Academic Education Publishing House

Higher Education and Practice Vol. 3 No. 8, 2026

Cultural Symbols as Cognitive Scaffolding: A Case Study of Integrating the *I Ching* into Binary Number System Conversion Instruction

Yibing Lu*, Yawei Hu, Jing Xing, Yingxi Li

School of Electronic Information Engineering, Xi'an Peihua University, Xi'an, Shaanxi, China

**Corresponding Author*

Abstract: Students can easily memorize operational rules for binary conversion in classroom learning, yet they often fail to comprehend its underlying working principles. This deficiency frequently leads to incorrect solutions when they handle variant problems. To improve this situation, this study develops a three-dimensional instructional framework that integrates cultural narration, algorithmic programming, and engineering practice. The framework takes the symbolic system of The I Ching as a cognitive support tool. Based on scaffolding theory and structure-mapping theory, the present teaching design constructs structural correlations between binary bit expansion rules and the hierarchical dichotomy logic of The I Ching. The classical cultural logic that Taiji generates the Two Forms, the Two Forms generate the Four Images, and the Four Images generate the Eight Trigrams perfectly matches the progressive expansion mechanism of binary numbers. This approach transforms traditional cultural elements from superficial classroom decoration into effective cognitive tools for abstract concept learning. Tiered programming exercises and targeted classroom implementation strategies are further adopted to consolidate teaching outcomes. This study also establishes three practical evaluation indicators to differentiate shallow cultural embedding from in-depth structural integration. The indicators cover structural mappability, instructional effectiveness, and boundary interpretability. Clear boundary interpretation is incorporated into formal instructional procedures to avoid student misunderstanding of structural analogy as historical origination. A preliminary teaching experiment was conducted among 53 students from two parallel classes of Electronic Information Engineering at Xi'an Peihua University. The core programming task completion rate reached 85 percent, and the overall in-class quiz accuracy approached 81 percent. Classroom observation reveals that boundary interpretation produces optimal instructional effects when delivered after students have established stable analogical cognition. This research provides a reusable and transferable instructional paradigm for integrating traditional indigenous culture into fundamental computer science education.

Keywords: Computing Education; Number System Conversion; STEM; Cognitive Scaffolding; Culturally Responsive Computing; I *Ching*

1. Introduction

Engineering education generally adopts C programming for introductory courses. Teaching C programming to absolute beginners, however, comes with well-known difficulties [1,2]. Within C language coursework, binary conversion is often the first fully abstract concept students encounter. It covers multiple symbol systems including binary, octal and hexadecimal notations, together with a complete set of conversion procedures. Systematic misconceptions about number representation therefore constitute a common learning barrier for new learners [3]. In traditional classes, students usually memorize conversion rules by rote. They can hardly grasp the inherent relationships between different number bases or their practical value in engineering scenarios.

Integrating fine historical and cultural heritage into STEM and engineering education has drawn growing attention from researchers worldwide. Existing studies show that culturally responsive computing pedagogy can boost students' motivation, strengthen their sense of identity and belonging, and encourage active classroom engagement [4,5]. In practice, nevertheless,

cultural integration faces two main challenges. One is superficial embedding: cultural materials merely serve as lesson openers or decorative add-ons, disconnected from core knowledge points [6]. The other is cultural overshadowing, where cultural narratives take up class time reserved for technical instruction. Hence, cultural integration should start with exploring deep structural parallels between cultural themes and technical knowledge. In this way, culture functions not as ornamentation, but as cognitive scaffolding [7].

The ancient Chinese text The I Ching constructs its symbolic system through the binary partition of yin and yang lines, forming a hierarchical generative logic: Taiji generates the Two Forms, the Two Forms generate the Four Images, and the Four Images generate the Eight Trigrams. This stepwise binary division shares formal structural similarities with the bit-expansion rules of binary counting [8]. Current scholarly discussions on this relationship remain mostly theoretical or historical. Classroom lesson designs built upon this logic are scarce, and rigorous empirical evaluations are even less common.

