Enhancing Distance Learning in Digital Logic Design Through Automated Self-Testing and Assignment Verification

Mohammed Fadhl Abdullah(1*)
Ammar Ali Mehdi Kurd(1)

Received: 01/01/2026
Revised: 26/01/2026
Accepted: 27/01/2026

---

[1] Deparrtment of Engineering, Faculty of Engineering and Computing, University of Science and Technology, Aden, Yemen.
*Corresponding Author's Email: m.albadwi@ust.edu

Enhancing Distance Learning in Digital Logic Design Through Automated Self-Testing and Assignment Verification

Mohammed Fadhl Abdullah
Deparrtment of Engineering, Faculty of Engineering and Computing, University of Science and Technology, Aden, Yemen.
m.albadwi@ust.edu

Ammar Ali Mehdi Kurd
Deparrtment of Engineering, Faculty of Engineering and Computing, University of Science and Technology, Aden, Yemen.
a.kurd@ust.edu

***Abstract*—** Distance learning systems, particularly Learning Management Systems (LMSs), have evolved from basic content-sharing tools into comprehensive platforms that support assignments, assessments, and virtual lectures. Their importance became especially evident during global disruptions, ensuring educational continuity. However, one major challenge in such systems is minimizing turnaround time, particularly in grading and verifying assignments in courses with large enrollments. This paper introduces an automated assignment verification technique designed for digital logic design courses, where multiple functionally equivalent circuit implementations make manual evaluation time-consuming and inconsistent. The proposed approach employs a signature analysis method known as the Concurrent Intermediate Checking (CIC) register, which verifies circuit behavior at predefined checkpoints through diagnostic signatures. By embedding this self-testing mechanism into students' workstations, the system delivers immediate feedback, reduces erroneous submissions, and enhances self-learning. Experimental validation using high-level synthesis benchmarks demonstrates the reliability, efficiency, and scalability of the proposed approach, improving both instructional productivity and student outcomes in distance learning environments.

***Keywords*—** Medical Images, Augmented Reality, Visualization , 3D Reconstruction, Segmentation.

I. INTRODUCTION

Computer engineering curricula are designed to ensure that students acquire both theoretical knowledge and practical skills relevant to the discipline. While programming has become a standard component of computer science education, the integration of circuit design techniques in computer engineering courses has progressed more slowly. This lag is largely due to the cost and complexity of hardware design tools compared with software compilers [1], [2]. Courses such as Logic Design introduce students to sequential circuits, Boolean methods, and memory structures. Assignments in these courses should ideally include practical design tasks involving various digital circuits. Traditional paper-and-pencil design approaches, however, are often error-prone and discouraging for students, reducing motivation and engagement. In contrast, the use of circuit design systems such as Verilog and PSpice greatly enhances learning outcomes. Through simulation, students can identify and correct design errors, increasing both their confidence and the quality of their work. Additionally, students gain valuable experience using professional tools employed in the field. For instructors, the use of such systems improves the readability of student submissions and enables automated correctness checking, especially in courses with large enrollments. This allows teachers to focus on evaluating the quality and efficiency of designs rather than verifying basic correctness.

Testing VLSI circuits remains one of the most time-consuming challenges in digital system design. Engineering students, besides learning how to design, also need to know how to test their designs effectively. Yet, only the most dedicated students typically pursue this skill independently. As modern engineers increasingly engage in System-on-Chip (SoC) technologies, they must have a strong understanding of Design-for-Test (DFT) and Built-In Self-Test (BIST) methodologies as applied to the Circuit Under Test (CUT) [3], [4].

A further challenge in distance learning environments is the significant time required to check and correct students' digital design assignments, particularly in large classes. Several approaches have been proposed to automate or simplify this process. For instance, the WebAssign system [6], [7], [8] provides modules for automatic grading of numerical, true/false, and multiple-choice questions submitted as HTML documents. While effective for theoretical exercises, it is less suitable for complex circuit design tasks.

Many studies have attempted to automate the testing of digital circuit design assignments using conventional compaction methods. Their approach depends on applying a massive number of test signals to the CUT and compacting the outputs into a final signature using a linear feedback shift register (LFSR) that is compared with a golden reference [2], [5], [9], [10]. However, the limitations with this method are a high time delay in generating the signature and the instructor dependency (since teachers must perform fault simulation to generate test inputs) [11].

