A Human-Centered Learning and Teaching Framework Using Generative Artificial Intelligence for Self-Regulated Learning Development in Teaching Digital Electronics Course

Jagadeesh Basavaiah[1], Audre Arlene Anthony[2], Naveen Kumar S[3], Mahadevaswamy[4], Naveen Kumar H N[5], Poornima H S[6]

[1,2,4,5] Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru,
[3]Civil Engineering, PES College of Engineering, Mandya,
[6]AI & ML, RNS Institute of Technology, Bengaluru
[1] jagadeesh.b@vvce.ac.in, [2]audre.arlene@vvce.ac.in, [3]naveenkumar@pesce.ac.in, [4]mahadevaswamy@vvce.ac.in, [5]naveenkumarhn@vvce.ac.in, [6]poornima.hs@rnsit.ac.in

***Abstract*—Generative Artificial Intelligence (AI) has emerged as a transformative force in engineering education, offering rapid content generation, intelligent code assistance, and simulation support. In Digital Electronics courses, AI can accelerate circuit design workflows, enhance self-regulated learning (SRL), and support deeper engagement with domain-specific concepts. This paper presents an engineering-adapted Human-Centered Learning and Teaching Framework (HCLTF) that integrates generative AI into the learning process while preserving analytical rigor and human-centered pedagogical values. The framework aligns with SRL's forethought, performance, and self-reflection phases, embedding AI in logic design, HDL development, simulation, and optimization tasks. A case study is presented in which 187 undergraduate students across three institutions design, simulate, and optimize a 4-bit synchronous up/down counter using a combination of traditional design methods and AI-assisted support. The AI was employed for Boolean simplification verification, Verilog code scaffolding, testbench generation, and timing optimization. AI-assisted workflows reduce HDL development time by 40% compared to traditional methods. Synthesis results reveal measurable differences between manual and AI-generated designs, including a 4.2% lower Fmax, 20% higher LUT usage, and a 9.5% increase in dynamic power prior to student-led optimization. Quantitative and qualitative findings indicate improvements in students' Boolean reasoning, debugging proficiency, and reflective judgment across SRL phases. The study highlights the pedagogical value of guided GenAI integration and provides a scalable model for embedding AI-supported learning within Digital Electronics and broader engineering curricula.**

***Keywords*—Digital Electronics; Engineering Education; Generative Artificial Intelligence; HDL Simulation; Human-Centered Learning; Self-Regulated Learning; Verilog**

***ICTIEE* Track—Emerging Technologies and Future Skills**
ICTIEE Sub-Track: AI, Machine Learning, and Digital Tools in Education

I. INTRODUCTION

THE emergence and rapid adoption of Generative Artificial Intelligence (GenAI) in engineering workflows have introduced a paradigm shift in how systems are conceived, modeled, implemented, and optimized. In undergraduate Digital Electronics education, this transformation is particularly impactful because students must bridge the gap between abstract theoretical knowledge—such as Boolean algebra, combinational and sequential logic design, and finite state machine theory—and its practical realization through Hardware Description Languages (HDLs) like Verilog or VHDL, culminating in hardware deployment on Field-Programmable Gate Arrays (FPGAs). The incorporation of GenAI-powered tools into this learning process offers the potential to accelerate concept acquisition, reduce design iteration cycles, and enhance overall design quality, while simultaneously exposing students to workflows increasingly common in professional engineering environments (Andersen et al., 2025).

Traditionally, the workflow of digital electronics education has been sequential, manual, and concept-based: requirement analysis, logic derivation, HDL implementation, simulation and debugging, and ultimately synthesis and deployment. Although this approach develops strong analytical abilities, a keen eye and a solid grasp of ideas, it is also extremely time-consuming. Long debugging periods tend to limit the amount of time students have to devote to higher-order problem solving tasks like analyzing design trade-offs or experimenting with new architectures. As an example, one working on a 4-bit synchronous up/down counter could spend hours figuring out how to fix timing problems in a reset signal, with no time to evaluate anything about Gray code vs. binary state encoding or even to test power-reduction strategies like clock gating.
There are various capabilities that can make this process more effective, and these may be provided by GenAI tools, including

ChatGPT, GitHub Copilot, and code generators specific to HDLs. These are Boolean expression checking, automatic writing of HDL code using natural language specifications, automatic testbench construction including edge-case coverage, optimization hints about timing and resource usage, and even interpretation of synthesis reports with repair advice. But they have their own dangers: when students use AI without critical analysis, they can overlook much in terms of necessary reasoning, forget how to derive things by hand (Đerić et al., 2025), and fail to notice subtle design errors like asynchronous reset hazards or inefficient state encoding that can pass simulation but fail in practice.

The modified HCLTF as applied to undergraduate Digital Electronics is shown in Fig. 1 as a three-circle Venn diagram comprising the Learning Domain, Teaching Domain, and Generative AI Domain. The Learning Domain is associated with activities led by students in setting goals and conceptual knowledge, as well as with iterative self-assessment, which are consistent with the phases of Self-Regulated Learning, including Forethought, Performance, and Self-Reflection. The Teaching Domain reflects the role of the instructor and creates an instructional guide, scaffold, and architectural AI to ensure cognitive participation. The Generative AI Domain encompasses AI tools that verify Boolean expressions, generate HDL programs, create testbenches, and optimise designs, primarily focusing on repetitive or syntactic tasks. The domains overlap shows the important interactions: instructor-led AI integration, student-led AI validation, and designed AI activities. The Design-Test-Reflect Loop at the centre of the diagram signifies the endless nature of the circuit creation, simulation, or hardware testing and reflective improvement process, so that AI is used to improve but not to replace analytical thought.