Three major obstacles hinder the integration of cultural elements into binary instruction. First, existing research on binary conversion pedagogy predominantly centers on visualization tools and gamified learning activities. Few studies exploit learners' familiar cultural symbol systems to provide cognitive scaffolding for abstract concepts [9]. Second, while abundant lesson plans exist for culturally responsive computing, frontline educators lack actionable criteria to distinguish superficial cultural ornamentation from deep structural integration, which complicates the selection of appropriate cultural resources [10]. Third, cross-domain analogies deployed in classrooms tend to confuse students regarding the historical connections between concepts. Nevertheless, computing education research rarely embeds boundary declaration as an inherent component of lesson design. Most instructors only remedy such misconceptions after they emerge [11,12].

Against this backdrop, this study targets computational thinking and programming competency as core training goals in engineering education, exploring classroom practices for binary instruction with cultural integration. Leveraging the structural parallels between the symbolic system of The I Ching and binary counting, we develop a practical curriculum framework and instructional design for binary number system conversion that unites cultural narration, algorithmic programming, and engineering practice. This paper proposes three operational criteria for screening cultural themes suitable for deep integration into computing courses: structural mappability, instructional functionality, and boundary declarability. It further presents preliminary observations from the first round of classroom implementation, focusing on a prominent phenomenon: students tend to spontaneously misinterpret structural analogy as historical origin, and we examine how explicit boundary declarations mitigate this misconception. This work is explicitly framed as a design report and makes no causal claims regarding the instructional effectiveness of the proposed pedagogy.

2. Research Background and Theoretical Framework

2.1 Teaching Binary Number System Conversion

Binary number system conversion is a foundational concept in computing education; its learning difficulties stem from the abstractness of the concept and the fragmented nature of conversion rules. Existing instructional practices fall into three categories: visualization tool–assisted instruction; pedagogical model innovations such as guided discovery learning, task-driven teaching, and gamified instruction; and error analysis targeting typical misconceptions. Most of these practices focus on technical knowledge itself and rarely mobilize symbol systems already familiar to learners from their own cultural backgrounds as scaffolds for learning abstract knowledge. Teaching binary concepts is of significant educational value for establishing foundational computational understanding [13].

2.2 Culturally Responsive Computing and its Limitations

The core premise of culturally responsive computing is that instruction that establishes substantive connections with learners' cultural experiences tends to improve learning outcomes. A number of design and empirical studies, such as computational thinking tasks situated in local cultural contexts and e-readers rooted in indigenous cultures—report better classroom experience and engagement. Much of the existing research on integrating cultural symbols into

classroom instruction remains confined to instructional design cases and reflective summaries. Many practices merely pair knowledge points with cultural anecdotes and fail to achieve structural integration [14]. The field still lacks actionable criteria to assess whether a given cultural theme can function as a genuine learning scaffold, as well as to guide practitioners in preventing cultural narratives from generating historical or technical misdirection.

2.3 Scaffolding Theory and Structure-Mapping Theory

The scaffolding theory proposed by Wood, Bruner, and Ross posits that scaffolds serve as temporary support mechanisms enabling learners to complete tasks beyond their unaided capacity; as learners build proficiency, these supports may be gradually withdrawn [14]. For abstract topics such as binary number conversion, symbolic systems familiar to learners through their cultural backgrounds can act as concrete, provisional learning scaffolds.

Structure-mapping theory spells out the prerequisites for analogies to foster deep learning. A useful analogy hinges on mapping relational structure between the source and target domains, rather than mere similarity of superficial features [15]. The correspondence between the trigram system of The I Ching and binary counting falls precisely into this relational category. Each additional line doubles the number of distinguishable states (the Two Forms generate the Four Images, and the Four Images generate the Eight Trigrams), matching perfectly the rule that each extra bit in binary doubles the span of representable values. Accordingly, this study frames the connection between The I Ching and binary numbers as a formal structural correspondence suitable for instructional analogy. Two boundary statements are embedded into the instructional design from its inception. First, the link constitutes a structural analogy rather than strict mathematical isomorphism. Second, historically speaking, The I Ching did not give rise to binary arithmetic. Leibniz encountered Fu Xi's arrangement of the sixty-four hexagrams only after he had formulated his binary arithmetic system. Available scholarship yields no evidence that ancient China possessed a binary numeration system. This design tightly constrains the cultural narrative to function solely as a learning scaffold and avoids any claims of historical cultural priority.

