Interactive Simulation Model of Hamming Codes for Training Purposes

Yuksel Aliev[1], Adriana Borodzhieva[2], Galina Ivanova[3]

Abstract

Course materials, practical exercises, simulations and assessments are increasingly online. Students have the flexibility to study at any time, and any place. Intelligent design of digital tools and resources can help the students become experienced and autonomous learners. Academic staff may also be able to help the students with core learning skills such as managing their time, taking and organizing notes, planning written assignments. Lecturers and students describe their experience on how to use the most useful teaching and learning technologies. The domains where the technology is used to enhance teaching and learning are data visualizations, flipped classrooms, gamification, modelling and simulations, lecture capture, mind maps, mobile devices, online assessments, polling tools, practical exercises, smartboards, tutorials, videoconferencing and webinars, virtual reality, and so on. The paper presents MATLAB-based interactive simulation model with a graphical user interface, demonstrating the processes of encoding and decoding using Hamming codes. The model allows demonstrating the process of detecting and correcting the wrong bit in the codeword received. The model can work in three modes: Step by Step, Automatic and Manual. In Step by Step mode, the micro operations required for the encoding and decoding are performed after pressing the key "Next", the user is able to monitor and learn every action in the encoding/decoding process. In Automatic mode, the entire encoding/decoding process takes place without interruption from the beginning to the end. In Manual mode, the user selects micro operations on his/her own and the system monitors the sequence of actions. If the choice is wrong, an error message appears and prevents the next action until the correct micro operation is performed. In this mode, the system enables the user to be evaluated by assigning points to his/her account for each correct answer. The tool will be used in the courses "Coding in Telecommunication Systems" and "Reliability and Diagnostics of Computer Systems" by students at the University of Ruse.

Keywords: *Teaching and learning, Hamming codes, Interactive simulation model, Encoding, MATLAB;*

1. Introduction

Nowadays the teaching staff in many universities develops their own digital education strategies. Good practices are University of Oxford [1], University of Cambridge [2], London's Global University [3]. In 2016 a Digital Education Strategy of Hungary was adopted as Annex to the Government's Proposal [4]. All these digital education strategies include the following commitments: personalizing student support, putting research and enquiry at the heart of learning, improving assessment and feedback, developing student engagement and leadership, revitalizing postgraduate taught education, creating a teaching estate to meet the needs, enriching digital learning, preparing students for the workplace and the world. The digital education strategies establish a framework for engagement and creativity in the area of education and innovation, ensuring the universities to become leaders in the integration of education and research by adopting the most exciting teaching innovations made possible by digital technologies [1,2,3,4].

Course materials, practical exercises, simulations and assessments move increasingly online and students have the flexibility to study at any time and any place. Intelligent design of these digital tools and resources can help them become experienced self-directed and autonomous learners. Academic staff may also be able to help their students' access productivity tools to help them with core learning skills such as managing their time, taking and organizing notes, planning written assignments and proofreading work. Lecturers and students describe their experience on how to get the most out of useful teaching and learning technology. Here are the domains showing where the technology is being used to enhance teaching and learning: data visualizations, flipped classrooms, gamification,

[1] University of Ruse "Angel Kanchev", Bulgaria
[2] University of Ruse "Angel Kanchev", Bulgaria
[3] University of Ruse "Angel Kanchev", Bulgaria

modelling and simulations, lecture capture, mind maps, mobile devices, online assessments, polling tools, smartboards, tutorials, videoconferencing and webinars, virtual reality, and so on [3].
Through the massive and effective use of ICT-based innovative educational technologies and didactic models, the education system is adapted to the digital generation. By introducing the research approach into the educational process, the education system is reoriented from mechanical learning to rediscovery of knowledge and developing skills. The paper presents the effective use of ICT-based educational technologies for studying Hamming codes in the courses "Coding in Telecommunication Systems" and "Reliability and Diagnostics of Computer Systems" in the University of Ruse.