To ensure that automation complements rather than replaces fundamental design thinking, we adapt the Human-Centered Learning and Teaching Framework (HCLTF) for Digital Electronics education. This adaptation defines a triadic relationship: students act as active designers and evaluators responsible for validating AI outputs; instructors serve as facilitators and cognitive coaches, guiding prompt formulation and critical evaluation; and AI tools function as scaffolding agents, accelerating repetitive or low-level tasks while preserving human responsibility for analytical reasoning, decision-making, and trade-off analysis. This alignment of AI-assisted learning with the phases of Self-Regulated Learning (SRL)—Forethought, Performance, and Self-Reflection—ensures that students cultivate the meta-cognitive skills required to thrive in an AI-augmented engineering environment (Peres et al., 2023).

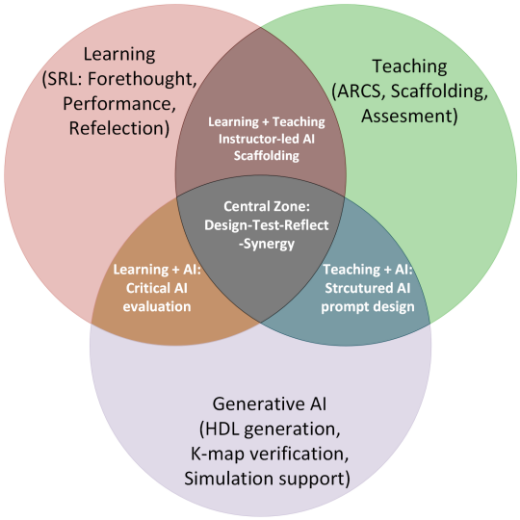

Fig. 1. Adapted HCLTF for undergraduate Digital Electronics

Existing AI-assisted circuit design tools—such as HDL generators, Boolean minimizers, and FPGA optimization assistants—primarily automate tasks but do not cultivate metacognition, SRL, or critical evaluation of AI outputs. This creates a need for a pedagogical framework where AI augments, rather than replaces, analytical decision-making. The HCLTF, originally designed for K–12 STEM settings, emphasizes instructor facilitation, student agency, and iterative reflection, but has limited adaptation for engineering domains involving HDL, synthesis, and FPGA workflows. Unlike TPACK and SAMR, which address technology integration but not learner autonomy or AI collaboration, HCLTF supports student-led verification of AI outputs and structured SRL engagement. Therefore, the adapted HCLTF offers a distinctive, SRL-aligned approach tailored for AI-intensive engineering education.

The differences between the traditional manual workflow and the proposed AI-assisted approach under the HCLTF are summarized in Table I.

TABLE I
DIFFERENCES BETWEEN THE TRADITIONAL MANUAL WORKFLOW AND THE PROPOSED AI-ASSISTED APPROACH UNDER THE HCLTF

Design Stage	Manual Workflow	AI-Assisted Workflow (HCLTF)
Requirement Analysis	Interpret the problem statement manually	Interpret with AI prompting suggestions, but validate the scope
Logic Derivation	Boolean simplification by hand (K-map, Quine–McCluskey)	Manual derivation, AI cross-check for minimization
HDL Implementation	Write Verilog/VHDL from scratch	AI generates draft code; student refines and optimizes
Simulation & Debugging	Manual testbench creation, manual error detection	AI generates testbench, student adds missing cases, and validates results
Optimization & Synthesis	Manual timing/power optimization	AI suggests optimizations, and the student verifies via synthesis tools

Foundational designs such as counters, adders, and simple state machines have well-defined logic rules, predictable HDL structures, and limited design complexity, which make AI-generated outputs easier to verify and compare against known correct behavior. This controlled environment allows students to safely evaluate AI reliability, identify inefficiencies or errors in AI-generated HDL, and develop essential skills in verification and reflective judgment before progressing to more complex digital systems.

This paper applies the adapted HCLTF to a case study in which undergraduate students design, simulate, and optimize a 4-bit synchronous up/down counter with asynchronous reset using both traditional and AI-assisted workflows. The study evaluates efficiency gains, tracks the development of SRL skills, and assesses the quality and reliability of AI-generated HDL in an educational setting. Through this approach, the research demonstrates how structured AI integration can preserve analytical rigor while leveraging the speed and versatility of modern GenAI tools in engineering education.

The organization of the paper is as follows: Section II gives the review of the existing literature, Section III provides the framework of HCLTF for digital electronics, Section IV discusses the impact of implementing HCLTF and provides a discussion on it, Section V provides the conclusion of the work and highlights the future work that can be implemented in near future.

II. Literature Review

Despite the growing use of GenAI in programming, data science, and problem-solving courses, significant gaps remain in the context of engineering education, particularly in digital electronics, HDL design, testbench generation, and FPGA-based workflows. Existing studies focus largely on productivity gains or automated code generation, with limited attention to how GenAI influences conceptual understanding, verification strategies, or student metacognition. Furthermore, most published work does not address transitions between traditional analytical workflows and AI-assisted workflows, leaving unclear how students should critically evaluate AI outputs. To bridge these gaps, this study positions GenAI within a structured, human-centered pedagogical framework to ensure that AI enhances—not replaces—core engineering reasoning skills. Recent advances in AI and GenAI have increasingly shaped engineering education, influencing how students learn programming, circuit design, simulation, and problem-solving. While prior studies demonstrate GenAI's potential to enhance efficiency and provide adaptive guidance, much of the existing work remains centered on programming, algorithmic learning, or Intelligent Tutoring Systems (ITS), with limited focus on hardware-oriented fields such as Digital Electronics (Anderson et al., 1995). These systems have since expanded to GenAI systems that can generate context-sensitive solutions and write code, as well as help to solve complex problems (Amuru & Abbas, 2024). Although most of these developments have been implemented in the context of programming and data science education, an increase in the awareness of the potential of AI in hardware-oriented fields, especially Digital Electronics, is observed.