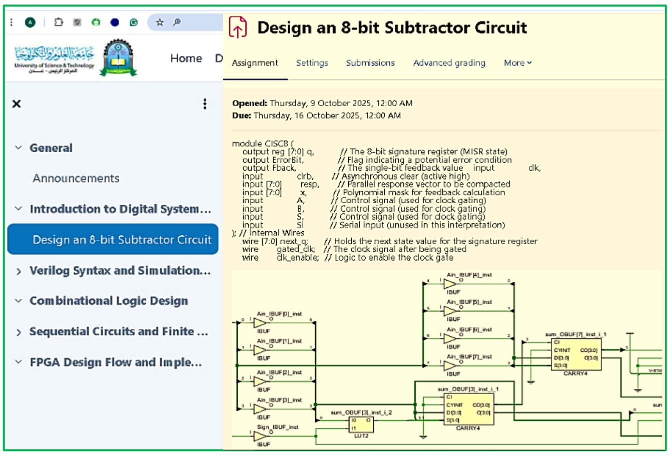
Agarwal [12] analyzed the area and power consumption of Transmission Gate-based TSS LFSR and found it more efficient than other LFSR architectures. The design required fewer gates, reduced power, and minimized area usage. While the study used the traditional stuck-at fault model, delay testing based on delay fault models is becoming increasingly important for verifying timing specifications. Despite several low-power methods proposed for stuck-at testing, few address low-power delay testing, which remains crucial for modern systems. Research in [13] examined Moodle's functionalities in a private higher education institution during the COVID-19 pandemic, highlighting the UTAUT-3 constructs as key to understanding Moodle adoption. The study found performance expectancy to be a major factor influencing lecturers' perception of Moodle's usefulness. Similarly, [14] provided evidence that motivation and perceived usefulness drive the use of intelligent tutoring systems, offering insights into student engagement in digital learning. Kasakowskij [15] identified major limitations in feedback practices in distance education, such as difficulty formulating feedback and managing multiple responses. A bidirectional feedback model was developed and implemented in Moodle, showing promising results, though further validation across disciplines is needed. In [16], the authors discussed the technical and psychological challenges faced by higher education institutions during the COVID-19 pandemic, emphasizing Moodle's effectiveness in supporting temporary distance education despite its limitations.

These challenges highlight the need for more efficient, automated, and reliable methods for evaluating digital circuit designs in distance learning environments. The present study addresses these challenges in evaluating digital circuit designs in distance learning environments by proposing a new student assignment method that reduces turnaround time and enhances both student learning and instructional efficiency. The approach integrates a Verilog simulator into a Moodle-based course management system developed for the faculty of Engineering and Computing at the University of Science and Technology. It uses online assignments and feedback that ensure improved learning outcomes. The proposed system enhances students' skills in digital hardware concepts by considering the students' own needs and also achieves the availability of course material independent of time and location.

The main contributions of this study are (i) proposing a checkpoint-based automated verification method for digital logic assignments in distance learning, (ii) introducing the CIC register for concurrent intermediate signature comparison with reduced aliasing, (iii) implementing a self-testing, LMS-integrated framework with personalized assignments and immediate feedback, and (iv) validating effectiveness through benchmark circuits, showing high fault coverage and scalability.

II. TEACHER–STUDENT INTERACTION MODEL IN THE PROPOSED SYSTEM

In this teaching environment, teachers are able to personalize an assignment for each student through a different set of testing parameters and adjust the working period (accessing the assignment and submitting the solutions). Here, the student can download his design assignment, start to develop a solution for it, and run a pretest process before submitting it back to the server. He also has the chance to repeatedly update his solution and resubmit it before the end of the working period is reached. There are no predefined measurements; instead, students must carry out the testing process themselves to verify the correctness of their designs. Since the self-test is executed on the student's own computing device, the system provides immediate feedback on whether the circuit functions correctly. Consequently, this approach helps reduce the number of faulty solutions submitted by students. Figure 1 illustrates the preparation of a Verilog testing template that is sent with an assignment for students to solve and test.



