The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
Paper ID #42185

Board 293: How to Teach Debugging? The Next Million-Dollar Question in Microelectronics Education

Haniye Mehraban, Oklahoma State University

Haniye Mehraban obtained her Master of Science degree in Electrical Engineering from K.N. Toosi University of Technology, Tehran, Iran, in 2017. Currently, she is a Ph.D. student in Electrical Engineering at Oklahoma State University, Stillwater, OK, USA. Her research interests are primarily focused on Analog Integrated Circuit Design.

Dr. John Hu, Oklahoma State University

John Hu received his B.S. in Electronics and Information Engineering from Beihang University, Beijing, China, in 2006 and his M.S. and Ph.D. in electrical and computer engineering from the Ohio State University, Columbus, OH, in 2007 and 2010, respectively. He worked as an analog IC designer at Texas Instruments, Dallas, between 2011 and 2012. He was a Member of Technical Staff, IC Design at Maxim Integrated, San Diego, CA, between 2012 and 2016, and a Staff Engineer at Qualcomm, Tempe, AZ, between 2016 and 2019. In 2019, he joined the School of Electrical and Computer Engineering at Oklahoma State University, where he is currently an assistant professor and Jack H. Graham Endowed Fellow of Engineering. His research interests include power management IC design, hardware security, and energy-efficient computing.

How to Teach Debugging? The Next Million-Dollar Question in Microelectronics Education

Introduction

The Chips and Science Act [1] has made semiconductor workforce development a top priority for US universities. Among the many skills undergraduates need to enter the semiconductor industry, debugging skills are often overlooked but essential for new product development [2]. As the transistor count and complexity of today's chips grow, thanks to Moore's Law [3], fewer new chips can work perfectly for the first time. Therefore, more effort is spent on debugging, a specific form of troubleshooting that identifies and fixes any discrepancies between the expected and measured chip behavior. However, such an important skill is rarely taught in college, not to mention how to teach it effectively. This paper seeks to motivate the electrical and computer engineering (ECE) community for this challenge through economic, cognitive, and psychological lenses. The economic data and impact shows that it is truly the next million-dollar question in microelectronics education.

Economics of Semiconductor Debugging

The global semiconductor industry is on track to be a trillion-dollar industry by 2030 [4]. Debugging is a special case of troubleshooting essential for new integrated circuits (chip) development [5, 6]. With the increasing transistor count and complexity of modern chips, thanks to Moore's Law [3], fewer and fewer chips can work perfectly for the first time. As a result, a lot of engineering effort goes into finding and fixing bugs [7]. The need for post-fabrication silicon debugging has given rise to a new profession known as validation[1] engineers [8]. As shown in Figure 1, between 2007 and 2020, the demand for Integrated Circuit/Application-Specific Integrated Circuit (IC/ASIC) design engineers demonstrated a slight increase of around 3 percent, while the need for IC/ASIC verification engineers increased by 6.8 percent [9]. According to estimates, debugging takes between 35 and 50 percent of the time on a typical semiconductor project [7]. Figure 2 shows the percentage of time IC/ASIC verification engineers spent for various tasks in 2020. The majority, constituting 41%, is allocated to debugging. Due to the uncertainty in the debug time and the time requires to fix the bugs, the debugging process has gained the nickname of the **Schedule Killer** [10], highlighting its impact on the project schedule and the company's bottom line.

[1] In this paper, we interchangeably refer to verification engineers and validation engineers.