The classic methodology of teaching digital system design is to proceed through a sequence of manual derivations of logic, the implementation of Hardware Description Language (HDL), simulation, and repeated debugging (Dames & Eves, 2025). Although this process can lead to profound conceptual knowledge, it can be time-consuming and decrease the student's possibilities to think at higher design levels. Recent studies have suggested AI-based design assistants that could be used to automate the process of verification of Boolean logic, produce synthesizable HDL code based on high-level specifications, and synthesize simulation test benches (Song et al., 2023). For example, HDL-specific GenAI models can translate natural language descriptions of digital circuits—such as counters, multiplexers, or finite state machines—into fully functional Verilog modules, often accompanied by verification scripts. However, without careful instructional design, these tools may create black box scenarios where students accept AI-generated outputs without verifying correctness or efficiency.

The challenge of integrating AI into Digital Electronics instruction is closely linked to SRL theory, which emphasizes learner autonomy in setting goals, monitoring progress, and evaluating outcomes (Tang, 2023). Educational research suggests that combining AI support with SRL strategies can maintain student agency while enhancing efficiency. The HCLTF initially applied in K–12 STEM contexts (Zhang et al., 2025) offer a balanced model where instructors act as facilitators, students serve as active evaluators, and AI tools operate as scaffolding agents. However, research on applying this framework to undergraduate Digital Electronics—with its emphasis on hardware timing constraints, synthesis optimization, and FPGA prototyping—is still scarce.

Another critical dimension is AI literacy, defined as the ability to use AI tools effectively, critically assess their outputs, and understand their limitations (Long & Magerko, 2020). In Digital Electronics, AI literacy encompasses evaluating the synthesizability of HDL code, recognizing potential timing violations, and identifying inefficient resource utilization in FPGA implementations. Empirical findings from AI-assisted circuit design studies (Khojah et al., 2025) indicate that, when implemented within a guided framework, AI can reduce development time by up to 50% without a drop in design quality. Conversely, unstructured AI usage often leads to over-reliance and reduced retention of manual design skills.

This review underscores the need to adapt the HCLTF for Digital Electronics to leverage AI's efficiency gains while safeguarding analytical rigor. Embedding AI interaction within SRL phases can provide structured engagement, ensuring that students develop both technical competence and the critical thinking skills necessary for AI-integrated engineering practice.

Table II shows the differences between HCLTF and two widely used educational technology models, TPACK and SAMR.

TABLE II
COMPARISON OF HCLTF WITH SAMR AND TPACK MODELS

Aspect	TPACK	SAMR	HCLTF (Proposed Framework)
Primary Focus	Balancing technology, pedagogy, and content knowledge	Classifying levels of technology integration	Human-centered learning with AI-assisted scaffolding
Learner Autonomy	Not explicitly emphasized	Not addressed	Strong emphasis on student agency and decision-making
Role of AI	Not considered	Not considered	AI treated as a cognitive collaborator to support reasoning
Self-Regulated Learning (SRL)	Not integrated	Not integrated	Fully aligned with SRL phases (Forethought, Performance, Reflection)
Critical Evaluation of AI Outputs	Not included	Not included	Students are required to verify, critique, and refine AI suggestions

To further clarify the theoretical basis of the proposed approach, a new subsection has been added highlighting how the adapted HCLTF differs from existing AI-assisted learning models. Unlike Intelligent Tutoring Systems, which centralize decision-making and automate feedback, or HDL automation tools that directly generate code from specifications, the adapted HCLTF positions AI as a scaffolding partner rather than an automation engine. Moreover, the framework explicitly maps AI support onto the Forethought, Performance, and Self-Reflection phases of SRL, offering a pedagogical integration does not present in conventional engineering AI tools. This distinction underscores the unique value of HCLTF in supporting reasoning-driven, AI-integrated learning environments.

III. FRAMEWORK DESIGN: HCLTF FOR DIGITAL ELECTRONICS

The HCLTF is an instructional model that integrates best pedagogical practices, SRL principles, and GenAI tools into a cohesive learning ecosystem. In the context of Digital Electronics education, its goal is to create a balanced workflow where automation accelerates repetitive tasks. At the same time, human reasoning remains central to problem-solving, design validation, and critical decision-making (Brenner, 2022).

A. Core Components of the HCLTF

The adapted framework consists of three interconnected domains:

1. *Learning Domain* – This represents the student's active cognitive process, mapped directly to the three phases of SRL:
   - *Forethought* – Students set learning objectives, identify constraints (e.g., timing < 50 ns, minimal gate count), and plan their design approach.
   - *Performance* – Students engage in design activities, simulation, optimization, and AI-assisted verification.
   - *Self-Reflection* – Students analyze simulation results, compare manual and AI-generated designs, and extract lessons learned for future tasks.
2. *Teaching Domain* – Instructors act as facilitators rather than sole knowledge transmitters. Their role includes:
   - Designing project-based tasks that require AI integration without replacing critical thinking.
   - Guiding students on how to formulate effective AI prompts for HDL design.
   - Ensuring that learning outcomes target both technical mastery and critical AI literacy.
3. *Generative AI Domain* – This represents the toolset that supports learning, including:
   - Boolean simplification checkers.
   - HDL code generators.
   - Testbench creators.
   - Optimization advisors for timing, power, and area. AI operates as a scaffolding agent—offering suggestions, templates, and optimizations—while the student decides what to adopt, modify, or discard.