Design an 8-bit Subtractor Circuit

Opened: Thursday, 9 October 2025, 12:00 AM
Due: Thursday, 16 October 2025, 12:00 AM

```
module CISC8 (
    output reg [7:0] q,         // The 8-bit signature register (MISR state)
    output ErrorBit,            // Flag indicating a potential error condition
    output Fback,               // The single-bit feedback value    input            clk,
    input           clrb,      // Asynchronous clear (active high)
    input [7:0]      resp,      // Parallel response vector to be compacted
    input [7:0]      x,         // Polynomial mask for feedback calculation
    input           A,         // Control signal (used for clock gating)
    input           B,         // Control signal (used for clock gating)
    input           S,         // Control signal (used for clock gating)
    input           Si         // Serial input (unused in this interpretation)
); // Internal Wires
    wire [7:0] next_q;      // Holds the next state value for the signature register
    wire       gated_clk;   // The clock signal after being gated
    wire       clk_enable;  // Logic to enable the clock gate
```

Figure 1. Preparing an assignment in CMS.

III. CHECKPOINT-BASED SIGNATUREM VERIFICATION SCHEME

Let $r_1^*, \dots, r_n^*$ be the fault-free responses of the digital circuit after applying n test patterns which are compacted by the MISR register into a single signature G_n (refer here as the golden signature). Let $r_1 \dots r_n$ be the corresponding outputs of the CUT and their compacted output be S_n. The CUT is declared faulty, if there are any differences between the outputs G_n and S_n. However, it is also valid to declare the design faulty if any output signal r_j differs from the corresponding correct signal r_j^* (referred here as a partial golden signature).

This testing scheme, where multiple intermediate signals are compared with the respective partial golden signatures (instead of comparing only the final signature S_n) improves the confidence of the testing process and reduces the probability of aliasing, where the signature of the faulty CUT matches the golden signature. However, this scheme has the problem of the storage and comparison of many signatures. In this study, a solution is presented to this problem based on the concept of checkpoints. The essential idea is that there exist several time instants, denoted as i, at which the golden partial signature can be indirectly derived from the responses of the CUT.

Let Boolean function F_i refer to a function that maps an input to an output, where bit at jth position is flipped iff the binary representation of i contains a 1 in the jth position $1 \le j < m$. The proposed model is based on functions $F_i(r_i^*) = \widetilde{G_{i-1}}$, at instance i. This instance i

is referred to as a correct, golden CP GCP_i. There are 2^m possible functions, we denoted them as F_0, $F_1 \dots F_{2^m-1}$. A check point CP_i is called a faulty check point if $r_i \neq \widetilde{S_{i-1}}$ for any function F_x and the signal at this instance is High and called an error bit. There must be at least one faulty check point to declare that the CUT is faulty. Our proposed scheme based on the availability of these nearly free check points. Figure 2 demonstrates the concepts of the partial signatures and the error bit generation.

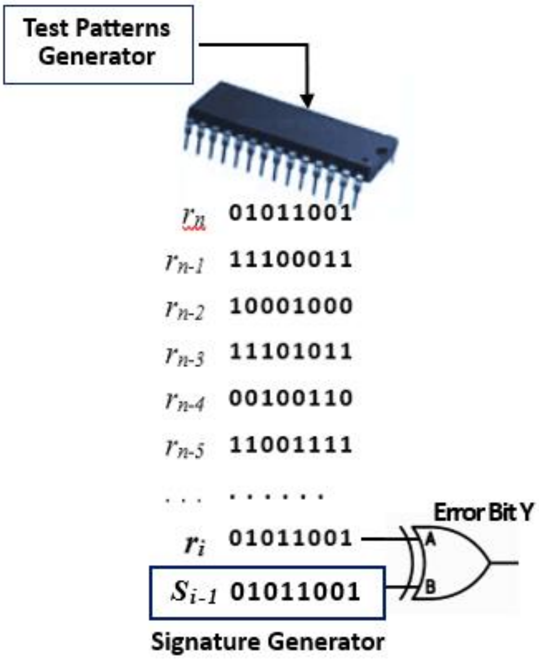


Figure 2. Generating the Error bit Y.

We modified the MISR register to implement the error bit, and referred to it as the “Concurrent Intermediate signature Comparison Register” (CIC) and is illustrated in Figure 3. The corresponding Verilog CIC register design is shown in Figure 4.