2. Hamming codes

Hamming codes, invented by Richard Hamming in 1950, are a family of linear error-correcting codes, widely used in telecommunications. Hamming codes can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors. Hamming codes are perfect codes, i.e. they achieve the highest possible rate for codes with their block length and minimum distance of three. In mathematical terms, Hamming codes are a class of binary linear codes. For each integer $r \geq 2$ there is a code with block length $n=2^r-1$ and message length $m=2^r-r-1$. The rate of Hamming codes is $R=m/n=1-r/(2^r-1)$, which is the highest possible for codes with minimum distance of three and block length 2^r-1. Due to the limited redundancy that Hamming codes add to the data, they can only detect and correct errors when the error rate is low, for example in computer memory (ECC memory), where bit errors are extremely rare and Hamming codes are widely used. In this context, extended Hamming codes having one extra parity bit are often used, they achieve a Hamming distance of four, which allows the decoder to distinguish between when at most one one-bit error occurs and when any two-bit errors occur. In this sense, extended Hamming codes are single-error correcting and double-error detecting (SECDED) [5].
The general algorithm for generating a single-error correcting (SEC) code for any number of bits [5,6] is illustrated in Fig.1.

Bit position			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits			C1	C2	S1	C3	S2	S3	S4	C4	S5	S6	S7	S8	S9	S10	S11	C5	S12	S13	S14	S15	
Parity bit coverage	1	C1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
	2	C2	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	…
	4	C3	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	
	8	C4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	
	16	C5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	

Fig.1. Visualization of the general algorithm

1. Numbering the bits starting from 1: bit 1,2,3,4,5, etc.
2. Writing the bit numbers in binary: 1,10,11,100,101, etc.
3. All bit positions that are powers of two (have a single 1 bit in the binary form of their position) are parity bits: 1,2,4,8,16, etc. (1,10,100,1000,10000), denoted as C1,C2,C3,C4,C5, etc.
4. All other bit positions, with two or more 1 bits in the binary form of their position, are data bits. They are denoted as S1,S2,S3,S4,S5, etc.
5. Each data bit is included in a unique set of two or more parity bits, determined by the binary form of its bit position:
    - Parity bit 1 covers all bit positions with the least significant bit set: bit 1 (parity bit), 3,5,7,9, etc. (check 1 bit, skip 1 bit)
    - Parity bit 2 covers all bit positions with the second least significant bit set: bit 2 (parity bit), 3,6,7,10,11, etc. (check 2 bits, skip 2 bits)
    - Parity bit 4 covers all bit positions with the third least significant bit set: bits 4 – 7, 12 – 15, 20 – 23, etc. (check 4 bits, skip 4 bits)
    - Parity bit 8 covers all bit positions with the fourth least significant bit set: bits 8 – 15, 24 – 31, etc. (check 8 bits, skip 8 bits)
    - In general each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

The form of the parity is irrelevant. Even parity is mathematically simpler, but there is no difference in practice. This general rule is shown in Fig.1. Only 20 encoded bits (5 parity, 15 data) are shown but the pattern continues indefinitely.

3. Interactive simulation model implementing Hamming codes

MATLAB-based interactive simulation model with graphical user interface (GUI) is developed and presented in the paper. It is compiled using MATLAB Application Compiler 6.4 (a collection of shared libraries and code enabling the execution of compiled MATLAB applications on systems without installed MATLAB). The application allows the choice of: language – Bulgarian or English (Fig.2,*a*) and mode of operation – Encoder or Decoder (Fig.2,*b*), to visualize the processes of encoding and decoding using Hamming codes.

