FRI-ONLINE-1-CCT1-13

WEB BASED LEARNING TOOL OF CYCLIC CODE ENCODING[13]

Yuksel Aliev, PhD Student
Department of Computing,
University of Ruse "Angel Kanchev", Bulgaria
E-mail: yaliev@uni-ruse.bg

Galina Ivanova, Assoc. Prof. PhD
Department of Computing,
University of Ruse "Angel Kanchev", Bulgaria
Phone: 086-821 521
E-mail: giivanova@uni-ruse.bg

Abstract: *This paper presents a developed web-based interactive tool for studying of cyclic code encoding. The purpose of the application is to help students to learn the complex processes of coding in cyclic code. The application represents two approaches for coding in cyclic code - polynomial and LFSR (Linear-feedback shift register). The maximum binary combination that can be encoded with the application is 26 bits, and students can detect double or triple errors. An important component of the application is the interactive algorithm (block diagram), which follow the process step by step and notifies in case of wrong actions. Statistical information is collected for each solved task, which is used to assess the work and the progress of the student. The developed application is part of the virtual library of the discipline "Reliability and diagnostics of computer systems", which is studied by students in the department "Computer Systems Technology" at the University of Ruse "Angel Kanchev". At the end of the report the results of the conducted survey for the opinion of the students about the developed application are presented and discussed.*

Keywords: *cyclic code, LSFR, polynomial, error detecting, error correcting, learning tool, interactive model, JavaScript, konvaJS, html5*

JEL Codes: I23, D83, L86

ВЪВЕДЕНИЕ

Решаването на задачи чрез класическия начин с лист и химикал при съвременните студенти, родени в ерата на цифровите технологии, става все по-нежелан и отегчителен начин на обучение. Това води до трудности при усвояване на преподавания материал. Интерактивните уеб базираните обучителни инструменти набират все по-голяма популярност в днешно време. Използването на подобни технологии в учебния процес в областта на теорията на кодирането е необходим и полезен предвид сложните математически операции. Друго важно предимство на интерактивните образователни инструменти е, че те улесняват самоподготовката на студента, в случаите когато програмният продукт следи работата, подсказва и уведомява при грешка по време на решаване на задачите.

С развитието на съвременните информационни и комуникационни технологии се стигна до изключителното повишаване на изискванията към надеждността на съхраняваната и предаваната информация. Поради което изучаването им от студентите бъдещи компютърни инженери е важно.

Съществуват редица методи за шумоустойчиво кодиране, които намират широко приложение в съвременните комуникационни и компютърни системи. Двоичните циклични кодове са важен подклас на линейните блокови кодове (Sklar, B., 2001). Основното им свойство е, че всяка двоична комбинация, получена чрез цикличен код, след преместване на разрядите на разрешена кодова комбинация пак е разрешена кодова комбинация. Характерното за тях, е че те лесно могат да бъдат реализирани апаратно чрез използване на линеен изместващ регистър с обратна връзка (LFSR – Linear Feedback Shift Register).

[13] Докладът е представен на заседание на секция 3.2 на 29 октомври 2021 с оригинално заглавие WEB BASED LEARNING TOOL OF CYCLIC CODE ENCODING

ИЗЛОЖЕНИЕ

Общи сведения за цикличните кодове

При построяването на цикличните кодове е възприето кодовата комбинация да се представя чрез полином спрямо някаква променлива, най-често отбелязвана с $\boldsymbol{x}$. Ако за представяне на полинома на информационна комбинация се използва $G(x)$, той ще има следният общ вид (Chawla, G., Chaudhary ,V., 2014):

$$G(x) = a_{n-1}.x^{n-1} + a_{n-2}.x^{n-2} + \cdots + a_1.x + a_0 , \quad (1)$$

където $a_{n-1}, a_{n-2}, a_1, a_0$ са коефициенти, приемащи стойности 1 или 0. Примерно, ако (1) са приложи върху двоичната комбинация 101001101, ще се получи $G(x) = x^8 + x^6 + x^3 + x^2 + 1$.

Един от широко използваните начини за представяне на полинома на закодираната кодова комбинация $F(x)$ при цикличните кодове има следния общ вид:

$$F(x) = x^k.G(x) + R(x) , \quad (2)$$

където:

$G(x)$ – полином на информационната комбинация, степента на която е $\leq (m-1)$;

$R(x)$ – остатък от делението на $x^k.G(x)$ на $P(x)$, като $P(x)$ е пораждащ полином от степен $k = n - m$, подбиран по специален начин.