3. Instructional Design

3.1 Design Principles: Three Integration Criteria

To differentiate superficial presentation from deep integration, this paper puts forward three criteria for screening cultural themes. First, structural mappability: a cultural theme and target knowledge should support mapping at the relational- structure level. In this study, mapping is established through hierarchical doubling logic, instead of relying merely on surface resemblance. Second, instructional functionality: the cultural element must deliver tangible instructional value, such as reducing conceptual abstraction, supporting recall, or structuring practice activities, rather than acting merely as decorative flourishes at the opening or closing of a lesson. Third, boundary declarability: the limits of an analogy need to be clearly expressible and explicitly communicated to students in class, so that the learning scaffold does not spawn misconceptions, including false assertions of historical priority. Any cultural theme failing to satisfy even one criterion is only appropriate for superficial presentation.

3.2 Three-Track Instructional Design

This instructional intervention comprises two class hours of teaching content on number system conversion, organized around three parallel tracks.

3.2.1 Track 1: Cultural Narrative

Objective: Leverage cultural symbols to build intuitive understanding of binary representation and the hierarchical growth of representable numeric ranges.

Lesson opening: A short popular-science video introduces the derivation logic recorded in the classic text: "Taiji - Two Forms - Four Images - Eight Trigrams." Students are guided to observe the quantitative pattern: each additional layer doubles the number of combinations.

Structure mapping: The symbolic correspondence is established step by step: the yang line corresponds to 1 and the yin line to 0; the Four Images correspond to all four combinations of 2 bits; the Eight Trigrams correspond to all eight combinations of 3 bits. The traditional trigram mnemonics are adapted as an aid for memorizing the full enumeration of 3-bit states.

Cultural extension: The Tai (Peace) hexagram from the sixty-four hexagrams is selected,

together with the Forbidden City's central axis of "Palace of Heavenly Purity – Hall of Union – Palace of Earthly Tranquility," as enrichment material for appreciating systems thinking in traditional culture. It is explicitly stated that this portion is extension content and not part of the assessed core.

3.2.2 Track 2: Algorithmic Programming

The learning objective is to master the core principles and multiple implementation methods of number system conversion. Adopting guided inquiry, the instruction avoids directly presenting the formula 2^N. Instead, students enumerate all bit strings of 1, 2, 3, and 4 bits. Through the operation of "carrying over upon reaching two," they independently uncover the growth pattern of representable numeric ranges.

A dual-path instructional model is deployed. The general path covers positional-weight expansion (conversion from any base to decimal) and repeated division-by-base with remainder collection (conversion from decimal to any base). The shortcut path introduces the 8421 positional-weight mnemonic, the three-bit grouping technique for octal conversion, and the four-bit grouping technique for hexadecimal conversion. Conversion rules are condensed into concise mnemonics to reinforce retention. It must be clarified that such mnemonics act only as retrieval cues once students grasp the underlying principles. Exercises that require students to explain these principles are assigned alongside, to guard against rote memorization.

3.2.3 Track 3: Engineering Application

Objective: Establish the connection between number system conversion and real embedded engineering practice, so that students appreciate that binary and hexadecimal are the actual working language of hardware control.

Using an 8051 microcontroller driving eight LEDs as the engineering vehicle, instruction proceeds through three levels: bit–byte–system. **Level 1 (bit-level control)** begins with lighting a single LED, presenting both bitwise addressing (P0^0 = 0;) and bit-definition (sbit LED = P0^0; LED = 0;) notations to help students build the intuition that "one binary bit corresponds to the level of one LED." **Level 2 (byte-level control)** introduces whole-port assignment (P0 = 0xFE;), with on-the-spot decomposition of the hexadecimal literal: 0xFE corresponds to binary 1111 1110, lighting the rightmost LED. Students then quickly convert and assign 0x7F, 0xBF, 0xF7, and other literals to light LEDs at different positions, repeatedly practicing the "binary-to-hex, four-bit grouping" technique in a hardware context so that the conversion methods acquired in Track 2 find genuine application. **Level 3 (system-level application)** covers a running-light program, P0 = ~(0x01 << cnt), which uses the left-shift and bitwise-NOT operators to generate bit patterns dynamically; combined with software delay, this produces a continuously moving lit effect, demonstrating the dynamic relationship between bitwise operations and binary bit patterns.

