Construction and Practical Path of an AI-Teaching-Platform-Empowered Smart Curriculum for Digital Electronics Technology

Yadongyang Zhu, Lanya Yu, and Bo Lan*

School of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing, China

Abstract: Aiming at problems in the traditional teaching of Digital Electronics Technology-fragmented knowledge structures, superficial classroom interaction, verification-oriented laboratory training, and delayed evaluation-this paper proposes a full-process smart curriculum construction path based on the functional system of an AI teaching platform. Taking the knowledge base, course model, AI applications, and decision center as the technical foundation and the curriculum knowledge graph as the organizing thread, the path embeds intelligent lesson preparation, one-click question generation, intelligent learning companions, AI teaching companions, intelligent grading, and teaching data analysis into preparation, classroom teaching, after-class learning, and evaluation. Taking arbitrary-modulus counter design as an example, an instructional process integrating these functions is designed. The study argues that the key lies not in stacking tools, but in reconstructing the teaching process around the course knowledge base and learning data, so that students transfer from knowledge memorization to logic design and from simulation verification to engineering expression through continuous feedback.

Keywords: Digital Electronics Technology, AI Teaching Platform, Smart Curriculum, Knowledge Graph, Intelligent Learning Companion, Process-Based Evaluation.

1. Introduction

Digital Electronics Technology is an important foundational course for majors such as electronic information, electrical engineering, automation, and computer science, covering number systems, Boolean algebra, combinational logic, flip-flops, counters, pulse generation and shaping, memories, and programmable logic devices. It emphasizes logical reasoning and circuit analysis and requires students to design digital systems from practical requirements, providing support for subsequent courses such as microcontroller principles, embedded systems, and FPGA development.

The difficulty of the course lies not in isolated knowledge points but in their hierarchical progression and engineering connections. Learning passively by chapter, students may remember local methods such as Karnaugh-map simplification and counter state transitions, yet fail to complete the full process of requirement analysis, logic abstraction, circuit design, simulation verification, and result evaluation in an integrated task. Meanwhile, pre-class preparation is invisible, in-class interaction concentrates on a few students, assignment feedback cycles are long, and evaluation is dominated by final examinations. AI teaching platforms, which embed large-model capabilities into preparation, teaching, assessment, and data analysis through dedicated knowledge bases, course models, intelligent companions, and decision centers, are highly adaptable to this knowledge-structured, data-rich course [1].

2. Major Problems in Traditional Course Teaching

2.1. Chapter-Based Knowledge Without a Navigable Course Model

The knowledge has clear dependencies: logic simplification underlies combinational design, flip-flop equations underlie sequential analysis, and counter design integrates flip-flops, state transitions, clearing and loading conditions, and chip function tables. Proceeding by chapter, students see a linear chapter list rather than a traceable knowledge network, and when learning asynchronous-clear counters they often cannot tell whether their weakness lies in the function table, the state sequence, or the clear-logic expression. This lack of navigation is an important cause of inefficient learning.

2.2. Experience-Dependent Interaction and Untimely Feedback

Classroom questioning and demonstration depend on teachers' immediate judgment, making it hard to grasp the whole class's preparation, error rates, and knowledge-point mastery, especially in large classes where interaction concentrates on a few active students. For difficult concepts such as transient states, race hazards, and flip-flop triggering conditions, the absence of timely explanation lets cognitive barriers accumulate in later learning.

2.3. Verification-Oriented Practice Weakening Engineering Expression

Common experiments-gate testing, decoder applications, flip-flop verification, and counter observation-consolidate basic knowledge, but students mostly reproduce steps with few chances to design schemes from real requirements. Even after completing wiring and simulation, many cannot explain the design rationale, state-transition process, or optimization path, so practice that stops at verifying correctness cannot support engineering design and complex problem-solving ability.

2.4. Scattered Evaluation Data Without a Continuous-Improvement Loop

Evaluation usually combines usual performance and a final examination, but data collection is discontinuous and feedback is coarse. Teachers can hardly tell whether a student is weak in combinational design or sequential analysis, or which knowledge points need reteaching for the whole class. Without process data organized by knowledge point, activity, and student, course improvement tends to remain experience-driven.