B. Interactions Between Domains

The three domains overlap in ways that are critical to effective AI integration:

- *Learning + Teaching* – Instructor-guided project design where AI tools are introduced at specific workflow stages to promote conceptual understanding before automation.
- *Learning + AI* – Students use AI to validate manual calculations (e.g., Karnaugh map minimization) but still perform the derivation themselves to reinforce logic theory.
- *Teaching + AI* – Instructors design scaffolded activities where AI outputs contain intentional

ambiguities or inefficiencies, encouraging students to identify and correct them.

- *Central Overlap* – All three domains converge when students **design**, **test**, and **reflect** using both human reasoning and AI support, ensuring an iterative improvement cycle.

C. Alignment with SRL in Digital Electronics

The adapted HCLTF maps directly onto the three SRL phases—Forethought, Performance, and Self-Reflection—ensuring that Generative AI integration supports, rather than bypasses, critical learning stages.

Forethought Phase – This is the planning and goal-setting stage. In a Digital Electronics course, students begin by analyzing design specifications, identifying constraints (e.g., maximum propagation delay, power budget, or logic minimization requirements), and selecting a high-level design strategy. Here, AI can be used as a brainstorming partner, offering possible block diagrams, design architectures, or logic flow suggestions. However, students must critically evaluate these proposals for feasibility, correctness, and adherence to course constraints.

Performance Phase – This is the execution and monitoring stage, where students engage in the actual design and implementation process. They perform Boolean simplification, write HDL code, simulate designs, and run synthesis. AI support during this phase might include generating initial Verilog/VHDL code templates, verifying manual Karnaugh map simplifications, or producing a baseline testbench. Crucially, students must still engage in manual derivation and simulation analysis to avoid over-reliance on AI.

Self-Reflection Phase– This is the evaluation and improvement stage, where students review simulation waveforms, analyze synthesis reports, and compare manual designs against AI-assisted versions. AI can assist in producing performance comparison tables or suggesting optimizations, but the interpretation and final design decisions remain with the student (Vosniadou et al., 2024). This phase reinforces metacognitive skills, encouraging students to identify what worked, what failed, and how to improve future projects.

This mapping ensures that AI is embedded strategically within each SRL phase, reinforcing rather than replacing student agency, and helping learners develop both technical competence and critical AI literacy. Table III illustrates how the SRL phases align with specific activities in a Digital Electronics course and how Generative AI supports each stage. In the Forethought phase, students plan their design by defining specifications and constraints, with AI suggesting possible architectures for review. During the Performance phase, AI assists in generating draft HDL code, verifying Boolean simplifications, and recommending timing optimizations while students carry out core design and simulation tasks. In the Self-Reflection phase, AI assists in creating performance comparison tables; however, students are responsible for interpreting the results and making the final design decisions.

TABLE III
INTEGRATION OF GENERATIVE AI WITHIN SRL PHASES FOR DIGITAL ELECTRONICS COURSE

SRL Phase	Digital Electronics Activity	AI Integration
Forethought	Define design specs, identify performance constraints.	AI suggests possible architectures or block diagrams for consideration.
Performance	Logic derivation, HDL coding, simulation, optimization.	AI generates draft HDL, verifies Boolean minimization, and proposes timing improvements.
Self-Reflection	Compare manual vs. AI solutions and analyze synthesis results.	AI assists in generating performance comparison tables, but conclusions are student-derived.

Fig. 2 depicts the sequential workflow of a Digital Electronics design process aligned with the phases of SRL. It begins with defining design specifications (Forethought), followed by logic design, HDL coding, simulation, and optimization (Performance), and concludes with result evaluation and improvement through reflection.

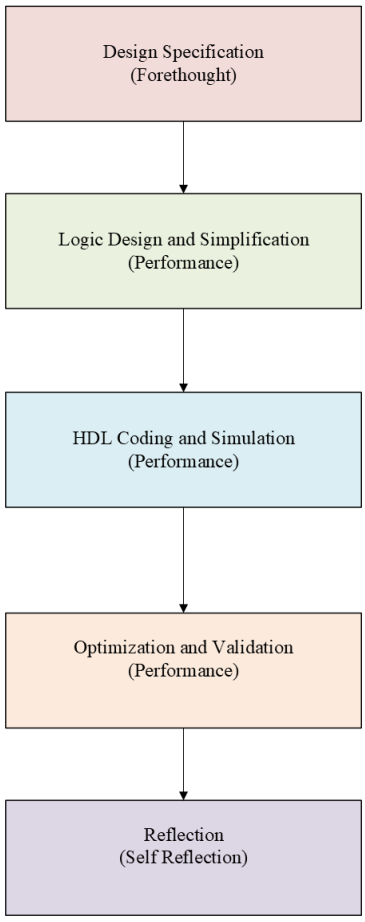

Fig. 2. Sequential workflow of SRL Phases in Digital Electronics Course

The adapted HCLTF is integrated into the Digital Electronics laboratory sequence through three structured AI-supported modules. Week 3 introduces AI-assisted exploration of combinational logic, including Boolean simplification and truth

table verification. Week 6 extends AI use to sequential circuit design, focusing on timing behavior, state progression, and HDL synthesis. Week 9 centres on testbench creation, waveform interpretation, and debugging using AI-guided refinement. To ensure effective implementation, instructors undergo short workshops on prompt engineering, monitoring SRL behaviours, and ethical and responsible AI practices. The required resources include Artix-7 FPGA boards, Vivado/ModelSim for synthesis and simulation, VS Code with GitHub Copilot for AI-assisted coding, and institutional guidelines that govern appropriate and secure student access to GenAI tools.

