистъра на командата Reg.I на флага "ПУСК" (бит №0) .

С последния запис на данни в УУСД от подготвителната процедура вътрешния управляващ автомат СА стартира поредния сеанс за управление на стъпковия двигател.

Флагът "ПУСК", който се появява в регистъра на командата Reg.I, разрешава изхода "Output Impulse Control" и извежда управляващия автомат СА от изходно състояние. Алгоритъмът за управление минава през следните етапи: етап У, етап РД, етап С (вижте фигура 3). Във всеки един от тези етапи се генерират стъпкови интервали с по един управляващ импулс. Генерирането на стъпковия интервал (период) започва с генерирането на единичния управляващ импулс. Генерирането на стъпковия интервал представлява генериране на продължителността на правата (единичната) фаза, след което следва генериране на продължителността на инверсната фаза. Така двете фази формират цялостно управляващия такт. Продължителността на времевите интервали се измерва с периода на основния тактов генератор. Импулсите на този тактов генератор модифицират съдържанието на всички броячи, които измерват времеви интервали.

Генерирането на правата фаза се осъществява от изходния RS-тригер. Управлението на този тригер се осъществява от сигналите "Coincidence Period" (CP) и "Coincidence Width" (CW). Тези сигнали се получават в резултат на сравнение на съдържанието на таймера Counter Ticks, измерващ времето в текущия период, с двете гранични стойности: продължителност на управляващия сигнал, намираща се в Reg.W и продължителността на текущия период, намираща се в Reg.P. За всеки период е в сила отношението (Reg.W) < (Reg.P). Тъй като изходното състояние на RS-тригера в началото на всеки период е единица, тази единица се поддържа до момента, в който се появи сигналът "CW", т.е. до постигане на равенството (Counter Ticks) = (Reg. W). С появата на сигнала "CW" RS-тригерът се превключва в нула. Този сигнал е с продължителност равна на

един период на времевия тактов генератор. Така започва генерирането на инверсната фаза в текущия период. Тя приключва когато съдържанието на брояча Counter Ticks се изравни със съдържанието на Reg.P. В този момент, когато (Counter Ticks) = (Reg.P) възниква сигналът "CP". Този сигнал бележи края на текущия период и същевременно началото на следващия период. Продължителността на този сигнал също е равна на един период на времевия тактов генератор. Сигналът "CP" се използва и за нулиране на брояча Counter Ticks.

Генерирането на двете фази на следващия период следва да започне при ново съдържание на регистъра на продължителността на периода Reg.P. Новото съдържание е подготвено (прочетено) от RAM-паметта предварително, по време на вече завършилия период. Това е постигнато чрез модификация на адресния брояч Counter Address. Модификацията на този брояч е в положителна посока по време на етапа У, липсва по време на етапа РД, и е в отрицателна посока по време на етапа С от закона за управление. Записването на продължителността на новия период в Reg.P се извършва автоматично от управляващия автомат СА, с което започва новото генериране. Ако генерираният период е последен, управляващият автомат генерира сигнала за край (сигнал за прекъсване) "IRQ".

Моментите за модификация на режимите на адресния брояч се съобщават на управляващия автомат чрез измерване на продължителността на отделните етапи за управление на стъпковия двигател – У, РД и С. Самото измерване става с помощта на брояча на цикли Counter С и дешифратора на неговото нулево съдържание, който формира сигнала "EQ2". Всяко начално съдържание на този брояч се зарежда през мултиплексор MX1 от регистър Reg.N или от регистър Reg.M. Този мултиплексор се управлява от сигнала "Register Select".

След генерирането и на последния управляващ импулс, състоянието на

структурата на УУСД е следното: адресният брояч Counter Address се намира в изходно състояние (сочи клетката, в която се намира първият елемент от масива); броячът на цикли Counter С е нулиран; таймерът Counter Ticks е нулиран и RS-тригерът е в единично състояние; Управляващият автомат СА е завършил алгоритъма за управление и се намира в изходно състояние, при което MX2 е превключен към входната адресна шина, регистърът на командата Reg.I е нулиран; Reg.N, Reg.M и Reg.P запазват старото си съдържание.

Логическата структура на УУСД е проектирана така, че чрез мултиплексора MX3 могат да бъдат прочитани с цел контрол съдържанията на част от регистрите от структурата на УУСД.

4. Заключение

Описаното устройство за управление на стъпков двигател е проектирано в технологичната среда WebPack ISE на фирма Xilinx, чрез HDL-езика Verilog. Реализацията му е в FPGA-матрица от фамилията Spartan II на същата фирма. Устройството за управление на стъпковия двигател е част от структурата на специализирана компютърна система, която е самостоятелна част от разпределената система, показана на фигура 1. Тази част от разпределената система е предназначена за управление на разрязващия агрегат на ролетни печатащи машини. Реализираното on-line управление на разрязващия агрегат, в сравнение с традиционно използваната електромеханична система, позволява да се постигне изключително висока точност, оценявана с отклонения в интервала ± 0.2 [мм].

5. Литература

- [1]. AVR – *Linear speed control of stepper motor*, Atmel 8-bit Microcontrollers Application Note, Rev. 8017A-AVR-06/2006.
- [2]. Austn, D., *Generate stepper-motor speed profiles in real time*, Embedded System Programming, January 2005.

- [3]. Тянев Д. С., *Организация на компютъра – проектиране на логически структури*, ТУ-Варна, ISBN 954-20-0259-9, 2004.
- [4]. ISE 8.1i – *Libraries Guide*, Xilinx.
- [5]. Banskер J., *Verilog HDL Synthesis a Practical Primer*, Star Galaxy, 1998.

За контакти:
доц. д-р инж. Димитър Тянев
катедра “Компютърни науки и технологии”
Технически университет – Варна
WEB: www.tyanev.com