International Journal of Education and Social Development
ISSN: 3078-9931 (Print), ISSN: 3078-2287 (Online) | Vol. 8, No. 1, 2026

Construction and Practice of a Smart Curriculum for Digital Electronics Technology Oriented Toward Engineering Competency Cultivation

Yadongyang Zhu, Bo Lan and Lanya Yu*

School of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing, China
* Corresponding author

Abstract: Digital Electronics Technology is a core course linking foundational theory and engineering application in electronic information programs. In traditional teaching, students can memorize knowledge points and complete exercises, yet struggle to perform logic modeling, circuit design, simulation verification, and scheme expression in authentic tasks. Aiming at problems such as fragmented knowledge, verification-oriented experiments, delayed feedback, and result-oriented evaluation, this paper proposes a construction path for a smart curriculum oriented toward engineering competency cultivation. Taking competency achievement as the objective, project tasks as the driving force, and knowledge graphs with an AI teaching platform as support, it builds a teaching loop of knowledge-graph-guided learning, project-task driving, intelligent-companion feedback, and data-based diagnostic improvement. Knowledge points become competency nodes and verification experiments become design tasks. The key is not merely adding intelligent tools, but reconstructing content, activities, and evaluation around students' engineering practice ability.

Keywords: Digital Electronics Technology; smart curriculum; engineering competency; project-based teaching; knowledge graph; AI teaching platform.

1. Introduction

Digital Electronics Technology is offered in majors such as electronic information engineering, communication engineering, and automation. It is a foundational course for understanding digital systems and mastering digital logic design methods. Covering number systems, Boolean algebra, combinational and sequential logic, counters, pulse circuits, memories, and programmable logic devices, it possesses both rigorous theoretical logic and strong engineering practicality.

Traditional teaching follows textbook chapters and relies on exercise training and verification experiments. This helps students master basic concepts but easily causes three problems: students remain at chapter-based memorization and miss the relationships among knowledge points; laboratory work is mostly verification-oriented, so students lack training in independent system design; and evaluation is biased toward outcomes. As a result, students often can simplify logic functions but cannot design a complete circuit, and can use simulation software but cannot improve a scheme from the results.

The development of artificial intelligence and smart teaching platforms provides new possibilities for reform [1, 2]. AI platforms can embed intelligent lesson preparation, question generation, knowledge graphs, intelligent learning companions, intelligent grading, and teaching data analysis throughout the process, helping teachers identify difficulties and evaluate performance more continuously [3]. However, if a platform is used only as a resource-distribution and question-answering tool, reform may remain formalistic. Smart curriculum construction should instead begin from engineering competency cultivation and make platform functions serve students' abilities to analyze, design, verify, and express.

2. Teaching Problems from the Perspective of Engineering Competency

2.1. Scattered Knowledge Points and Insufficient Systematic Cognition

The knowledge in this course has strong dependencies: Karnaugh-map simplification underlies combinational logic design, while flip-flop equations, state diagrams, and state tables underlie sequential analysis and counter design. Proceeding by chapter, students easily regard knowledge points as independent items rather than seeing their role in system design. When designing a 74LS161 counter, merely memorizing the mod-N steps leaves a student helpless once the modulus or chip changes, whereas understanding the enable, clearing, and loading terminals supports flexible redesign.

2.2. Experiments Emphasize Operation over Design

Typical experiments often provide explicit circuit diagrams and procedures, so students mainly complete wiring and record data. Such verification consolidates knowledge but does little to develop independent design awareness. Engineering competency requires scheme design from real requirements: designing a quality-inspection circuit demands identifying inputs, determining qualification rules, building a truth table, simplifying functions, and verifying outputs. These abilities emerge only through sustained project-based tasks.

2.3. Discontinuous Learning Feedback

Teachers mainly perceive learning through classroom questions, assignment grading, and laboratory reports, so feedback cycles are long and early misunderstandings go undetected. Some students interpret asynchronous clearing as clearing after the next clock arrives; some cannot distinguish an invalid state from a clearing-trigger state. Process evidence from AI platforms—companion records, courseware completion, self-test accuracy, and high-frequency questions—helps teachers locate difficulties from what students ask and whether they ask repeatedly.

2.4. Result-Oriented Evaluation Lacking Competency Evidence

Evaluation usually relies on the final examination, with usual performance from attendance, assignments, and reports. This distinguishes basic knowledge to some extent but pays insufficient attention to design processes, teamwork, simulation debugging, and reflective expression, so good scores do not guarantee the ability to solve practical problems. Competency-oriented evaluation should value whether students can explain design ideas and state-transition logic, identify problems from simulation results, and present scheme improvement.

3. Objectives and Overall Approach of Smart Curriculum Construction

Construction oriented toward engineering competency should shift from what has been taught to what tasks students can complete. Course objectives span four dimensions: logic modeling—abstracting problems into variables, constraints, and relationships, and establishing truth tables, state tables, or state diagrams; circuit design—selecting gates, chips, or programmable logic devices for combinational and sequential design; simulation and debugging—using tools such as Multisim, Proteus, or Quartus to revise circuits from waveforms and outputs; and engineering expression—explaining requirement analysis, design basis, key logic, and results in standardized reports.

Around these objectives, construction follows the idea of one main line, two graphs, three stages, and four types of data. The one main line drives learning through engineering tasks. The two graphs are the knowledge graph and the competency graph. The three stages are pre-class guidance, in-class inquiry, and after-class enhancement. The four types of data are learning-behavior data, intelligent-companion question-and-answer data, knowledge-point mastery data, and project evaluation data.