D. Example Workflow: 4-bit Synchronous Up/Down Counter

To demonstrate the practical application of the HCLTF in an undergraduate Digital Electronics setting, we conducted a case study involving the design, simulation, and optimization of a 4-bit synchronous up/down counter with asynchronous reset. This case study demonstrates how the framework aligns SRL phases with the integration of GenAI, ensuring that students maintain active control over the design process while benefiting from AI-driven efficiency gains.

1. Problem Specification (Forethought Phase)

The design objective was to implement a 4-bit synchronous counter that:

- Supports both up-counting and down-counting modes.
- Includes an asynchronous reset that clears the counter to 0000.
- Operates at a maximum clock frequency of 20 MHz or higher when implemented on a target FPGA board (Xilinx Artix-7).

At this stage, students manually interpret the problem statement to extract functional and performance requirements. GenAI is introduced only as a support tool—providing potential block diagram layouts and suggesting alternative design approaches, such as ripple counters vs. fully synchronous counters. The students evaluate each AI-suggested architecture, ultimately selecting a fully synchronous design to meet timing requirements.

2. Logic Derivation and HDL Development (Performance Phase)

2.1 Manual Derivation

Students begin with the state transition table for the 4-bit counter, considering both up and down modes. They derive the next state equations for each flip-flop using Karnaugh maps. For example, for the least significant bit (Q_0), the next state toggles every clock pulse, whereas higher bits toggle only when all lower bits meet specific conditions.

2.2 AI Verification

Once the manual minimization is complete, students use a GenAI tool to verify their Boolean expressions. The AI cross-checks Karnaugh map results, flags redundant logic, and suggests minimal forms where applicable. This serves as a conceptual safeguard—ensuring students spot simplification errors early without replacing their manual effort.

2.3 HDL Coding

Students proceed to implement the counter in Verilog HDL. They manually write the core module but use AI to:

- Suggest syntactic corrections.
- Generate reusable parameterized modules.
- Provide a draft testbench including clock generation, reset application, and mode switching scenarios.

3. Simulation and Debugging (Performance Phase)

Students simulate their design using ModelSim or Vivado Simulator. They examine waveform outputs to confirm:

- Correct counting sequences in both modes.
- Proper reset behavior.
- No metastability or glitching in mode control.

AI assistance here is limited to diagnostic hints—for instance, suggesting where a reset signal may need synchronization or proposing minor code refactoring for cleaner simulation output.

4. Synthesis, Optimization, and Hardware Testing (Self-Reflection Phase)

4.1 Comparative Synthesis

Students synthesize both their manually designed counter and the AI-generated counter on the same FPGA platform. They compare key metrics from the synthesis report, such as:

- Maximum clock frequency (Fmax)
- Logic utilization (LUT count)
- Flip-flop count
- Dynamic power consumption

4.2 Observations and Refinement

In this case, the AI-generated version introduced an unnecessary combinational enable logic, resulting in a slightly lower Fmax compared to the manual design. Recognizing this inefficiency, students modified the AI code to streamline the enable path, achieving performance parity while maintaining low resource usage.

4.3 Hardware Verification

The optimized design was then programmed onto an Artix-7 FPGA development board. Students validated real-time performance by connecting LEDs to the counter outputs and

observing correct count sequences under varying mode and reset inputs.

Through this workflow, the HCLTF ensures a balanced integration of human reasoning and AI assistance. Students continue to perform the core logical derivations manually, which helps them retain a strong grasp of fundamental concepts and reinforces their ability to reason through design problems independently. At the same time, AI is leveraged to accelerate repetitive and time-consuming tasks, such as generating HDL boilerplate code and creating initial testbench structures, allowing students to focus more on problem-solving and optimization. Ultimately, the critical analysis, interpretation of results, and final design decisions remain firmly under human control, ensuring that students develop both technical mastery and the judgment required to evaluate and refine AI-generated outputs.

A total of 187 second-year Electronics and Communication Engineering (ECE) students participated in this study, all of whom were enrolled in the Digital Electronics course across three higher-education institutions. Most students had limited prior exposure to HDL design and no formal experience using Generative AI tools for circuit development. Before beginning the activity, all participants received a brief orientation on responsible AI use, verification practices, and expectations related to Self-Regulated Learning. Participation was voluntary, and no personally identifiable information was collected. The demographics of the participants are given in Table IV. No student data is uploaded to external AI systems, ensuring full privacy and anonymization throughout the study. Academic integrity is maintained through strict guidelines that prohibit fully AI-generated submissions and require all students to complete manual derivations before engaging with AI tools. To avoid over-reliance, students are explicitly informed about AI limitations, potential biases, and the non-deterministic nature of model outputs, ensuring that AI supports—but does not replace—conceptual understanding and human-led reasoning.