After class, extension design tasks are provided—right-shifting running lights, patterned running lights, and a "bird returning to nest" effect—for students seeking further challenge. Through these cases, students are conveyed the message that number system conversion (especially the four-bit-grouping conversion between binary and hexadecimal) is a fundamental skill of embedded development, not merely an exam topic.

3.3 Tiered Programming Exercises

Two C programming tasks are designed to accommodate learners with different learning paces. The core task requires students to take a decimal integer no greater than 65535. Without using the %o or %x format specifiers, they manually implement number-system conversion logic and print the corresponding binary, octal and hexadecimal outputs. The extension task, named the Trigram Generator, asks learners to define a structure to store the hexagram name, numeric value, line pattern and mnemonic. They parse the three-line state mapped from a given number via bitwise operations and render the hexagram derivation process using a simple delay animation.

3.4 Classroom Implementation Strategies

This instructional design incorporates three classroom implementation strategies. First, a progressive question chain covers all knowledge points and reveals student misconceptions in real time. Second, error reuse converts typical student mistakes into bidirectional conversion exercises, guiding learners to resolve errors through reverse reasoning instead of receiving direct corrections from the instructor. Third, immediate assessment employs short in-class written quizzes to regulate lesson pacing.

For the boundary declaration segment, a prepared instructional script is delivered to students. Such

as "The I Ching shares structural similarities with binary numbers that can support our learning. Binary arithmetic originated in 17th-century Europe, and the I Ching is not the origin of binary." In this implementation, the boundary statement was presented after the structure-mapping instruction. Classroom observations (see Section 4.3) suggest that introducing this declaration before students encounter the analogy tends to leave only a weak impression.

3.5 Setting, Participants, and Data Sources

This study is a report of instructional design practice, employing a descriptive case study method to document the implementation status and observations of the first-round classroom delivery; it involves no control condition and makes no causal inference.

The setting was the compulsory C programming course in the second semester of the freshman year of the Electronic Information Engineering program at our institution (an application-oriented undergraduate university). Participants were 53 students in two parallel classes (Class A, n = 26; Class B, n = 27), taught by the same instructor; students were not grouped by programming background at enrollment. Both classes covered identical technical content, programming tasks, and assessment requirements, with this design course implemented as the organizing framework of the unified teaching content. The intervention spanned two class hours (50 minutes each), delivered according to the three-track design described in Section 3.2.

Data sources fall into three categories. The first consists of records of students' performance on tiered programming tasks, tabulated according to core versus extension tasks and individual versus collaborative completion. The second includes scores from the in-class quiz. This quiz contained fifteen items measuring bidirectional conversion between decimal, binary, octal and hexadecimal, and was developed by the course team based on the day's instructional content. The third source is the post-lesson open-ended feedback questionnaire. Fifty-three copies were distributed, yielding forty valid responses. It included an open prompt such as "What impressed you most in this lesson?", and the instructor coded the replies by theme (cultural narrative, programming task, or other).

4. Classroom Implementation and Observations

This instructional design was trialled in the compulsory C Programming course. It targeted first-year Electronic Information Engineering students at Xi'an Peihua University. One instructor taught two parallel classes with a combined enrollment of 53 students: 26 in Class A and 27 in Class B. Technical content, assigned tasks and assessments remained identical across both sections. This design served as the organising framework to structure the unified teaching materials.

What follows presents only descriptive observations collected from this first round of implementation. The goal is to document how the design operated in practice, rather than to supply comparative evidence for instructional effectiveness.

4.1 Student Programming Outcomes

```
===== Taiji gives birth to Two Modes, Two Modes to Four Symbols, Four Symbols to Eight Trigrams =====


Taiji: Chaos differentiates, producing two forces of Yin and Yang
Two Modes: Yang line --- represents 1; Yin line - - represents 0
Two Modes produce Four Symbols: 2 lines combined, 4 types; Four Symbols produce Eight Trigrams: 3 lines
combined, 8 types (2^3=8)
Eight Trigrams mnemonic: Qian three connected, Kun six broken; Zhen upturned bowl, Gen overturned bowl;
Li empty in middle, Kan full in middle; Dui broken on top, Xun broken at bottom.
================================================================================

------------ Generating Eight Trigrams in sequence ------------

- -
- -
- -
[Kun] Value:0 | Kun six broken, all three lines Yin

———
- -
- -
[Gen] Value:1 | Gen overturned bowl, one Yang above, two Yin below

- -
———
- -
[Kan] Value:2 | Kan full in middle, Yang in middle, Yin above and below
```