Важен момент при кодирането в цикличен код е изборът на пораждащ полином $P(x)$, чрез който се обуславя шумоустойчивостта на кода. Критериите за избор на $P(x)$ са следните:

- степента на $P(x)$ трябва да е равен на броя контролните битове *k;*
- пораждащият полином $P(x)$ трябва да е прост, неприводим, т.е. да се дели без остатък само на единица и на себе си.
- За цикличен код с $d = 4$, пораждащия полином се образува по следния начин: $P(x) = (1 + x).P_1(x)$. Полиномът $P_1(x)$ трябва да отговаря на предните два критерия. В този случай на практика контролните битове се увеличават с единица.

Примерна задача за кодиране в цикличен код чрез полиномно деление

Нека да се извърши кодиране в цикличен код, откриващ всички двойни грешки за двоичната комбинация: $\mathbf{X} = \mathbf{1010101101}$. Първата стъпка е определяне параметрите на кода:

- Броят на информационните битове $m = 10$;
- Броят на грешките, които кода може да открива $l_0 = 2$;
- За кодовото (Хемингово) разстояние се изчислява $d = l_0 + 1 = 2$;
- За броят на контролните битове от условието $2^k - k = m + 1$ се получава $k = 4$

Следва образуване на информационния полином $G(x)$ и съответно умножението му с x^k, както е показано по-долу.

$$G(x) = x^9 + x^7 + x^5 + x^3 + x^2 + 1 , \quad (3)$$

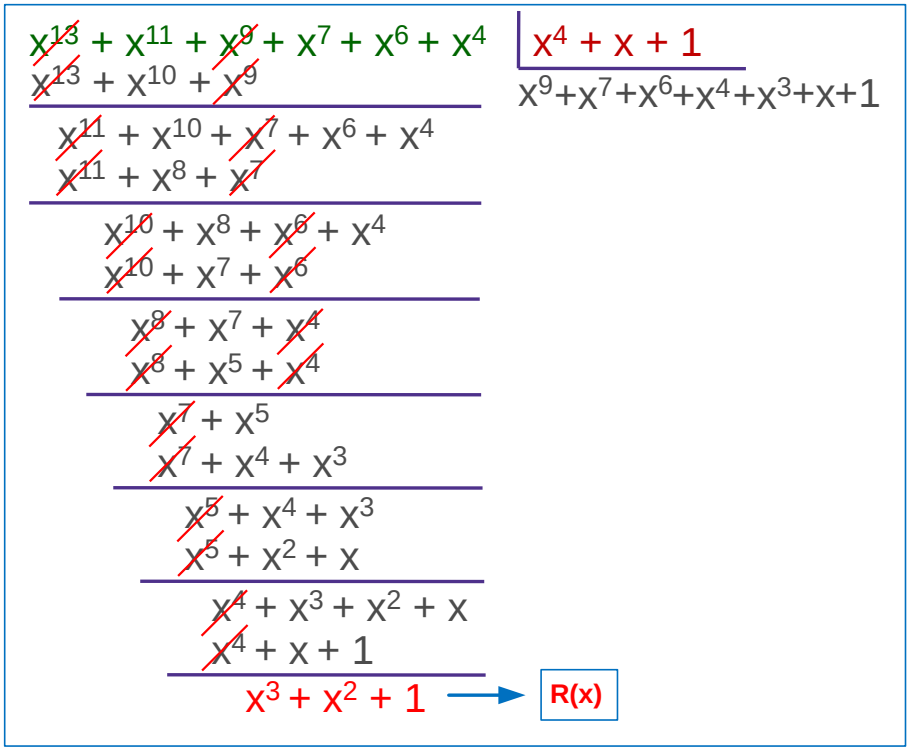
$$x^k.G(x) = x^{13} + x^{11} + x^9 + x^7 + x^6 + x^4 \quad (4)$$

Следващата стъпка изборът на пораждащия полином $P(x)$. На критериите за избор на пораждащ полином за $k = 4$ се получават два полинома, между които трябва да се направи избор.

$$P(x) = x^4 + x + 1 \text{ или } P(x) = x^4 + x^3 + 1 \quad (5)$$

Нека за пораждащ полином да е избран $\boldsymbol{P(x) = x^4 + x + 1}$.