TABLE IV
DEMOGRAPHICS OF PARTICIPANTS

Category	Details
Total Participants	187 students
Program / Branch	Electronics and Communication Engineering (ECE)
Year of Study	Second-year
Course Enrolled	Digital Electronics
Prior GenAI Experience	81% had no prior exposure
Prior HDL Experience	23% had experience beyond basic lab exercises
HDL Proficiency Levels	62% Beginner, 29% Intermediate, 9% Advanced
Pre-Activity Orientation	Responsible AI use, verification practices, SRL guidelines
Ethical Considerations	Participation voluntary; no personal data collected

The adapted HCLTF includes embedded safeguards to prevent AI over-dependence, such as requiring manual Boolean derivation before any AI assistance, introducing AI only after human design decisions, and using reflection sheets to compare manual and AI-generated code. Instructor-controlled prompts and restricted AI use during core assessments ensure that students verify, critique, and refine AI outputs rather than rely on them unquestioningly.

The selection of the 4-bit synchronous up/down counter is pedagogically intentional. This circuit provides a level of controlled complexity that allows the study to isolate and analyze the effects of Generative AI assistance without introducing unrelated design variables. It also enables reliable, consistent comparisons between manual and AI-assisted workflows, and serves as a fundamental HDL building block that forms the basis for more advanced digital systems. To demonstrate that the adapted HCLTF scales beyond introductory designs, we additionally describe planned extensions to more complex modules, including finite state machines (FSMs) with multi-signal control, arithmetic logic units (ALUs), basic pipelined processor stages, and small VLSI datapath components. These extensions show that although the present study focuses on a foundational design, the framework is structured for broader applicability across progressively sophisticated digital design tasks.

IV. RESULTS AND DISCUSSIONS

The application of the HCLTF to the design of a 4-bit synchronous up/down counter provided an opportunity to evaluate both the technical impact of AI assistance and the pedagogical outcomes for students. This section presents quantitative results, qualitative observations, and a discussion of implications for undergraduate Digital Electronics course.

A. Quantitative Results

1. HDL Development Time

A comparison of manual and AI-assisted workflows revealed a significant reduction in *design and coding time* when GenAI tools were used. Manual design—covering Boolean simplification, Verilog coding, and testbench creation—took approximately 4.5 hours on average. In contrast, the AI-assisted process required only 2.7 hours, a 40% time saving. The reduction was primarily due to AI-generated HDL boilerplate and automated testbench scripts, which allowed students to focus on debugging and optimization rather than syntax writing. Fig. 3 compares HDL development time, showing that AI-assisted design reduced implementation time from 4.5 hours to 2.7 hours.

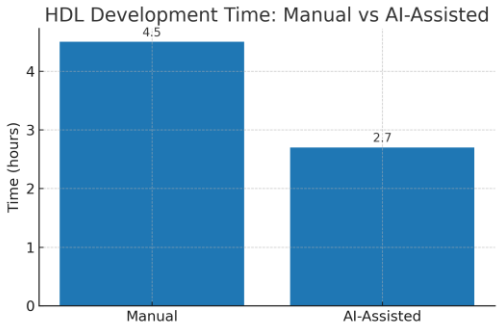

Fig. 3. HDL Development Time comparison

2. Synthesis and Hardware Performance

The counter designs—manual and AI-assisted—were synthesized on a Xilinx Artix-7 FPGA using Vivado. Table V summarizes the synthesis results.

TABLE V
COMPARISON OF SYNTHESIS METRICS FOR MANUAL VS AI-ASSISTED HDL DESIGNS

Metric	Manual Design	AI-Assisted Design	Difference
Max Clock Frequency (Fmax, MHz)	123.4	118.2	-4.2%
LUT Usage	20	24	+20%
Flip-Flop Usage	16	16	0%
Dynamic Power (mW)	2.1	2.3	+9.5%

Fig. 4 presents synthesis results, indicating that while AI-assisted designs slightly decreased maximum clock frequency and increased LUT usage and dynamic power, the differences remained within acceptable design margins.

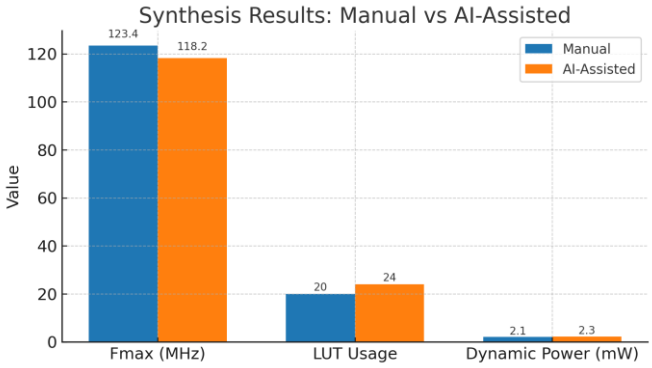

Fig. 4. Synthesis Performance Comparison of Manual and AI-Assisted HDL Designs

The AI-assisted design showed a slightly lower Fmax and marginally higher LUT usage due to unnecessary combinational logic in the enable path. This overhead was eliminated after student-led optimization, resulting in equivalent performance between the two approaches.

A paired t-test was conducted to statistically validate the difference between manual and AI-assisted development times, confirming that the observed reduction is significant ($p < 0.001$).

3. Simulation Validation

Both designs passed functional simulation, but the AI-generated version initially contained redundant reset logic that increased propagation delay during reset deassertion. Students identified and corrected this issue during the Self-Reflection phase, demonstrating the value of critical post-analysis even in AI-supported workflows.

B. Qualitative Observations

1. Student Reflections

Students reported that AI assistance was most valuable in generating initial testbenches, suggesting syntax corrections, and providing optimization hints. However, they also acknowledged the risk of becoming overly dependent on AI-generated HDL, especially for logic derivation. Several students noted that they felt more confident after verifying AI outputs themselves, as it reinforced their conceptual understanding (Basavaiah et al., 2022).