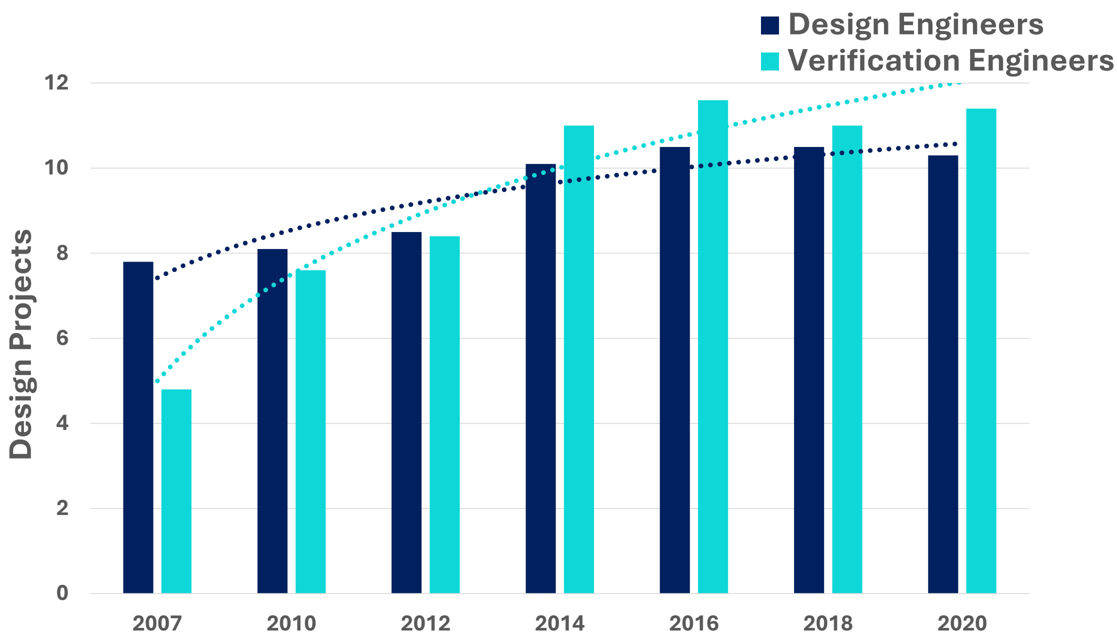

Figure 1: Mean Peak Number of Engineers on ASIC/IC Projects. Source: Wilson Research Group and Mentor, a Siemens Company [9].

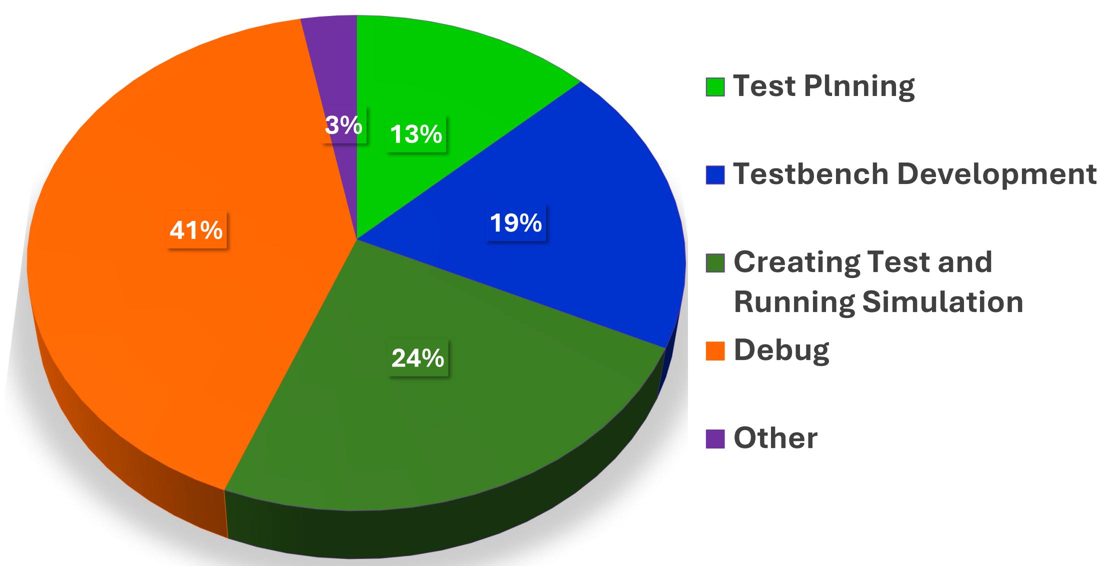

Figure 2: Where ASIC/IC Verification Engineers Spend Their Time in 2020. Debugging takes up the most of the time. Source: Wilson Research Group and Mentor, a Siemens Company [9].