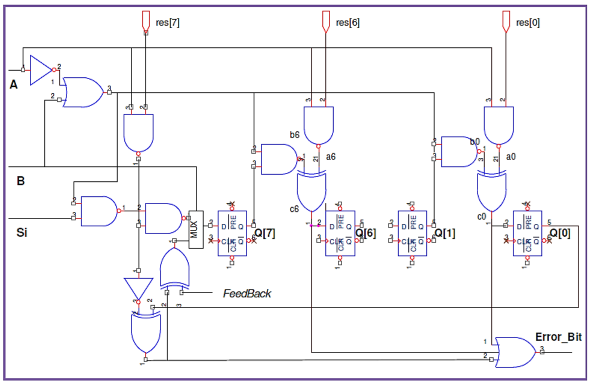


Figure 3. CIC register design.

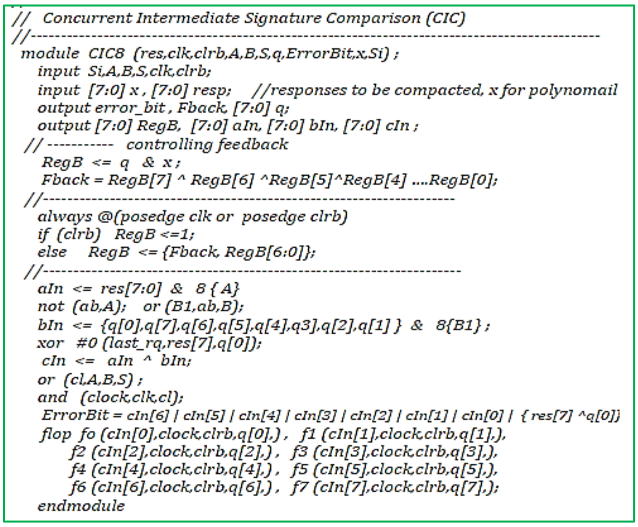
At the instant i the error bit Y is calculated as

$$Y_i = \begin{cases} 0 & if \quad r_i = S_{i-1} \\ 1 & otherwise \end{cases}.$$

And with an indicator bit X at the time instant i,

$$X_i = \begin{cases} 0 & if \;\; cp \; instance \; i \\ 1 & otherwise \end{cases}$$

Faulty CUT is declared when at time instant i, the logical AND of the indicator bit X and the error bit Y is 1 (a faulty check point $S_{i-1} \neq r_i$).



```
// Concurrent Intermediate Signature Comparison (CIC)
//------------------------------------------------------------------------------------------------
 module CIC8 (res,clk,clrb,A,B,S,q,ErrorBit,x,Si) ;
    input Si,A,B,S,clk,clrb;
    input [7:0] x , [7:0] resp;    //responses to be compacted, x for polynomail
    output error_bit , Fback, [7:0] q;
    output [7:0] RegB, [7:0] aIn, [7:0] bIn, [7:0] cIn ;
  // ----------  controlling feedback
     RegB <= q  & x ;
     Fback = RegB[7] ^ RegB[6] ^RegB[5]^RegB[4] ....RegB[0];
  //---------------------------------------------------------------------
    always @(posedge clk or  posedge clrb)
    if (clrb)  RegB <=1;
    else   RegB <= {Fback, RegB[6:0]};
  //---------------------------------------------------------------------
    aIn <= res[7:0] &  8 { A}
    not (ab,A);  or (B1,ab,B);
    bIn <= {q[0],q[7],q[6],q[5],q[4],q3],q[2],q[1] } &  8{B1} ;
    xor  #0 (last_rq,res[7],q[0]);
     cIn <=  aIn ^ bIn;
    or (cl,A,B,S) ;
    and  (clock,clk,cl);
     ErrorBit = cIn[6] | cIn[5] | cIn[4] | cIn[3] | cIn[2] | cIn[1] | cIn[0] | { res[7] ^q[0]}
     flop fo (cIn[0],clock,clrb,q[0],) ,  f1 (cIn[1],clock,clrb,q[1],),
          f2 (cIn[2],clock,clrb,q[2],) ,  f3 (cIn[3],clock,clrb,q[3],),
          f4 (cIn[4],clock,clrb,q[4],) ,  f5 (cIn[5],clock,clrb,q[5],),
          f6 (cIn[6],clock,clrb,q[6],) ,  f7 (cIn[7],clock,clrb,q[7],);
    endmodule
```

Figure 4. A Verilog module of an 8_bit CIC register.