2. Instructor Feedback

From the instructor's perspective, the adapted HCLTF successfully maintained student engagement in core analytical tasks while using AI to streamline repetitive processes. However, instructors stressed the need for structured prompts that ensure students cannot bypass essential derivation and reasoning steps (Dewan et al., 2025).

3. Discussion of Educational Implications

The results suggest that AI-assisted workflows can provide measurable efficiency gains in Digital Electronics education when structured within the HCLTF. The 40% reduction in development time allowed students to spend more hours on higher-order tasks—such as performance optimization, exploring alternative architectures, and conducting in-depth simulation analysis—without sacrificing conceptual mastery.
However, the findings also reinforce that AI integration is not a plug-and-play enhancement. Without the guiding structure of the HCLTF, there is a significant risk that students could delegate critical reasoning to AI tools, leading to superficial understanding and reduced long-term problem-solving ability.
When implemented properly, this framework offers a balanced model:

- AI handles the mechanical tasks of syntax generation and initial verification.
- Students retain ownership of analytical thinking, debugging, and optimization.
- Instructors curate AI interactions to maintain cognitive engagement and learning depth.

These results have broader implications beyond this case study, suggesting that the AI + HCLTF model could be adapted to courses in VLSI Design, Embedded Systems, and Computer Architecture, where HDL development and performance optimization are core activities.

Table VI shows the summary of Pre- and Post-Intervention Learning Outcomes

TABLE VI
SUMMARY OF PRE- AND POST-INTERVENTION LEARNING OUTCOMES

Learning Outcome Measured	Pre-Intervention Score	Post-Intervention Score	Improvement
Boolean Simplification Accuracy	62%	84%	+22%
HDL Debugging Accuracy	71%	88%	+17%
Synthesis Interpretation Skills (timing, LUT, power)	39%	76%	+37%
Ability to Detect Inefficient or	44%	81%	+37%

Redundant AI-Generated Logic			
SRL: Goal-Setting	56%	88%	+32%
SRL: Monitoring / Performance Control	52%	81%	+29%
SRL: Reflection Quality	49%	90%	+41%

A structured survey was conducted with 187 undergraduate students enrolled in Digital Electronics courses across three engineering institutions to evaluate the integration of Generative AI within the HCLTF. The participants, drawn from second-year cohorts, had foundational knowledge of logic design and Hardware Description Languages (HDL), with varied experience in AI-assisted tools. The questionnaire focused on three dimensions: perceived benefits of AI integration, potential challenges, and overall learning impact. Each of the eight statements was rated on a five-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree).

The survey was administered during the final two weeks of the semester, after students completed a 4-bit synchronous counter design project incorporating AI-based HDL generation and optimization. Both online and paper-based formats were used, and participation was anonymous to encourage honest feedback. Pilot testing with 15 students ensured clarity, and internal consistency measured via Cronbach's alpha was **0.87**, indicating strong reliability. A total of 187 valid responses were obtained (96.4% completion rate), and descriptive statistics were computed to analyze trends in perceptions and identify relationships between prior AI experience and reported learning outcomes.

Table VII summarizes student perceptions of AI-assisted learning in Digital Electronics, showing high agreement on time savings, improved problem-solving focus, and overall learning enhancement. Moderate concerns were noted regarding over-reliance and occasional inefficiencies in AI-generated designs.

TABLE VII
STUDENT SURVEY RESULTS ON AI-ASSISTED LEARNING IN DIGITAL ELECTRONICS (N = 187)

Q.No	Survey Question	Mean Score	Std. Dev.	Agreement (%)
1	AI-assisted workflow helped me complete designs faster	4.58	0.52	93.6%
2	AI allowed me to focus more on complex problem-solving	4.42	0.59	88.2%
3	AI-generated HDL code was easy to understand and edit	4.14	0.66	81.3%
4	AI improved my Boolean simplification and verification accuracy	4.33	0.54	85.6%
5	The framework encouraged me to critically evaluate AI outputs	4.05	0.73	78.6%
6	I became over-reliant on AI for certain design tasks	3.21	0.96	42.8%
7	AI sometimes produced inefficient designs	3.29	0.84	46.0%
8	Overall, AI integration improved my learning experience	4.49	0.48	91.4%

Over 90% of students reported that AI integration accelerated design work and enhanced their overall learning experience, with approximately 88% noting that it allowed them to dedicate more time to higher-order problem-solving activities. Moderate concerns were expressed regarding over-reliance (42.8%) and occasional inefficiencies in AI-generated designs (46%), reflecting an awareness of the technology's limitations. Furthermore, the high agreement rate on the ease of understanding AI-generated HDL code (81%) indicates that the implemented workflow successfully balanced automation with the retention of core conceptual and practical skills.

While the findings of this study demonstrate clear improvements in efficiency, reasoning accuracy, and SRL development, certain limitations must be acknowledged. First, the work is not longitudinal; students were assessed within a single semester, limiting insights into long-term retention and transfer of skills. Second, although the study included 187 students across three institutions, its scope is confined to foundational digital design tasks, and broader validation across more complex courses is still required. Third, AI-generated HDL behavior may vary with different prompting or model versions, requiring instructor oversight to ensure consistency. These limitations inform and motivate the planned longitudinal extension of this research.