Следва основната стъпка, а именно полиномното деление $\frac{x^k.G(x)}{P(x)}$, показано на фиг. 1.



Фиг. 1. Деление на полиномите $\frac{x^k.G(x)}{P(x)}$

След делението на полиномите се получава полиномът на остатъка $R(x)$.

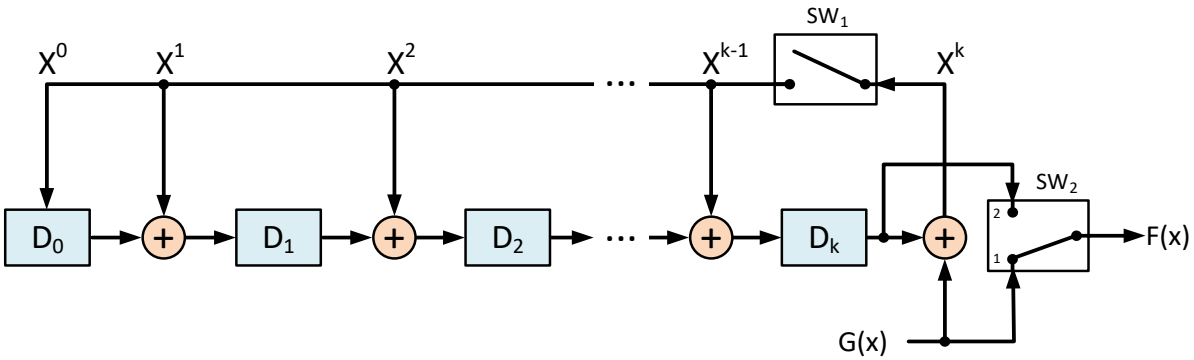
$$R(x) = x^3 + x^2 + x + 1. \tag{6}$$

След заместване в (2) за полинома на закодираната кодова комбинация се получава:

$$\begin{aligned} F(x) &= x^k.G(x) + R(x) = \\ &= x^{13} + x^{11} + x^9 + x^7 + x^6 + x^4 + \boldsymbol{x^3} + \boldsymbol{x^2} + \boldsymbol{x} + \mathbf{1}. \end{aligned} \tag{7}$$

В крайна сметка след преобразуване от полинома в двоична форма за кодовата комбинация се получава: $[X] = 1010101101\,\mathbf{1111}$, в която последните $k = 4$ бита са контролни.

Апаратната реализация на цикличен кодер може да се реализира чрез използване на изместващ регистър с линейна обратна връзка (LSFR - Linear-Feedback Shift Register), показан на фиг. 2. На фигурата е даден общият вид на схемата, като конкретната схема се определя от пораждащия полином $P(x)$.



Фиг. 2. Общ вид на цикличен кодер, реализиран с LFSR

Разработване на интерактивни обучителни модели на цикличен кодер

Разработени са два авторски уеб базирани модела (инструмента) за изучаване на процеса на кодиране при цикличните кодове. Първият модел представя полиномния начин на

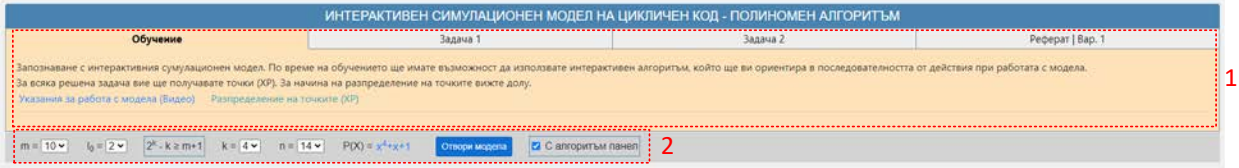
кодиране, а втория – апаратния начин чрез използване на линеен изместващ регистър с обратна връзка (LFSR). И двата модела са предвидени да бъдат интегрирани във виртуалната библиотека по дисциплината „Надеждност и диагностика на компютърни системи“. Важна функционална възможност на виртуална библиотека е, че за всяка решена задача се пази статистическа информация, която по-късно ще служи за осигуряване на обобщена информация за работата на студента, както и за получаване на информация за използваемостта на всеки модел и допуснатите грешки.

За разработка на интерактивните програмните модели са използвани езиците *HTML5* и *JavaScript* (Mackie, S., 2018). За създаване на отделните компоненти са използвани JavaScript библиотеките *KonvaJS* (KonvaJS, 2020,) и *JQuery* (JQuery, 2020).