An Important Topic that is Rarely Taught

All engineers, regardless of whether they specialize in software- or hardware-based systems, need to be skillful in troubleshooting [11]. Debugging is a skill that can be taught even though it may seem intuitive. In order to achieve this, debugging must be emphasized through efficient teaching methods [12]. As early as the 1970s, the computer science and software engineering domains began to acknowledge and explore debugging strategies [13, 14]. Numerous studies on software debug education have been reported [6, 15, 16, 17]. However, the same cannot be said for semiconductor or chip debugging. The closest example of debug education on circuits is in the context of physics education and automatic debugging tools [18]. Most electrical and computer engineering (ECE) educators do not teach debugging in their classes. Students lack these skills as a result [19]. Therefore, the need for dedicated semiconductor debugging education is clear, and addressing it has a significant economic impact.

Cognitive Research on Debugging

To develop an effective debug education program, we first study various cognitive models for troubleshooting, which will provide us with the theoretical basis for improving students' debugging skills. As defined in [20], Cognitive Task Analysis is "a family of methods used for studying and describing reasoning and knowledge". This section offers an overview of different cognitive task analyses related to the process of troubleshooting.

Johnson et al. [21] proposed a cognitive troubleshooting model with four sub-processes: problem space construction, problem space reduction, hypothesis generation/testing, and solution generation/verification. A three-phase model was later suggested by Axton et al. [22]. Schaafstal et al. [23] characterized debugging as four sub-tasks: formulate problem description, generate causes, test, and repair and evaluate. The term "formula problem description" describes the first phase of the troubleshooting process, in which the troubleshooter finds out both what the system is doing wrong and what it is doing right. The process of producing causal hypotheses is known as "generate causes". Experts typically recognize common symptoms, but troubleshooters typically employ reasoning skills, functional thinking, and external documentation when they meet an issue for the first time. Testing is the process of measuring, testing or verifying if a suggested cause is, in fact, the real issue that needs to be fixed. Finally, to restore the system to its typical operating state, repair and evaluate include generating, enacting, and verifying solutions [24, 23]. However, none of these early researches included experiences in their cognitive model.

A troubleshooting learning architecture was proposed by Jonassen et al. [25] in order to adequately capture the role of experience in the cognitive model which comprises three essential elements: a multi-layered model of the system that includes topographic, function, strategic, and procedural representations; a simulator for testing hypotheses; and a case library that maintains relevant past experiences as guidance for the learner. This learning architecture has been used by the most recent study in computer science education to uncover research gaps in programming debugging tools and intervention creation [26]. In Physics Education, this cognitive framework has also been modified to establish a correlation between circuit debugging and model-based reasoning [24] and to classify most existing troubleshooting education into a paradigm of the cognitive apprentice [27, 28]. Some of the most recent pedagogical innovations, such collaborative pair debugging [24, 29], which makes use of socially mediated metacognition [30], were inspired by the cognitive apprentice classification.

Affective Research on Debugging

In addition to cognitive models, we also want to address the affective components of debugging, such as emotions. Debugging inspires a wide range of feelings. When bugs are encountered, some students may experience frustration, fear, and anxiety that cause them to become disengaged and avoid the subject [31]. They might also fail the course because they don't have the patience to "fix" their issues [12]. Fields et al. [32, 33] observed feelings of comfort and competency when high school students designed buggy projects (debugging by design, or DbD) for their peers to debug. Other positive feelings including mischievousness, fun, empathy, and sensitivity were also recognized. Despite the fact that the DbD intervention was developed within the concept of productive failure, the emotional shift can likely be attributed to putting students in charge of bug creation and subsequent debugging [32].