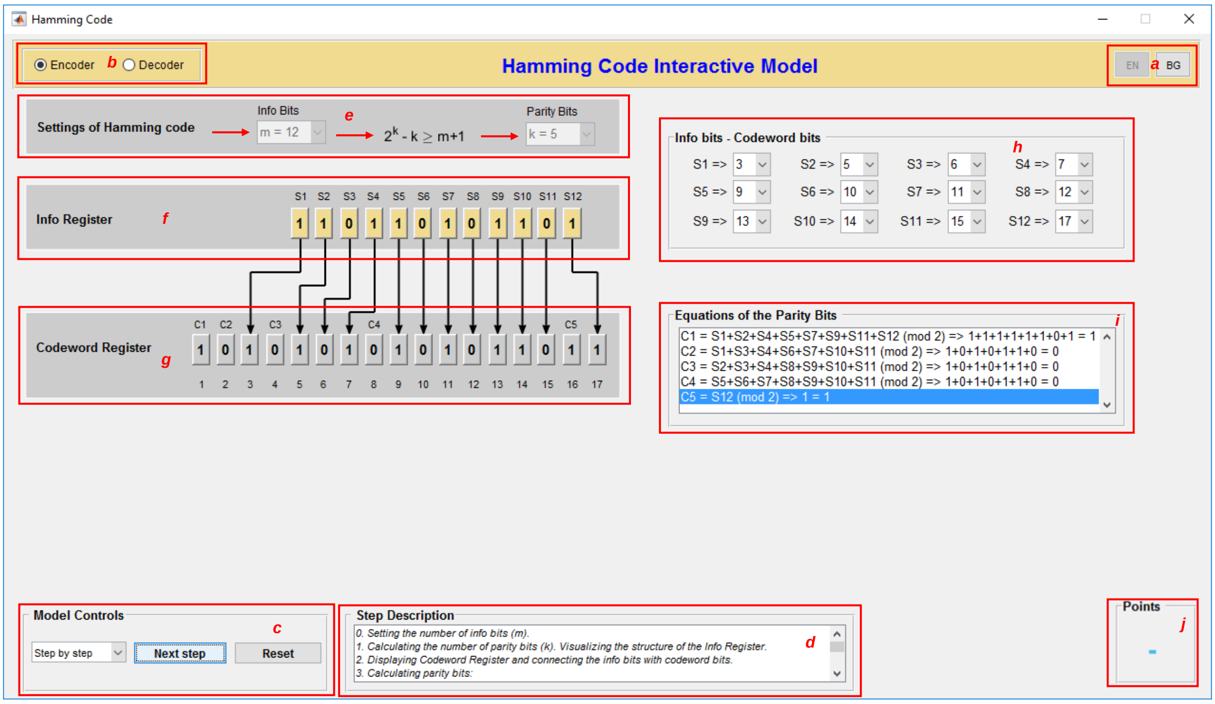

Fig.2. MATLAB-based interactive simulation model with GUI (encoding)

The user can select the mode of operation from the drop-down menu: Step by Step, Automatic and Manual (Fig.2,*c*, "Model controls"). In *Step by Step* mode, the micro-operations required for the encoding are performed after pressing the key "Next", the user is able to monitor and learn every action in the encoding process. The description of the step is shown in "Step Description" (Fig. 2,*d*). In *Automatic* mode, the encoding process takes place without interruption from beginning to end. This mode is applicable to checking the student's work, but also allows tracking the encoding process. In *Manual* mode, the user selects the micro-operations and the system monitors the sequence of actions and displays an error message if wrong choice and prevents the next action until the correct micro-operation is performed. In this mode, the system allows the user to be evaluated by points (Fig.2,*j*) for each correct answer (not implemented now).

The user selects the number of info bits (m=4…12) using drop-down menu in "Settings of Hamming code" (Fig.2,*e*). Then the number of parity bits k is calculated and the structure of Info Register (Fig.2,*f*) is visualized. After inserting the message bits in Info Register by clicking on zeros for changing to ones, the student can see in detail the steps in encoding (Fig.2,*d*). At the end of simulation, the corresponding encoded message is visualized in Codeword Register (Fig.2,*g*). The correspondence between info and codeword bits is shown in Fig.2,*h* and equations for calculating the parity bits are visualized in Fig.2,*i*. Their visualization assists the students in perception the process of encoding messages using Hamming codes, as at any time the student can trace the encoding execution. The way of calculating the parity bit C3 and its result are shown in Fig.3.