Потребителските интерфейси и на двата програмни модели са реализиран на два езика - български (BG) и английски (EN). По-долу се представят двата модела.

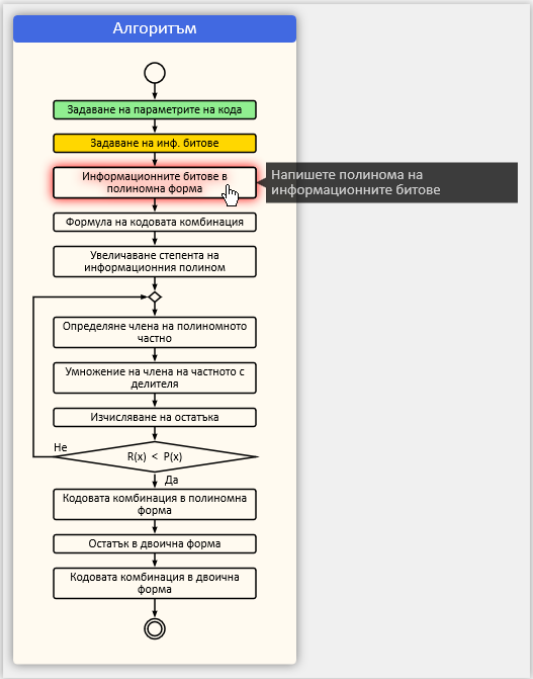
Интерактивен модел за кодиране в цикличен код чрез полиномен метод

Първоначалният изглед на модела при отваряне е показан на фиг.3. Интерфейсът се състои от две части. От първата част (фиг. 3 под номер 1) се избира режима на работа с модела – „*Обучение*“ или „*Задача*“.



Фиг. 3. Начален екран на интерактивния модела

В режим „***Обучение***“ студентът се запознава с интерактивния модел. За улеснение се предоставя и видео инструкция. При този режим се разрешава използването на т.нар. интерактивен алгоритъм (блок-схема), който предоставя цялостна информация за стъпките на работа (микрооперации) и пояснения за всяка отделна стъпка при посочването й с мишка (фиг. 4).