Figure 1. Implementation of the Trigram Generator

Table 1 presents the statistics on students' completion of the tiered programming tasks described in Section 3.3. The core-task completion rate was 84.6% in Class A and 85.2% in Class B, indicating that the design's implementation was stable across classes and did not depend on class-specific characteristics. The extension-task completion rate was 73.1% in Class A and 70.4% in Class B, suggesting that the cultural narrative appeal of the Trigram Generator motivated most students to take on the extension challenge voluntarily.

Figures 1 and 2 illustrate students' implementation of the extension task.

Table 1. Completion of Tiered Programming Tasks in the Two Parallel Classes

Class	Total	Core task (%)	Independent core	Extension task (%)	Independent extension
A	26	84.6%	22	73.1%	19
B	27	85.2%	23	70.4%	19
Total	53	84.9%	45	71.7%	38

```
=================== Bagua Symbol Query System ====================
Yang yao shown as -- (1), yin yao shown as - - (0)
------------------------------------------------------------------
[Bagua Menu]
  0 → Kun
  1 → Gen
  2 → Kan
  3 → Xun
  4 → Zhen
  5 → Li
  6 → Dui
  7 → Qian
------------------------------------------------------------------
Enter 0~7 to view the corresponding trigram, or enter q to quit

Please enter a number (0-7): 5

──
- -
──
[Li] value:5 | Li: empty in the middle (yin between two yang)
```

Figure 2. Implementation of the Hexagram Symbol Query System

4.2 Classroom Engagement and Feedback

The in-class interactions included several open-ended questions and directed follow-up probes, with an overall response coverage rate of approximately 80%. The lesson opening with the "Eight Trigrams" quickly sparked students' curiosity; when the teacher probed "Can the yang and yin lines be mapped to 1 and 0?", several students actively drew on the generative logic of "Two Forms generating the Four Images, Four Images generating the Eight Trigrams" for analogy. Typical response trajectories showed two patterns: a **consistently active** type, whose members kept volunteering answers even at difficult points, and a **silent-then-activated** type, whose members did not respond early on but were gradually drawn into participation by the active classroom atmosphere. The in-class quiz centered on bidirectional conversion between decimal and other number bases. It returned an overall accuracy of roughly 81%, of which the excellence rate stood at around 33%. In terms of stratified performance, high-scoring students' errors were mostly concentrated in choosing the wrong divisor K in "divide by K, take remainder," reflecting a still-mechanical understanding of positional weight; low- and mid-scoring students' errors were concentrated in the conversion procedures themselves. The post-class open feedback yielded 40 valid questionnaires. Of these, 27 students (approximately 68%) mentioned the "trigram narrative," and 18 students (approximately 45%) mentioned the "programming tasks (computer/binary)." The cross-distribution of the feedback was notable, indicating a positive interplay between cultural narrative and technical application.

4.3 Observations Related to the Boundary Declaration

The most instructive observations concern misconception intervention. During and immediately after the structure-mapping instruction, some students spontaneously developed the misconception that binary arithmetic historically originated from the *I Ching*. A scripted boundary declaration delivered after students had established the analogical cognition served as a reference point for correcting the misconception in subsequent class discussion. Two instructional design implications follow. First, the analogy itself is persuasive enough to lead students to infer a historical conclusion that does not exist; any culture-integrated computing instructional design must therefore build an explicit, pre-designed boundary declaration into the design itself, rather than rely on ad hoc post hoc supplementation. Second, timing is critical. This implementation found that delivering the boundary declaration after students have formed some cognition of the analogy works best, because only then do students genuinely care about the analogy's limitations.

5. Discussion

Contribution to culturally responsive computing: This case integrates values education with disciplinary teaching and demonstrates how a cultural theme can shed its decorative role and

become a genuine cognitive scaffold. The *I Ching* is not an ornament of the binary classroom but the vehicle through which students first organize their understanding of positional representation. The three screening criteria in Section 3.1 operationalize this idea and are not limited to this case; other cultural themes in the future (knotted cords for graph theory, the abacus for place value, traditional games for state machines) can likewise be evaluated against the three criteria of structural mappability, instructional functionality, and boundary declarability before instructional design begins.