The equations used in encoding are as follows:

$C1=S1\oplus S2\oplus S4\oplus S5\oplus S7\oplus S9\oplus S11\oplus S12$, $C2=S1\oplus S3\oplus S4\oplus S6\oplus S7\oplus S10\oplus S11$,

$C3=S2\oplus S3\oplus S4\oplus S8\oplus S9\oplus S10\oplus S11$, $C4=S5\oplus S6\oplus S7\oplus S8\oplus S9\oplus S10\oplus S11$, $C5=S12$.

After encoding the message, the interactive model offers decoding the codeword when entering a single-bit error in the codeword. The decoding model is very similar (Fig.4).

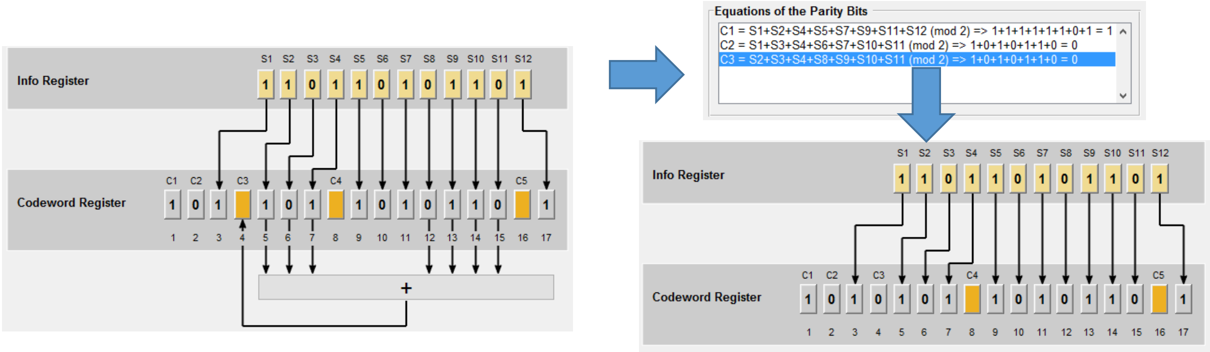

Fig.3. Calculating the parity bit C3

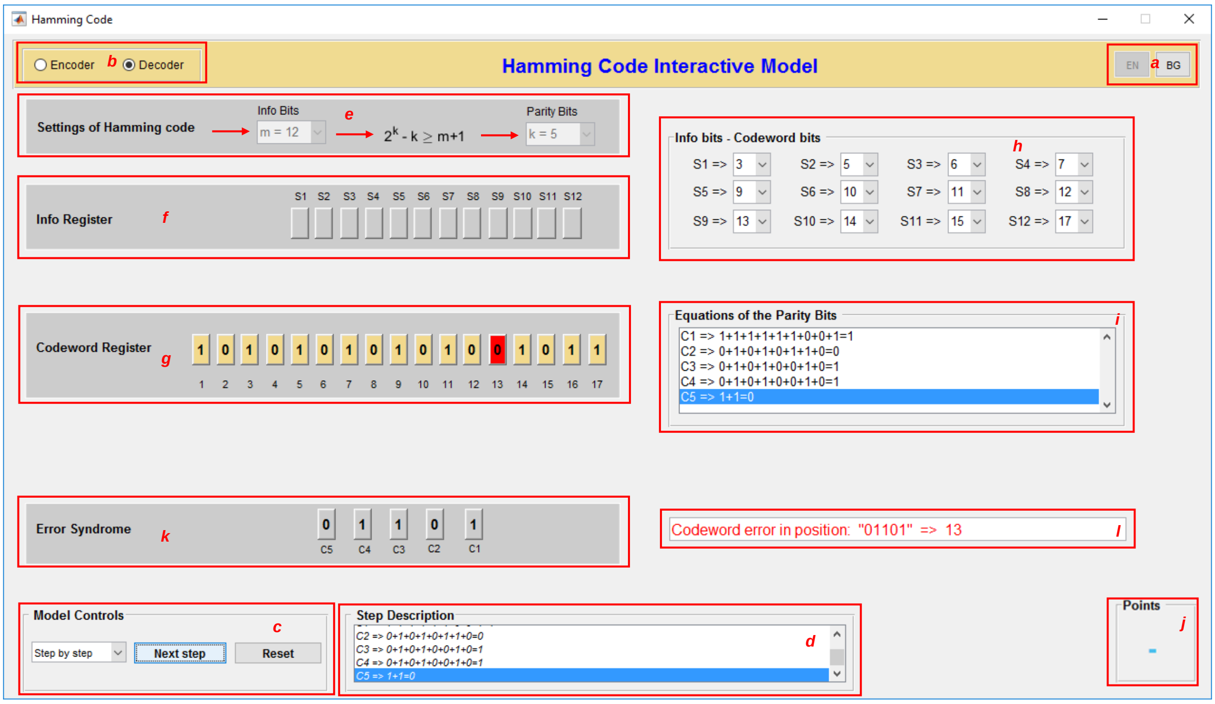

Fig.4. MATLAB-based interactive simulation model with GUI (decoding)

The bits of the codeword are filled in Codeword Register (Fig.4,*g*). For example, for illustrating the process of decoding and correcting a single-bit error, let the 13^{th} bit of the codeword be wrong (by clicking on the bit, it changes its value; it is on red background). Then the bits of the syndrome C1…C5 (Fig.4,*k*) are calculated step by step (Fig.5).

After calculating the syndrome, it is found that the bit in position 01101 (binary) or 13 (decimal) is wrong (Fig.4,*l*). The model inverts the erroneous bit and transfers the codeword bits corresponding to the information word in Info Register (after correcting the wrong bit). These processes are shown in Fig.6.

4. Conclusions

The paper presents MATLAB-based interactive simulation model with GUI, demonstrating the processes of encoding *m*-bit messages (*m*=4…12) and decoding the codeword using a single-error correcting (SEC) Hamming codes. The model can work in three modes: Step by Step, Automatic and Manual. The Step by Step mode is presented in details in the paper. The application is used in the courses “Coding in Telecommunication Systems” and “Reliability and Diagnostics of Computer Systems" by undergraduate students of the specialties “Internet and Mobile Communications” and “Computer Systems and technologies” at the University of Ruse.