V. CONCLUSION AND FUTURE WORK

This study presented an adaptation of the HCLTF for undergraduate Digital Electronics education by integrating Generative AI into the design, simulation, and optimization workflow. Using the 4-bit synchronous up/down counter as a case study, the framework demonstrated how AI support can be aligned with SRL phases so that students maintain ownership of analytical reasoning and manual derivation while benefiting from AI-assisted verification and code scaffolding. The results showed a substantial reduction—approximately 40%—in HDL development time, enabling students to focus on higher-order tasks such as optimization and synthesis interpretation. Although AI-generated HDL occasionally introduced redundant logic or minor inefficiencies, these were effectively identified during the self-reflection phase, confirming that the framework strikes a balance between AI efficiency and

conceptual understanding.

From a pedagogical standpoint, the adapted HCLTF ensures that students remain active decision-makers throughout the design process while AI serves as a supportive cognitive partner. Instructors guide the process through structured safeguards, including mandatory manual derivation, staged AI integration, and reflection-based validation. This approach prevents over-reliance and encourages deeper engagement with design concepts, making it a viable model for modernizing Digital Electronics education. Future work will extend the framework to more complex digital systems such as FSMs with multiple control signals, ALUs, pipelined stages, and small VLSI datapath components. Additional lines of work include integrating AI directly into FPGA toolchains for real-time optimization and evaluating how AI-augmented learning environments influence graduates' technical competence, ethical awareness, and industry readiness.

Despite the promising outcomes, the study has certain limitations. The framework was tested on a foundational circuit, which may constrain generalizability to more complex architectures. AI tools sometimes generated suboptimal HDL, requiring instructor oversight and student diagnostic skills, and SRL development varied across learners. To address these gaps, a structured longitudinal research plan has been proposed. This includes Year 1–2 assessments of SRL retention using non-AI tasks, Year 2–3 tracking of student performance in advanced digital design courses, internship supervisor evaluations, employability metrics for HDL/FPGA roles, and 6- and 12-month conceptual retention tests. An AI-Competence Growth Index (AI-Lit Index) will also be used to measure sustained progress, providing a comprehensive roadmap for evaluating the long-term educational impact of the adapted HCLTF.

REFERENCES

Creswell, J. W. (2012). Educational research: Planning, conduAndersen, J. P., Degn, L., Fishberg, R., Graversen, E. K., Horbach, S. P. J. M., Schmidt, E. K., Schneider, J. W., & Sørensen, M. P. (2025). Generative artificial intelligence (GenAI) in the research process: A survey of researchers' practices and perceptions. Technology in Society, 81, 102813. https://doi.org/10.1016/j.techsoc.2025.102813

Đerić, E., Frank, D., & Milković, M. (2025). Trust in generative AI tools: A comparative study of higher education students, teachers, and researchers. Information, 16(7), 622. https://doi.org/10.3390/info16070622

Peres, R., Schreier, M., Schweidel, D., & Sorescu, A. (2023). On ChatGPT and beyond: How generative artificial intelligence may affect research, teaching, and practice. International Journal of Research in Marketing, 40(2), 269–275. https://doi.org/10.1016/j.ijresmar.2023.03.001

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. Journal of the Learning Sciences, 4(2), 167–207. https://doi.org/10.1207/s15327809jls0402_2

Amuru, D., & Abbas, Z. (2024). AI-assisted circuit design and modeling. In AI-enabled electronic circuit and system design (pp. 1–40). https://doi.org/10.1007/978-3-031-71436-8_1

Dames, C., & Eves, M. (2025). Digital design principles and practices. Cammy Fetchens LLC.

Song, F., Xu, W., & Wang, J. (2023). Design and research of automatic generation control code for dual three-phase PMSM based on FPGA. In 2023 26th International Conference on Electrical Machines and Systems (ICEMS) (pp. 1917–1921). https://doi.org/10.1109/icems59686.2023.10345207

Tang, H. (2023). Examining self-regulated learner profiles in MOOCs: An item response theory perspective. AERA 2023. https://doi.org/10.3102/ip.23.2018003

Zhang, S., Jaldi, C. D., & Schroeder, N. L. (2025). Learning and teaching with AI in STEM education: An umbrella review. https://doi.org/10.31219/osf.io/472nm

Long, D., & Magerko, B. (2020). What is AI literacy? Competencies and design considerations. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (pp. 1–16). https://doi.org/10.1145/3313831.3376727

Khojah, R., Werth, A., Broadhead, K. W., et al. (2025). Integrating generative artificial intelligence tools and competencies in biomedical engineering education. Biomedical Engineering Education, 5, 135–151. https://doi.org/10.1007/s43683-025-00175-9

Brenner, C. A. (2022). Self-regulated learning, self-determination theory and teacher candidates' development of competency-based teaching practices. Smart Learning Environments, 9(3). https://doi.org/10.1186/s40561-021-00184-5

Vosniadou, S., Bodner, E., Stephenson, H., et al. (2024). The promotion of self-regulated learning in the classroom: A theoretical framework and an observation study. Metacognition and Learning, 19, 381–419. https://doi.org/10.1007/s11409-024-09374-1

Basavaiah, J., Anthony, A. A., & Patil, C. M. (2022). Transformation of engineering education in India through student-centric learning approach. Wireless Personal Communications, 124, 489–497. https://doi.org/10.1007/s11277-021-09370-7

Dewan, U., Hingle, A., McDonald, N., & Johri, A. (2025). Engineering educators' perspectives on the impact of generative AI in higher education. In 2025 IEEE Global Engineering Education Conference (EDUCON) (pp. 1–10). https://doi.org/10.1109/EDUCON62633.2025.11016518