4. Construction Path of the Smart Curriculum

4.1. From Knowledge Graph to Competency Graph

Construction must first change how content is presented. Teachers can build a knowledge graph dividing the course into five modules—foundational knowledge, combinational logic, sequential logic, integrated device applications, and integrated system design—each refined into knowledge points linked to micro-lessons, guides, and self-test questions. A competency graph then maps these points to competency objectives, so that students see each knowledge point not only as examination content but as the competency basis for engineering tasks [4, 5].

4.2. Connecting Knowledge Points through Project Tasks

Project tasks are the key carrier, organized at three levels. Basic verification tasks—gate logic testing, flip-flop observation, and decoder verification—emphasize conceptual accuracy and operational standardization. Unit design tasks—majority voter, quality-inspection circuit, and arbitrary-modulus counter—require variable definition, logic modeling, device selection, and simulation verification. Integrated tasks—traffic-light controller, electronic password lock, and digital clock—focus on whether the scheme is reasonable, the structure clear, and the debugging sufficient. Each project marks associated knowledge points and competency objectives, forming a path of task-driven knowledge learning [6].

4.3. Reshaping Pre-class, In-class, and After-class Processes

Before class, teachers publish graph nodes, courseware, micro-lessons, and self-test questions; the platform records viewing time, self-test accuracy, and unfinished points, while the companion opens recommended questions to guide students into class. During class, teachers adjust priorities according to platform data, organizing discussion for high-error points, while the teaching companion supports interaction through submissions, bullet-screen comments, and exercises. After class, students complete tiered assignments, simulation tasks, and reports; the companion gives hint-based tutoring without replacing design work, and intelligent grading provides preliminary assessment before manual review.

4.4. Transforming Companion Data into Teaching-Diagnosis Evidence

The companion's value lies not only in answering questions but in generating diagnosis data. Through the decision center, teachers view high-frequency questions, their types, and entry points. Frequent questions about Karnaugh-map circling indicate shaky simplification rules, and many questions before report submission indicate weak engineering expression. Teachers should organize these into supplementary resources, standard question-answer pairs, and discussion questions, turning individual tutoring into a source for continuous course improvement.

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

Taking a mod-10 counter based on 74LS161 as an example, this topic integrates logic analysis, sequential design, and simulation verification. Before class, teachers associate the counter design node with the 74LS161 function table and micro-lessons on synchronous loading and asynchronous clearing; students write the valid sequence from 0000 to 1001 and judge which state should trigger clearing or loading. During class, after displaying error statistics, teachers organize discussion of how many valid states the counter has, whether 1010 is stable, and which terminals generate the clear signal, then compare three submitted schemes.

After class, students submit simulation files and reports containing design objectives, chip function analysis, state sequence, control logic, and improvement reflection. Intelligent grading conducts a preliminary assessment by rubric—whether the state analysis is complete, the control logic correct, and the reflection demonstrative of the difference between synchronous and asynchronous methods—after which common errors are added to the question-answer pairs. Students thereby experience knowledge invocation, logical reasoning, simulation verification, and engineering expression within one complete task, and the focus shifts from the teacher explaining steps to students explaining why they design this way.

6. Design of a Diversified Evaluation System

Evaluation should emphasize process, competency, and diagnosis, and the total grade can consist of five parts. Knowledge-graph learning and pre-class tasks account for 15%; classroom interaction and problem inquiry for 15%; chapter assignments and unit tests for 20%; experimental projects and design reports for 30%, evaluating logic modeling, circuit design, simulation verification, fault analysis, and engineering expression; and the final comprehensive assessment for 20%, covering core knowledge and integrated application ability.

The platform provides data such as the knowledge-point completion rate, self-test accuracy, companion question records, and intelligent grading results for subjective questions. Teachers should combine these with human judgment rather than fully relying on machine scoring; for reports and design reflections in particular, AI assessment should provide diagnostic clues while teachers review according to course objectives.

7. Summary

A smart curriculum for Digital Electronics Technology should take engineering competency cultivation as its core rather than technology display as its goal. By reconstructing content through knowledge and competency graphs, driving knowledge application through project tasks, supporting learning with intelligent companions, and improving evaluation through intelligent grading and decision-center data, the course can shift from knowledge transmission to a data-driven, task-led, and competency-generating model. In the future, it can establish shared graphs with subsequent courses such as EDA Technology, Microcontroller Principles, and FPGA Design, and accumulate typical projects and student error-case libraries to support engineering practice and innovation ability.

Acknowledgements

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] Wang, Z., & Hu, S. (2017). Teaching reform on digital circuit and logic design course. American Journal of Education and Learning, 2(2), 148–152. https://doi.org/10.20448/804.2.2.148.152.

[2] Skliarova, I. (2021). Project-based learning and evaluation in an online digital design course. Electronics, 10(6), 646. https://doi.org/10.3390/electronics10060646

[3] Zhu, P., Zhong, W., & Yao, X. (2019). Auto-construction of course knowledge graph based on course knowledge. International Journal of Performability Engineering, 15(8), 2228–2236. https://doi.org/10.23940/ijpe.19.08.p23.22282236

[4] Huang, X. (2011). Study of personalized e-learning system based on knowledge structural graph. Procedia Engineering, 15, 3366–3370. https://doi.org/10.1016/j.proeng.2011.08.631

[5] Pagiatakis, G., & Voudoukis, N. (2022). Load lines of transistor circuits: A unifying educational approach using blended learning and flipped classroom. European Journal of Electrical Engineering and Computer Science, 6(2), 73–80. https://doi.org/10.24018/ejece.2022.6.2.397

[6] Perikos, I., Grivokostopoulou, F., & Hatzilygeroudis, I. (2017). Assistance and feedback mechanism in an intelligent tutoring system for teaching conversion of natural language into logic. International Journal of Artificial Intelligence in Education, 27(3), 475–514. https://doi.org/10.1007/s40593-017-0139-y