Mindset is one of the other non-cognitive elements that may be connected to debugging. The ability of students to work systematically is correlated with their debugging skills, according to Bottcher et al. [11]. This suggests that before debugging can be properly taught, educators should focus on developing a set of non-technical foundational skills. O'Dell et al. [34] highlighted the importance of self-theory [35] for programming education. In university CS education, interventions to promote a growth mindset have been extensively invesigated, but the results of such inventions are mixed. Cutts et al. [36] reported positive results in academic performance, whereas Simmons et al. [37] saw no significant difference. Other studies saw improvements in student interest [38] and effort [39] but not performance [38, 39]. Quille et al. [40] reaffirmed Cutts' results but observed that mindset interventions might affect different groups (e.g., age and performance level) differently. Scott et al. [31] found that a domain-specific aptitude differs from a general mindset toward intelligence, and the former predicts the students' performance in the subject better. Rangel et al. [39] speculated that a generic growth mindset intervention needs domain support to change academic performance.

Recommendations

The takeaway message from the literature review so far is that a genuinely effective debugging education intervention must be **holistic** and **domain specific**. Holistic means that the intervention should address both cognitive and affective components. Domain specificity means that any growth mindset message should be contextually situated within the subject matter materials. Hence, our next step is to develop a pilot debug training program within the laboratory sessions of an introductory microelectronic course (Sedra & Smith [41]) at our institution.

Conclusion

The semiconductor industry worldwide is poised to experience significant growth and become a trillion-dollar industry by 2030. With debugging becoming an increasingly important piece for new chip development and the lack of current methods to cultivate this skill, how to teach debugging is truly the next "million-dollar question." Future approaches must address both cognitive and affective components of debugging and embed within existing microelectronics curriculum.

Acknowledgement

This work was supported by the National Science Foundation Award EES-2321255.

References

[1] T. Ryan, "H.R.4346 - 117th Congress (2021-2022): Chips and Science Act," Aug. 2022. [Online]. Available: https://www.congress.gov/bill/117th-congress/house-bill/4346

[2] R. Sarmento, F. Pereira, and C. Felgueiras, "Hardware Test subjects in academic education," Jun. 2022, pp. 1–5, iSSN: 2766-2616.

[3] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[4] O. Burkacky, J. Dragon, and N. Lehmann. The semiconductor decade: A trillion-dollar industry. [Online]. Available: https://www.mckinsey.com/industries/semiconductors/our-insights/the-semiconductor-decade-a-trillion-dollar-industry#

[5] R. Sarmento, F. Pereira, and C. Felgueiras, "Hardware test subjects in academic education," in *2022 Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (XV Technologies Applied to Electronics Teaching Conference)*, 2022, pp. 1–5.

[6] I. R. Katz and J. R. Anderson, "Debugging: An Analysis of Bug-Location Strategies," *Human–Computer Interaction*, vol. 3, no. 4, pp. 351–399, Dec. 1987. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1207/s15327051hci0304_2

[7] Debug: The removal of bugs from design. [Online]. Available: https://semiengineering.com/knowledge_centers/eda-design/verification/debug/

[8] A. Mutschler. (2018, Dec.) Debug tops verification tasks. [Online]. Available: https://semiengineering.com/debug-tops-verification-tasks/

[9] The 2020 Wilson Research Group Functional Verification Study. [Online]. Available: https://blogs.sw.siemens.com/verificationhorizons/2021/01/06/part-8-the-2020-wilson-research-group-functional-verification-study/

[10] B. Bailey. (2021, Jun.) Debug: The schedule killer. [Online]. Available: https://semiengineering.com/debug-the-schedule-killer/

[11] A. Böttcher, V. Thurner, K. Schlierkamp, and D. Zehetmeier, "Debugging students' debugging process," in *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–7.