3. Overall Framework: Knowledge Base, Course Model, AI Applications, and Decision Center

Empowerment should develop around the authentic teaching process rather than the isolated use of a single tool, and can be divided into four layers. The knowledge base is the foundation: teachers associate teaching materials, laboratory guides, simulation cases, examination papers, and common questions with the course, and retrieval-augmented generation reduces generic or hallucinated responses. The course model is its structured representation, reflecting not only chapters but knowledge, competency, and problem levels-counter design, for instance, links to flip-flops, state diagrams, modulus, and clearing conditions. AI applications are the teaching-oriented presentation of the course model, including intelligent lesson preparation, one-click question generation, intelligent learning companions, AI teaching companions, and intelligent grading. The decision center underpins continuous improvement, collecting data on attendance, interaction, courseware learning, assignments, and companion interactions, so that high-frequency questions, completion rates, and self-test accuracy reveal students' status and guide teaching adjustment.

4. Construction Path of the Smart Curriculum

4.1. Dedicated Knowledge Base and Course-Corpus Governance

Construction must first answer what AI relies on when answering and what students rely on when learning. Teachers can build the knowledge base around five categories: basic theory (textbooks, notes, and formula explanations), experiment and practice (Multisim, Proteus, or Quartus cases, 74LS-series materials, laboratory guides, and project reports), extended application (FPGA, domestic chips, and intelligent hardware), evaluation (chapter tests, past papers, and rubrics), and curriculum ideology. Rather than merely uploading files, teachers should govern the corpus: for easily confused points such as asynchronous clearing versus synchronous loading or valid versus invalid states, standard question-answer pairs are established and returned first, turning accumulated teaching experience into reusable resources.

4.2. Knowledge Graph: Course, Competency, and Problem Graphs

The platform builds three graphs. The course graph presents the knowledge system, refining first-level nodes such as combinational logic, flip-flops, and sequential circuits into knowledge points linked to videos, texts, and self-test questions. The competency graph presents expected outcomes through six nodes-logical expression, logic simplification, circuit analysis, circuit design, simulation and debugging, and engineering expression-so that arbitrary-modulus counter design links not only to counter knowledge but to state analysis, control-signal design, and report expression. The problem graph organizes common difficulties from classroom questions and companion logs, letting students enter knowledge through problems and problems through knowledge.

4.3. Intelligent Lesson Preparation, Question Generation, and Learning Companions

Intelligent lesson preparation serves problem-chain design rather than replacing teachers: for a product quality-inspection circuit, students identify variables, build a truth table, simplify the expression, and select gates or a decoder, after which teachers reorganize the content to fit classroom pacing. One-click question generation produces tiered exercises by knowledge point, type, and difficulty, subject to a teacher's second review. The intelligent learning companion provides all-day, traceable tutoring across pre-class, in-class, and after-class stages, while the AI teaching companion supports interaction through instant questions, random roll call, and submissions; hint-based instructions ensure the companion prompts reasoning instead of giving full answers [2, 3].

4.4. Intelligent Grading and Process-Based Evaluation

Intelligent grading conducts preliminary scoring of subjective products such as design explanations, laboratory reports, and project summaries using teacher-defined rules covering requirement analysis, truth-table or state-diagram correctness, expression derivation, simulation support, fault analysis, and formatting, after which teachers review and revise. The total grade combines knowledge-graph learning and pre-class tasks (15%), classroom interaction (15%), chapter tests and assignments (20%), experiments and project reports (30%), and the final examination (20%), so that assessment covers the whole learning process rather than the final examination alone [4].