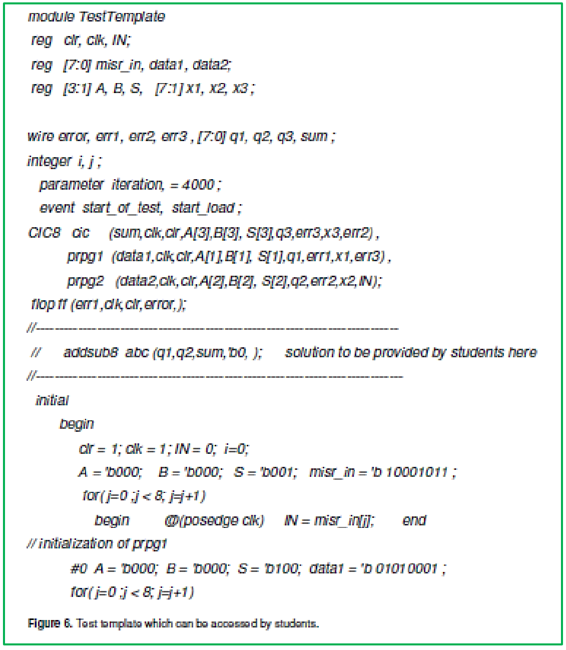
IV. ASSIGNMENT SOLVING USING THE CIC-BASED TEST TEMPLATE

The test template is illustrated in Figure 5. It is designed for the self-testing of a simple adder/subtractor circuit provided to students. The template includes two CIC register modules that generate input test vectors, and students can modify the feedback configuration settings as needed. In this mode, they must define the initial state, configure the feedback structure, and specify the length of the test sequence. During the test generation phase, students can also select a target fault and execute tests for different experimental objectives.

4.1. *Personalized Assignment Tasks*

Every student, after solving the assigned task of a circuit design, has to update the downloaded test template by injecting his own code in between the pattern generator and the CIC register code and running the test process. Reaching the non-faulty checkpoint and the circuit working correctly, the student can now submit his solution (design files) to the faculty server.

The teacher in our proposed scheme can use different initial seed values for the CIC registers to personalize the exercise for each student. The main advantage of this self-testing method is the fast feedback to the student. However, for complex designs, such as sequential complex circuits, this scheme also greatly reduces the time of checking the students' assignments.



```
module TestTemplate
 reg   clr, clk, IN;
 reg   [7:0] misr_in, data1, data2;
 reg   [3:1] A, B, S,   [7:1] x1, x2, x3 ;

wire error, err1, err2, err3 , [7:0] q1, q2, q3, sum ;
integer  i, j ;
  parameter  iteration, = 4000 ;
  event  start_of_test,  start_load ;
CIC8   cic     (sum,clk,clr,A[3],B[3], S[3],q3,err3,x3,err2) ,
         prpg1  (data1,clk,clr,A[1],B[1], S[1],q1,err1,x1,err3) ,
         prpg2   (data2,clk,clr,A[2],B[2], S[2],q2,err2,x2,IN);
 flop ff (err1,clk,clr,error,);
//-----------------------------------------------------------------------------
 //     addsub8  abc (q1,q2,sum,'b0, );     solution to be provided by students here
//------------------------------------------------------------------------------
  initial
       begin
           clr = 1; clk = 1; IN = 0;  i=0;
           A = 'b000;    B = 'b000;   S = 'b001;   misr_in = 'b 10001011 ;
            for( j=0 ;j < 8; j=j+1 )
               begin        @(posedge clk)     IN = misr_in[j];      end
// initialization of prpg1
         #0  A = 'b000;  B = 'b000;  S = 'b100;  data1 = 'b 01010001 ;
         for( j=0 ;j < 8; j=j+1 )
```

Figure 6. Test template which can be accessed by students.

Figure 5. Test template which can be accessed by students.

4.2. *Timing-Aware Evaluation of Student Designs*

The simulator includes checking of time and gate delay. For example, if the assignment asks the students to design a carry-select adder from a given 8-bit adder and multiplexer blocks, the correct solution to this assignment does only differ in the clocking time, but not the functionality of the circuit. And by running the testing at a clock speed, the teacher can easily detect the slower (i.e., the wrong) design solution, without the need for additional timing checks.