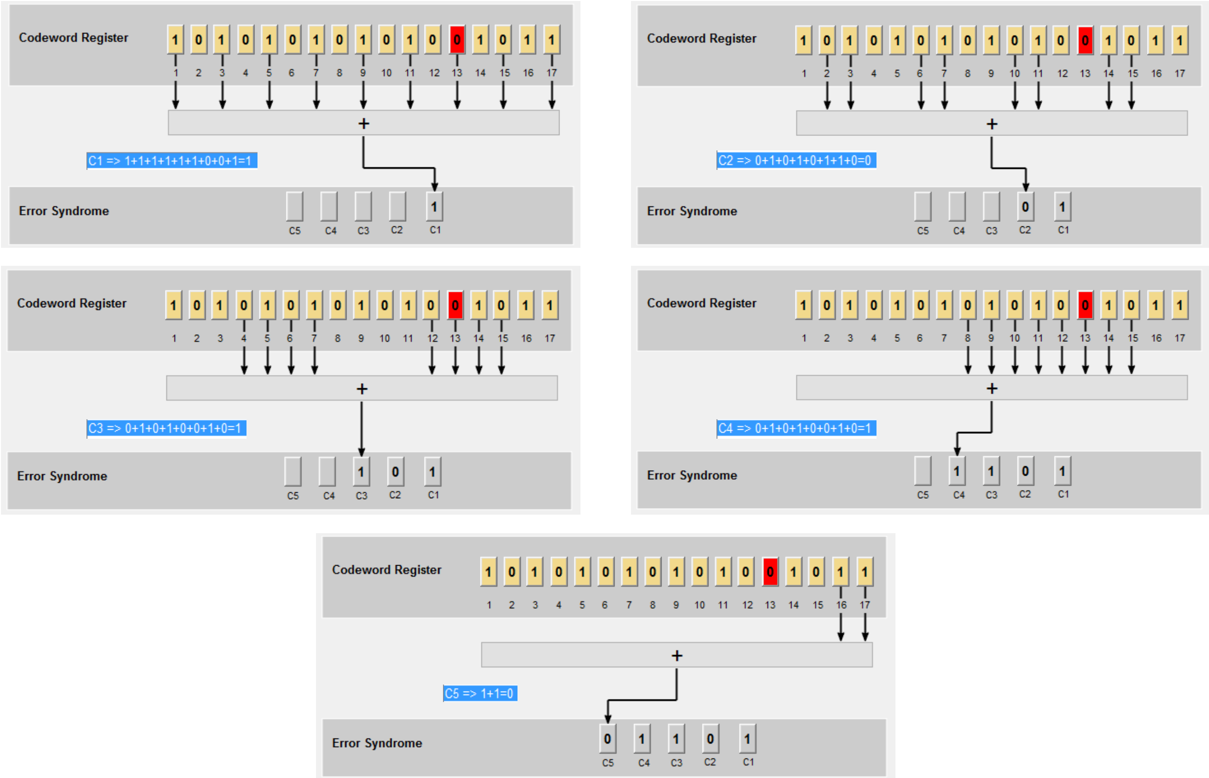

Fig.5. Calculating the syndrome

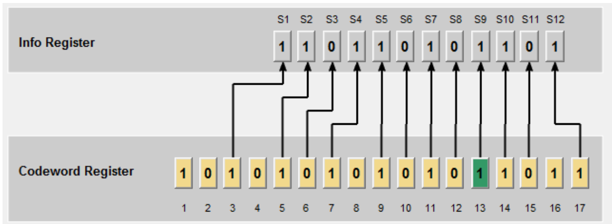

Fig.6. Correcting single-bit error and decoding

The study was supported by contract of University of Ruse “Angel Kanchev”, № BG05M2OP001-2.009-0011-C01, “Support for the development of human resources for research and innovation at the University of Ruse “Angel Kanchev””. The project is funded with support from the Operational Program “Science and Education for Smart Growth 2014-2020” financed by the European Social Fund of the European Union.

References

[1] Digital Education Strategy 2016-2020, University of Oxford, April 2016, https://www.admin.ox.ac.uk/media/global/wwwadminoxacuk/localsites/educationcommittee/documents/notesofguidance/Digital_Education_Strategy_2016_-_2020-_final.pdf (January 2018).
[2] Digital Strategy for Education, University of Cambridge, Cambridge Centre for Teaching and Learning, 2016, https://www.cctl.cam.ac.uk/news/digital-strategy-education (January 2018).
[3] UCL Education Strategy 2016-21, London’s Global University, 2016, https://www.ucl.ac.uk/teaching-learning/education-strategy (January 2018).
[4] Digital Education Strategy of Hungary. Annex to the Government’s Proposal, 30 June 2016, http://www.kormany.hu/download/0/4b/21000/The%20Digital%20Education%20Strategy%20of%20Hungary.pdf (January 2018).
[5] Hamming code. https://en.wikipedia.org/wiki/Hamming_code (January 2018).
[6] Calculating the Hamming Code. http://users.cis.fiu.edu/~downeyt/cop3402/hamming.html (January 2018).