5. Teaching Case: Arbitrary-Modulus Counter Design Based on 74LS161

Taking a mod-10 counter using 74LS161 as an example, before class teachers associate the counter-design node with the function table, videos comparing synchronous loading and asynchronous clearing, and pre-class quizzes; the platform records viewing time, completed pages, self-test accuracy, and unfinished points, while the companion opens recommended questions on control terminals and clearing-state judgment, letting teachers set priorities from high-frequency questions. During class, using the teaching companion, teachers collect students' judged state sequences and clearing conditions and present typical answers-correct valid states but no explanation of the clearing trigger, a clearing expression ignoring the active-low terminal, or correct simulation without explaining the transient state-then guide discussion along the chain from function table to state sequence, invalid state, clearing logic, and verification.

After class, students submit simulation screenshots, design explanations, and reflections; intelligent grading assesses completeness of state analysis, correctness of clearing logic, and waveform-goal correspondence, and confirmed common errors are added to the knowledge base as question-answer pairs. The learning process thus shifts from listening to worked examples toward discovering problems from data, locating knowledge through the graph, verifying schemes by simulation, and revising expression from feedback.

6. Data-Driven Continuous Improvement

The decision center supports improvement at three levels. Usage-overview data, such as total and average companion use counts and time, show whether AI tools have truly entered learning; low usage prompts teachers to optimize recommended questions or to demonstrate effective use. High-frequency-question data locate difficulties in reverse, so recurring questions on synchronous loading, race hazards, or Karnaugh-map circling are folded into the next class or supplementary micro-lessons. Knowledge-point data on completion, self-test accuracy, and mastery per student support precise assistance, letting teachers push resources to struggling students, reteach low-mastery points, and assign extended tasks such as FPGA implementation or Verilog description to stronger students [5]. This forms a cycle of data collection, problem identification, intervention, resource updating, and re-evaluation [6].

7. Conclusion

A smart curriculum for Digital Electronics Technology should not stop at platform use, but should reconstruct the course around the knowledge structure, the learning process, and competency goals. On an AI teaching platform, a dedicated knowledge base settles what AI relies on, a knowledge graph settles course navigation, intelligent preparation and question generation improve design efficiency, learning and teaching companions strengthen support and interaction, and intelligent grading with decision-center data drive evaluation and continuous improvement. Connecting course resources, knowledge relationships, learning behavior, and evaluation shifts teaching from one-way instruction toward teacher-student-machine collaborative construction, providing a stable and replicable paradigm for applied talent cultivation under emerging engineering education.

Acknowledgments

The authors gratefully acknowledge the financial support from The Key project of University-level education and Teaching reform and research of Beijing Institute of Petrochemical Technology (Smart Curriculum No.XNPSZHKC202503005); and Beijing Higher Education Society: 2025 General Project (Project No. MS2025441).

References

[1] Rospigliosi, P. A. (2024). What is the role of ChatGPT and other large language model AI in higher education? *Interactive Learning Environments*, 32(2), 393–394. https://doi.org/10.1080/10494820.2024.2330836

[2] Das, B., Majumder, M., Phadikar, S., et al. (2021). Automatic question generation and answer assessment: A survey. *Research and Practice in Technology Enhanced Learning*, 16(1). https://doi.org/10.1186/s41039-021-00151-1

[3] Lakshmi, P. S., Simha, J. B., & Ranjan, R. (2024). Empowering educators: Automated short answer grading with inconsistency check and feedback integration using machine learning. *SN Computer Science*, 5(5). https://doi.org/10.1007/s42979-024-02954-7

[4] Villegas-Ch, W., Buenano-Fernandez, D., Navarro, A. M., et al. (2025). Adaptive intelligent tutoring systems for STEM education: Analysis of the learning impact and effectiveness of personalized feedback. *Smart Learning Environments*, 12(1). https://doi.org/10.1186/s40561-025-00389-y

[5] Fazil, M., Risquez, A., & Halpin, C. (2024). A novel deep learning model for student performance prediction using engagement data. *Journal of Learning Analytics*, 11(2), 23–41. https://doi.org/10.18608/jla.2024.7985

[6] Audre, A. A., Basavaiah, J., Patil, C. M., et al. (2026). A pedagogical framework for teaching Verilog-based digital system design using inquiry-based learning. *Journal of Engineering Education Transformations*, 39(S2), 407–415. https://doi.org/10.16920/jeet/2026/v39is2/26049