4.3. *Alarm Check Points (ACPS)*

Our simulation process has shown that the compaction of a faulty design may generate many similar instants to that of the original fault-free check points but occur at different time instants. These instances are referred to here as alarm checkpoints. While the basic golden CPs for a particular digital circuit are determined only by simulation, no simulations are required to determine the Alarm Check Points.

This study has shown that $ACPs$ have a property that is superior over basic CPs, where only a single ACP is sufficient to declare designed circuit developed by the student as a faulty one. Table 1 shows the existence of many of such signals, basic, faulty, and alarm check points, in the testing of a number of benchmarks circuits, which had been infected by a stuck-at-one fault in a wire. As shown in the table, there are adequate number of the fault check points and the alarm check points.

Table 1. Number of CPs for faulty circuits.

Benchmark circuits (CUTs)	**Fault_free circuit**			**Faulty_circuit (stuck at 1)**		
	BCPs	**FCPs**	**ACPs**	**BCPs**	**FCPs**	**ACPs**
Adder/Subtractor	40	0	0	40	5	**8**
Differential Equation	18	0	0	18	3	**6**
Chinese R. Theorem	16	0	0	16	4	**5**
AR filter	4	0	0	4	8	**9**
Exponential Func.	19	0	0	19	12	**3**
S713	35	0	0	35	8	**3**
S1196	**36**	**0**	**0**	**36**	**11**	**6**

The fault coverage percentage for CUT computed as

$$FC = \frac{No.\ detected\ faults}{Total\ number\ of\ CKT\ faults} \times 100\%$$

Table 2. Fault coverage of stuck_at faults in the 32 primary inputs of 8-bit subtractor (1, 2, 6 = x + $x^2 + x^6$).

Ch/s Polys	**Fault coverage (%)**			
	BCPs	**FCB**	**FCA**	**FCT**
$\mathbf{1 + x + x^2 + x^5 + x^8}$	32	81.45	96	**100**
$\mathbf{1 + x + x^3 + x^5 + x^7}$	21	83.75	84.7	**91.75**
$\mathbf{1 + x^3 + x^7 + x^8}$	18	68.0	87.0	**91.75**
$\mathbf{1 + x^2 + x^4 + x^8}$	**26**	**84.20**	**92**	**90**

Table 3. CPs of particular benchmarks for 4000 test patterns

Benchmark circuits (CUTs)	**No. basic CPs**		
	F_1	F_8	$\boldsymbol{F_{16}}$
Adder/Subtractor	40	8	**16**
differential equation	18	12	**20**
Chinese Reminder Theorem	16	22	**18**
AR filter	4	11	**6**
exponential functions	19	11	**35**
ISCAS'89 Benchmark circuits			
S420	42	22	**32**
S713	35	31	**44**
S1196	**36**	**29**	**30**

In the context of automated assignment verification, ACPs are particularly important for grading fairness, as they provide a deterministic indicator of faulty designs independent of implementation variations. By requiring only a single ACP to reject an incorrect solution, the proposed system ensures consistent and unbiased evaluation of all student submissions.

V. EVALUATION

Each circuit under test can have up to 2^{2b} sets of the BCPs (2^b different sets of BCPs in an b-bit CIC and 2^b possible functions). Table 2 illustrates the stuck-at-zero/one fault coverage of an 8-bit subtractor (32 primary inputs) for the functions FC_B, FC_A, and FC_T and for the fault coverage due to the Basic check points, Alarm check points and due to both, respectively. The proposed technique enhances the process of checking logic circuit assignments by varying the length of the test patterns from 1 to 5000.

The simulated CPs for certain high-level synthesis benchmarks (including some of ISCAS'89 circuits) are shown in Table 3 for functions F_1, F_8, and F_{16} tested with 4000 test patterns. There exists at least $\frac{n}{2^b}$ CPs for n test patterns, irrespective of the characteristic polynomial and the function.

VI. CONCLUSION

This paper presented and implemented a novel approach to address key challenges in distance education for digital circuit design courses. The proposed method improves the teaching and evaluation process by introducing the concept of partial checkpoints as a basis for verifying student designs. The approach involves observing CPs generated during testing and tracing alarm points in defective circuits to localize faults efficiently. The main contribution of the proposed scheme is an improved reliability and the fast fault circuit detection at a responsible number of computational resources. The test frame can also be available to the students. They can utilize it for either self-training or observing the circuit behavior. Furthermore, the system proves particularly effective for large sequential circuits. Overall, the proposed approach shortens the correction cycle, supports scalable teaching, and enhances the overall quality and efficiency of distance learning in digital design education.