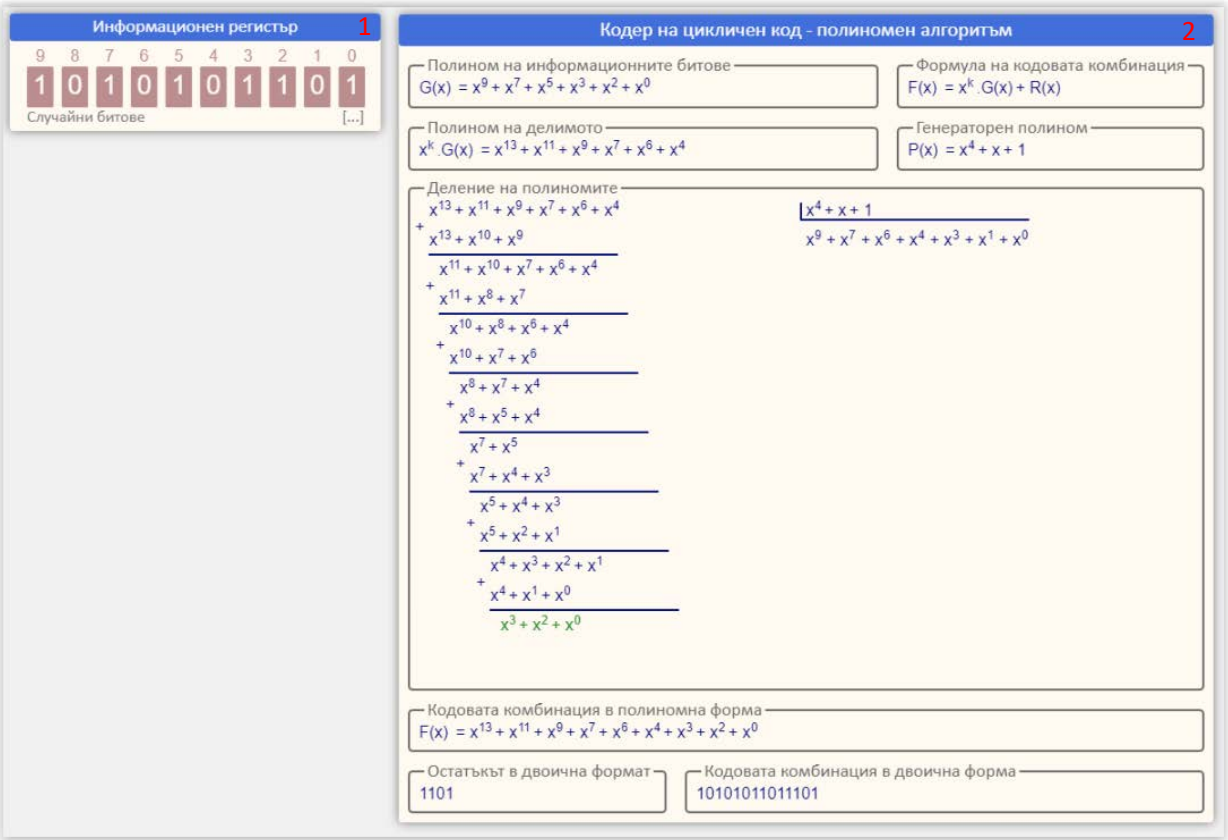


Фиг. 4. Интерактивен алгоритъм при полиномния метод на кодиране

В режим „***Задача***“ студентите решават конкретна задача по конкетно задание. Като при този режим интерактивният алгоритъм е забранен и не може да бъде визуализиран.

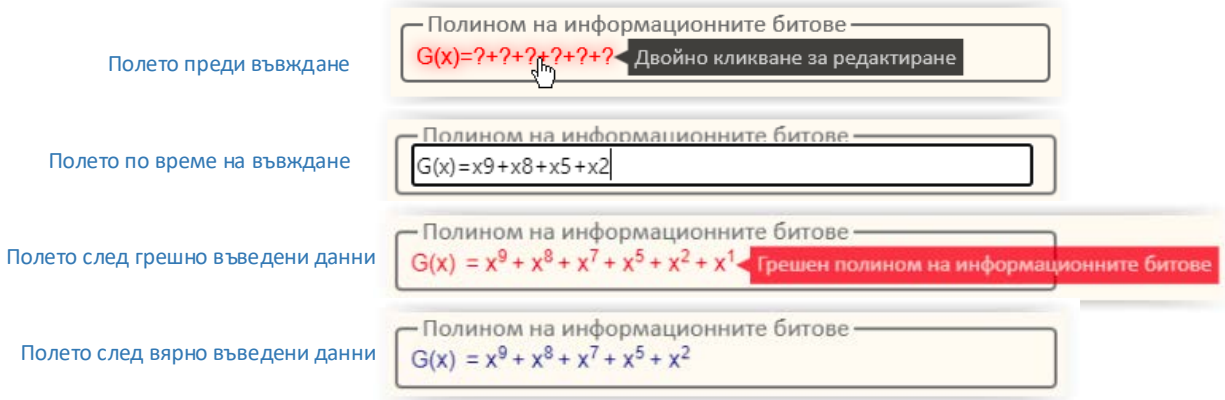
Във фиг. 3 под номер 2 се вижда панела за задаване на параметрите на кода, които са: брой на информационните битове ***m***, брой на контролните битове ***k*** и генераторния

(пораждащия) полином $P(x)$. След правилно зададени параметри, чрез бутона „Отвори модела“ се отварят няколко панела, които представляват същинската част на модела, показани на фиг. 5.



Фиг. 5. Основните панели на цикличния кодер за полиномен метод

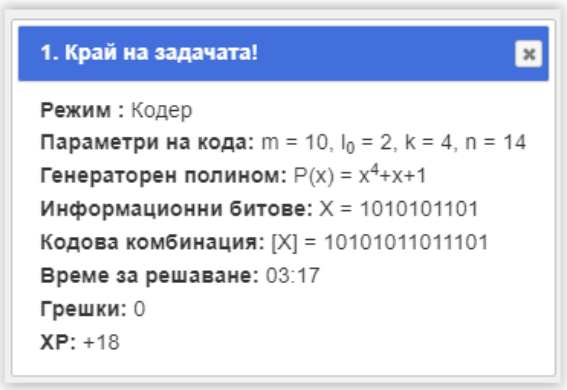
Панелът „Информационен регистър“(фиг. 5 под номер 1) служи за въвеждане на битовете на информационната комбинация, като това може да стане по няколко начина – чрез бутона „Случайни битове“, чрез кликане върху всеки един бит и чрез въвеждане в текстово поле от бутона „[…]“. Панелът „Кодер на цикличен код – полиномен алгоритъм“ е главният панел, в който се извършват различните полиномни преобразувания и операции според алгоритъма от фиг. 4. Въвеждането на полиномите става чрез двойно кликване върху съответното поле, като етапите на въвеждането са показани на фиг. 6.