[12] P. Nagvajara and B. Taskin, "Design-for-debug: A vital aspect in education," in *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*, 2007, pp. 65–66.

[13] J. D. Gould and P. Drongowski, "An Exploratory Study of Computer Program Debugging," *Human Factors*, vol. 16, no. 3, pp. 258–277, 1974. [Online]. Available: https://doi.org/10.1177/001872087401600308

[14] J. D. Gould, “Some psychological evidence on how people debug computer programs,” *International Journal of Man-Machine Studies*, vol. 7, no. 2, pp. 151–182, 1975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020737375800058

[15] A. Marzieh, D. Elliman, and C. Higgins, “An analysis of patterns of debugging among novice computer science students,” vol. 37, 09 2005, pp. 84–88.

[16] M. Ducassé and A.-M. Emde, “A review of automated debugging systems: knowledge, strategies and techniques,” *Proceedings. [1989] 11th International Conference on Software Engineering*, pp. 162–171, 1988. [Online]. Available: https://api.semanticscholar.org/CorpusID:10989956

[17] R. Chmiel and M. C. Loui, “Debugging: from novice to expert,” in *Technical Symposium on Computer Science Education*, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:8671590

[18] D. Drew, J. L. Newcomb, W. McGrath, F. Maksimovic, D. Mellis, and B. Hartmann, “The toastboard: Ubiquitous instrumentation and automated checking of breadboarded circuits,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 677–686. [Online]. Available: https://doi.org/10.1145/2984511.2984566

[19] H. Duwe, D. T. Rover, P. H. Jones, N. D. Fila, and M. Mina, “Defining and supporting a debugging mindset in computer engineering courses,” in *2022 IEEE Frontiers in Education Conference (FIE)*, 2022, pp. 1–9.

[20] B. Crandall, G. A. Klein, and R. R. Hoffman, *Working Minds: A Practitioner’s Guide to Cognitive Task Analysis*. The MIT Press, 07 2006.

[21] J. J. J. Scott D. Johnson, Jeff W. Flesher and A. Ferej, “Enhancing Electrical Troubleshooting Skills in a Computer-Coached Practice Environment,” *Interactive Learning Environments*, vol. 3, no. 3, 1993, publisher: Routledge pages = 199–214,.

[22] T. Axton, D. Doverspike, S. Park, and G. Barrett, “A model of the information-processing and cognitive ability requirements for mechanical troubleshooting,” *International Journal of Cognitive Ergonomics*, vol. 1, no. 3, pp. 245–266, 1997.

[23] A. Schaafstal *et al.*, “Cognitive task analysis and innovation of training: The case of structured troubleshooting,” *Human Factors*, vol. 42, no. 1, p. 75, 2000, spring. [Online]. Available: link.gale.com/apps/doc/A63993986/AONE?u=anon~5588433b&sid=googleScholar&xid=1f45aa77

[24] D. R. Dounas-Frazer, K. L. Van De Bogart, M. R. Stetzer, and H. J. Lewandowski, “Investigating the role of model-based reasoning while troubleshooting an electric circuit,” *Phys. Rev. Phys. Educ. Res.*, vol. 12, p. 010137, Jun 2016. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.12.010137

[25] D. H. Jonassen and W. Hung, “Learning to Troubleshoot: A New Theory-Based Design Architecture,” *Educational Psychology Review*, vol. 18, no. 1, pp. 77–114, Mar. 2006.