Author Contributions

Conceptualization, M.F.A.; Methodology, M.F.A. and A.A.M. K.; Validation, A.A.M. K; Writing Original Draft Preparation, M.F.A..; Writing Review & Editing, M.F.A and A.A.M. K; Supervision, M.F.A.

REFERENCES

1. T. Michael, "Integrated Online Courseware for Computer Science Courses," in *Proc. Conf. Innovation Technol. Comput. Sci. Educ. (ITiCSE),* Dundee, Scotland, Jun. 25–27, 2007.
2. H. Han-Pang and L. Chiou-Hwa, "Java-Based Distance Learning Environment for Electronic Instruments," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 1–8, Feb. 2003.
3. M. Kumar, "An Efficient Fault Detection of FPGA and Memory Using Built-in Self-Test [BIST]," *Am. J. Electr. Comput. Eng.*, vol. 3, no. 1, pp. 38–44, 2019. doi: 10.11648/j.ajece.20190301.15.
4. Z. Peng, "Research on Low Power BIST Based on LFSR Reseeding," *Acad. J. Sci. Technol.*, vol. 3, no. 2, pp. 203–208, 2022. doi: 10.54097/ajst.v3i2.2174.
5. B. Koenemann, J. Mucha, and G. Zwiehoff, "Built-in Logic Block Observation Techniques (BILBO)," in Proc. Int. Test Conf., 1979, pp. 37–41.
6. H. Norman, "Automatic Checking of Students' Designs Using Built-In Self-Test Method," in Proc. 4th Eur. Workshop Microelectron. Educ. (EWME), Baiona, Spain, May 2002.
7. M. Vladimir, P. Plamen, and L. Mihail, "Tutoring Tool for Logical Schemes Design," in Proc. Int. Conf. Comput. Syst. Technol. (Comp. Sys. Tech.), 2004.
8. J. Brunsmann, A. Homrighausen, H. Six, and J. Voss, "Assignments in a Virtual University: The WebAssign-System," in Proc. 19th World Conf. Open Learn. Distance Educ., Vienna, Austria, 1999.
9. R. Ubar and H. D. Wuttke, "Action Based Learning System for Teaching Digital Electronics and Test," in Proc. 3rd Eur. Workshop Microelectron. Educ. (EWME), Aix-en-Provence, France, 2000, pp. 107–110.
10. N. Hendrich, "A Java-Based Framework for Simulation and Teaching: Hades," in Microelectron. Educ., Proc. EWME 2000, 2000, pp. 285–288.
11. M. L. Bushnell, "Increasing Test Coverage in a VLSI Design Course," in Proc. Int. Test Conf., Atlantic City, NJ, USA, 1999.
12. S. Agarwa and S. Rathore, "Area and Power Calculation of TSS LFSR and Its Effect on Benchmark Circuits," J. Circuits Syst., vol. 7, no. 3, pp. 6–10, 2019.
13. R. Jabar and D. Govender, "Moodle Use in South African Distance Education: Based Exploration on Student and Lecturer Perspective," Int. J. Bus. Ecosyst. Strategy, vol. 7, no. 4, pp. 309–322, 2025.
14. M. Janson and S. Janke, "The Influence of E-Learning on Exam Performance and the Role of Achievement Goals in Shaping Learning Patterns," Int. J. Educ. Technol. High. Educ., vol. 21, no. 56, pp. 1–22, 2024. doi: 10.1186/s41239-024-00488-9.
15. R. Kasakowskij and J. Haake, "Supporting a Bidirectional Feedback Process for Self-Assessment Tasks in a Digital Learning Environment," *Int. J. Educ. Technol. High. Educ.*, vol. 22, no. 15, pp. 1–25, 2025.
16. O. V. Kisel et al., "The Moodle System in the FL Distance Learning During the COVID-19 Pandemic," in *Proc. 17th Int. Sci. Conf. eLearning Softw. Educ.*, Bucharest, Romania, Apr. 21–23, 2021, pp. 270–277.