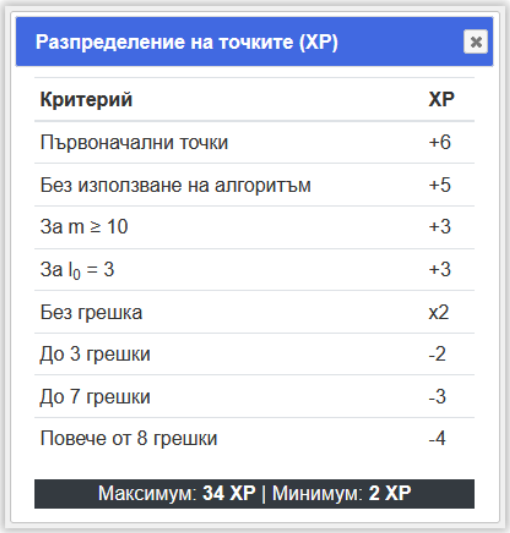
Фиг. 6. Въвеждане на информационния полином

След приключване на всички стъпки по алгоритъма се извежда съобщение за край с обобщена информация за решената задача, както е показано на фиг. 7. Част от данните от съобщението са запазват в базата данни на виртуалната библиотека, с цел по-нататъшна статистика за модела и студента. Последният ред от съобщението се отнася за брой получени

точки (XP - experience point) от задачата, като целта на тази информация е съставяне на класация с най-добрите студенти. Условията за получаване на точките от всяка решена задача се определя по таблицата, показана на фиг. 8.



Фиг. 7. Финално съобщение в края на задачата



Разпределение на точките (XP)

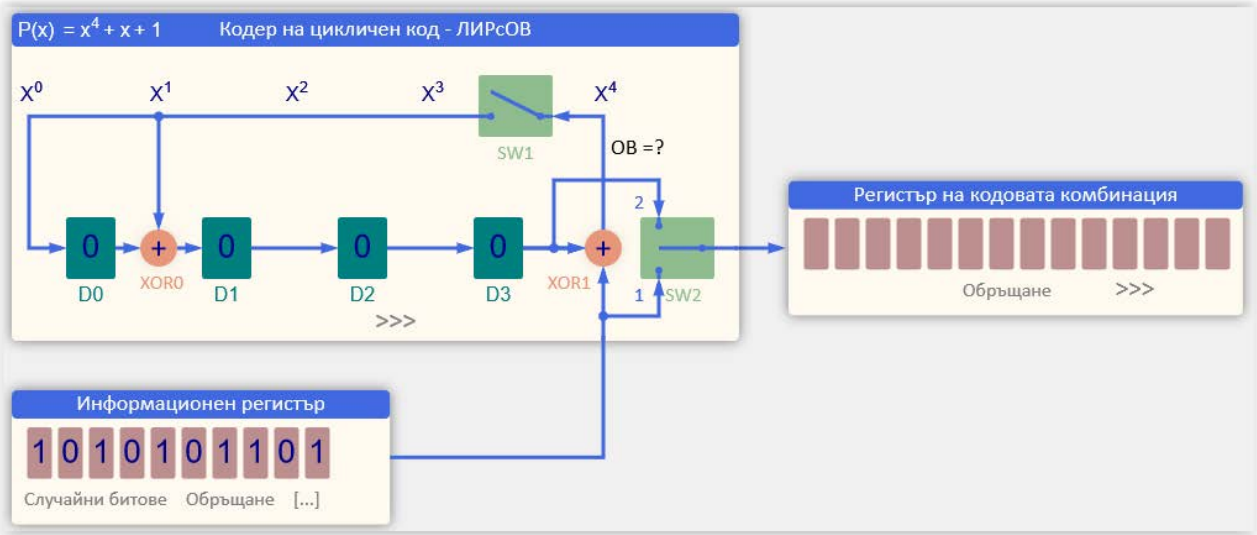
Критерий	XP
Първоначални точки	+6
Без използване на алгоритъм	+5
За m ≥ 10	+3
За l_0 = 3	+3
Без грешка	x2
До 3 грешки	-2
До 7 грешки	-3
Повече от 8 грешки	-4