Implications for scaffolding and analogy theory in computing education: This case demonstrates the applied value of structure-mapping theory in computing instruction. The instructional value of the *I Ching* derives precisely from the alignment of relational structures between the two domains; the misconception observed in Section 4.3 is a predictable consequence of analogy's strong pull effect, consistent with the theory's proposition that both surface and relational similarity drive analogical mapping.

Boundary statements as a formal design element: This paper proposes that in all indigenous-culture-integrated computing instruction, the boundary declaration should be a first-class design component—pre-planned, scripted, timed, and revisited. This practice addresses a common criticism of culture-integrated instruction: its susceptibility to essentialist narratives or historical distortion.

6. Conclusion

Building on a formal-level structural correspondence, this instructional design practice integrated the *I Ching* symbolic system into the binary number conversion classroom through a three-track teaching framework accompanied by tiered programming exercises. The value of this work lies not in this single lesson but in a transferable framework: three screening criteria for selecting cultural instructional materials, and the design principle that "boundary declarations are an inherent component of instructional design, not an afterthought." When a cultural theme and a technical concept genuinely possess relational-structural correspondence, the cultural narrative has the opportunity to serve as cognitive scaffolding rather than classroom decoration. Future work will replicate the implementation across multiple classes and institutional contexts, introducing delayed post-tests and control conditions to examine the framework's transferability and long-term effects.

Acknowledgments

This work was supported by the 2025 Curriculum-based Ideological and Political Education Teaching Reform Research Project of Xi'an Peihua University (Grant No. PHKCSZ202501).

References

[1] Robins A, Rountree J, Rountree N. Learning and teaching programming: A review and discussion. Computer Science Education, 2003, 13(2): 137–172.

[2] Zhang Y, Chen J. Learning obstacles in introductory college C programming and pedagogical countermeasures (in Chinese). Journal of Computers in Education , 2022(5): 142–146.

[3] Herman G L, Zilles C, Loui M C. How do students misunderstand number representations? Computer Science Education, 2011, 21(3): 289–312.

[4] Lachney M, Bennett A G, Eglash R, et al. Teaching in an open village: A case study on culturally responsive computing in compulsory education. Computer Science Education, 2021, 31(4): 462–488.

[5] Wang Y, Ma Z. A review of research on culturally responsive computational thinking instruction (in Chinese). China Educational Technology , 2023(8): 118–126.

[6] Putra Z H, Ramiati, Zufriady, et al. Development of computational thinking tasks based on Riau Malay culture: A study of fifth-grade public school students in Pekanbaru, Indonesia. Education 3-13, 2024, 52(8): 1387–1397.

[7] Wood D, Bruner J S, Ross G. The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry, 1976, 17(2): 89–100.

[8] Ryan J A. Leibniz' binary system and Shao Yong's "Yijing". Philosophy East and West, 1996, 46(1): 59–90.

[9] Li N. Cognitive difficulties in number system conversion instruction and analogy-based teaching strategies (in Chinese). Journal of Higher Education Science , 2022(3): 88–94.

[10] Muhiddin M, Saleh S F, Akib I. Culturally contextualized ebook based on West Papua's culture to support elementary students'

mathematical connection skills. MIMBAR PGSD Undiksha, 2025, 13(1): 96–104.
[11] Brown D E. Misconceptions in analogical instruction: When structural similarity leads to historical misattribution. Journal of Science Education, 2023, 24(2): 189–204.
[12] Wing J M. Computational thinking. Communications of the ACM, 2006, 49(3): 33–35.
[13] Bell T, Hickman H. Why teach about binary numbers? //Creative Mathematical Sciences Communication: 7th International Conference, CMSC 2024. LNCS 15229. Springer, 2024: 130–135.
[14] Liu F. Pathways and evaluation framework for integrating Chinese excellent traditional culture into university computer courses (in Chinese). Computer Era，2024(2): 112–115.
[15] Gentner D. Structure-mapping: A theoretical framework for analogy. Cognitive Science, 1983, 7(2): 155–170.