[26] C. Li, E. Chan, P. Denny, A. Luxton-Reilly, and E. Tempero, “Towards a framework for teaching debugging,” in *Proceedings of the Twenty-First Australasian Computing Education Conference*, ser. ACE ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 79–86. [Online]. Available: https://doi.org/10.1145/3286960.3286970

[27] J. S. Brown, A. Collins, and P. Duguid, “Situated Cognition and the Culture of Learning,” *Educational Researcher*, vol. 18, no. 1, pp. 32–42, 1989. [Online]. Available: https://doi.org/10.3102/0013189X018001032

[28] D. R. Dounas-Frazer and H. J. Lewandowski, “Electronics lab instructors’ approaches to troubleshooting instruction,” *Phys. Rev. Phys. Educ. Res.*, vol. 13, p. 010102, Jan 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.13.010102

[29] K. L. Van De Bogart, D. R. Dounas-Frazer, H. J. Lewandowski, and M. R. Stetzer, “Investigating the role of socially mediated metacognition during collaborative troubleshooting of electric circuits,” *Phys. Rev. Phys.*

Educ. Res., vol. 13, p. 020116, Sep 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.13.020116

[30] Van De Bogart, Kevin L. and Dounas-Frazer, Dimitri R. and Lewandowski, H. J. and Stetzer, MacKenzie R., "The role of metacognition in troubleshooting: An example from electronics," in *2015 Physics Education Research Conference Proceedings*, ser. 2015 PERC. American Association of Physics Teachers. [Online]. Available: http://dx.doi.org/10.1119/perc.2015.pr.080

[31] M. J. Scott and G. Ghinea, "On the domain-specificity of mindsets: The relationship between aptitude beliefs and programming practice," *IEEE Transactions on Education*, vol. 57, no. 3, pp. 169–174, 2014.

[32] D. A. Fields, Y. B. Kafai, L. Morales-Navarro, and J. T. Walker, "Debugging by design: A constructionist approach to high school students' crafting and coding of electronic textiles as failure artefacts," *British Journal of Educational Technology*, vol. 52, no. 3, pp. 1078–1092, 2021. [Online]. Available: https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.13079

[33] D. A. Fields and Y. B. Kafai, "Debugging by design: Students' reflections on designing buggy e-textile projects," *Proceedings of Constructionism 2020*. [Online]. Available: https://par.nsf.gov/biblio/10162899

[34] D. H. O'Dell, "The debugging mindset: Understanding the psychology of learning strategies leads to effective problem-solving skills." *Queue*, vol. 15, no. 1, p. 71–90, feb 2017.

[35] C. S. Dweck, *Mindset: The new psychology of success*. New York, NY, US: Random House, 2006.

[36] Q. Cutts, E. Cutts, S. Draper, P. O'Donnell, and P. Saffrey, "Manipulating mindset to positively influence introductory programming performance," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 431–435. [Online]. Available: https://doi.org/10.1145/1734263.1734409

[37] B. Simon, B. Hanks, L. Murphy, S. Fitzgerald, R. McCauley, L. Thomas, and C. Zander, "Saying isn't necessarily believing: influencing self-theories in computing," in *Proceedings of the Fourth International Workshop on Computing Education Research*, ser. ICER '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 173–184. [Online]. Available: https://doi.org/10.1145/1404520.1404537

[38] J. L. Burnette, C. L. Hoyt, V. M. Russell, B. Lawson, C. S. Dweck, and E. Finkel, "A growth mind-set intervention improves interest but not academic performance in the field of computer science," *Social Psychological and Personality Science*, vol. 11, no. 1. [Online]. Available: https://par.nsf.gov/biblio/10091354

[39] J. G. C. Rangel, M. King, and K. Muldner, "An incremental mindset intervention increases effort during programming activities but not performance," *ACM Trans. Comput. Educ.*, vol. 20, no. 2, feb 2020. [Online]. Available: https://doi.org/10.1145/3377427

[40] K. Quille and S. Bergin, "Promoting a growth mindset in cs1: Does one size fit all? a pilot study," in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 12–18. [Online]. Available: https://doi.org/10.1145/3341525.3387361

[41] A. Sedra, K. Smith, T. C. Carusone, and V. Gaudet, *Microelectronic circuits*, 8th ed. Oxford University Press, 2020.