Максимум: **34 XP** | Минимум: **2 XP**

Фиг. 8. Условие за получаване на точки (XP) от решена задача

Интерактивен модел за кодиране в цикличен код чрез линеен изместващ регистър с ОВ (LFSR)

Този интерактивен програмен модел е базиран на схемата дадена на фиг. 2 - общ вид на цикличен кодер, реализиран чрез LFSR. Първоначалният екран на модела е еднакъв като този на полиномния модел (фиг. 3) и всичко казано за него важи и за този. Общият вид на модела, съответстващ на пораждащ полином $P(x) = x^4 + x + 1$ е показан на фиг. 9. Състои се от три панела – „*Кодер на цикличен код - ЛИРсОВ*“, „*Информационен регистър*“ и „*Регистър на кодовата комбинация*“. В информационния регистър се зареждат битовете на информационната комбинация. В него има една допълнителна функционалност, наречена „*Обръщане*“, която служи за огледално обръщане (flip) на стойностите на регистъра.



Фиг. 9. Цикличен кодер базиран на линеен изместващ регистър с ОВ, съответстващ на пораждащ полином $P(x) = x^4 + x + 1$

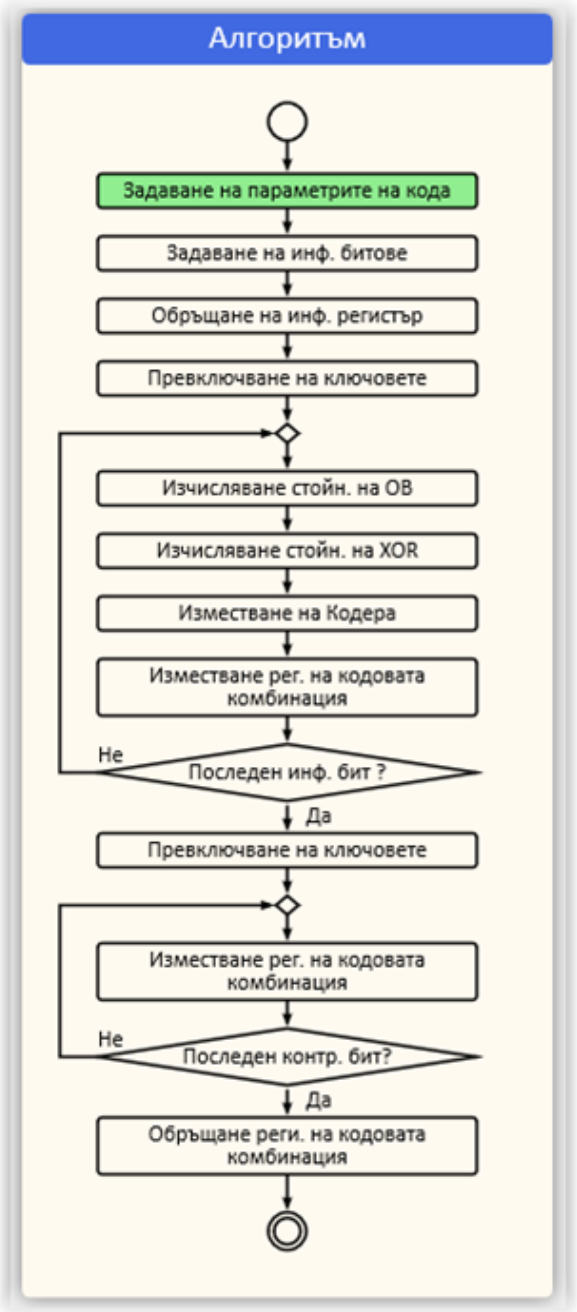
В панела „*Кодер на цикличен код - ЛИРсОВ*“ се изчисляват контролните битове на кодовата комбинация. Той включва няколко интерактивни елемента: ключове, отбелязани с

SW1 и SW2; суматори по модул 2, отбелязани с XOR и бутон за изместване на дясно отбелязан с “>>>”. Броят на XOR елементите се определя според пораждащия полином. Първоначалното състояние на ключовете е по следния начин: SW1 е затворен, а SW2 е на позиция 1.

В регистъра на кодовата комбинация се получават битовете на закодираната кодова комбинация. Този регистър за разлика от информационния регистър притежава функционалност за изместване на дясно, отбелязана с бутона „>>>“.

Алгоритъмът на модела е показан на фиг. 10. В него се забелязват два цикъла на работа. В първия цикъл след *m*-итерации в разрядите на изместващия регистър с линейна обратна връзка се получават контролните битове. След това ключовете се превключват: SW1 се отваря, а SW2 се превключва на позиция 2. После във втория цикъл след *k*-итерации се извършва прехвърляне на контролните битове в регистъра на кодовата комбинация.

В края на задачата, както при предходния интерактивен модел, се извежда съобщение, изглеждащо по същия начин като това на фиг. 7 и статистическата информация се регистрира в базата данни на виртуалната библиотека.



Фиг. 10 Алгоритъм на работа на цикличния кодер чрез LFSR

ЗАКЛЮЧЕНИЕ

Разработените уеб базирани интерактивни програмни модели намериха успешно приложение в учебния процес по дисциплината „Надеждност и диагностика на компютърните системи“. В процеса на работа с моделите бяха открити грешки, които в последствие бяха отстранени.

Беше проведено анкетно проучване с 38 студентите относно удовлетвореността им от всеки един модел. Те оцениха моделите по 8 различни характеристики чрез пет степенна ликертова скала (*силно недоволен, доволен, неутрален, доволен и много доволен*). Характеристиките, по които оцениха моделите са:

- Дизайн на интерфейса;
- Леснота за ориентация;
- Бързодействие на сайта;
- Ръководство за решаване на задачите;
- Описание на математическия алгоритъм;
- Практикуване и затвърждаване на умения за решаване на задачите;
- Статистиката с точки за модела;
- XP точките за поощрение.

На фиг. 12 са показани резултатите от анкетното проучване на интерактивния цикличен кодер за полиномния метод, а на фиг. 13 – за цикличен кодер чрез ЛИРсОВ.

Фиг. 11. Резултати от анкетно проучване за модела на цикличен кодер – полиномен метод

Фиг. 12. Резултати от анкетно проучване за модела на цикличен кодер – чрез ЛИРсОВ

От резултатите на анкетното проучване и за двата интерактивни модела се забелязва, че почти за всички характеристики преобладава най-високата оценка „силно доволен“ (70-80%), следвана от следващата положителна оценка „доволен“ (20-30%). Негативните оценки са пренебрежимо малки (0-3%) за всички характеристики.

В заключение може да се обобщи, че представените интерактивни програмни модели са полезен образователен инструмент за използване в дистанционна форма на обучение. Те са лесно достъпни в уеб среда и позволяват на студентите да се упражняват неограничен брой пъти. Анализът на активността на студентите показа, че някои от тях са използвали възможността да се упражняват над 30 пъти с даден модел, което е довело до много високи резултати в общото им представяне. Студентите намират предложените интерактивни модели за забавни и много полезни в усвояването на учебния материал по дисциплината.

БЛАГОДАРНОСТИ

Настоящото изследване е подкрепено от проект № 2021-РУ-01 „Проектиране и изграждане на смарт учебно-изследователска лаборатория за обучение на докторанти – Фаза 2“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

Chawla, G., Chaudhary ,V. (2014). "*FPGA Implementation of Cyclic Code Encoder and Decoder*." Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 4, pp. 273-278

Blahut, R. E. (1983). *Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley. ISBN: 978-0-470-84356-7

Sklar, B. (2001). *Digital Communications: Fundamentals and Applications.* Second Edition, Upper Saddle River, NJ: Prentice-Hall.

KonvaJS, (2020). *HTML5 2d canvas js library for desktop and mobile applications*. URL:

KonvaJS (2020) - https://konvajs.org/ (Accessed on 20.02.2020).

JQuery, (2020). *JavaScript library*. URL: https://jquery.com/ (Accessed on 20.02.2020).

Mackie, S., (2018). *JavaScript: Best Practice*. SitePoint Pty. Ltd.

Gherman, V., Evain, S., Seymour, N., & Bonhomme, Y. (2011). *Generalized parity-check matrices for SEC-DED codes with fixed parity*. In 2011 IEEE 17th International On-Line Testing Symposium (pp. 198-201). IEEE.

Moon, T. K. (2005). *Error correction coding: mathematical methods and algorithms*. John Wiley & Sons.