Hardware Systems within an Information Technology Curriculum

Michael G. Bailey, C. Richard G. Helps, Barry M. Lunt
Brigham Young University

Abstract

Information Technology (IT), as a discipline, focuses on the integration of various computer technologies to create working systems to meet users' needs. Appropriately, much emphasis is placed upon core topics such as software, web systems, networks, databases and human computer interaction. Hardware systems integration is not as strongly emphasized as these core topics, but a sound understanding of hardware will significantly enhance the expertise of IT professionals as system integrators.

At BYU, hardware systems content has been incorporated into the IT curriculum. The curriculum is time constrained and hardware topics have to be balanced against other topics competing for limited credit hours. Realizing that our students are not striving to become hardware designers, much of the hardware content is taught at the conceptual level, with more depth being applied to the particular areas required for effective systems integration.

This paper presents the hardware systems content that has emerged over several years of refining it within an IT curriculum. Areas emphasized include computer systems hardware, serial and parallel busses, physical layer communications protocols, frequency response, and broadband technologies. We discuss how this content can be presented effectively using just a few credit hours of the IT program.

Introduction

The emerging discipline of Information Technology (IT) has many origins, depending on the particular educational institution hosting the program[1]. These programs have emerged from Computer Science, Information Systems, Engineering Technology and Telecommunications departments, among others. As would be expected from such diverse backgrounds, there is considerable variability in the curricula vouchsafed at these institutions. However, there is a common body of knowledge that ties IT programs together – what has come to be called in the community the Pillars of Information Technology. These pillars are: software, web systems, networks, databases and human-computer interaction.

As has been previously asserted[2], the unique approach and emphasis that IT gives to these topics is ideal for the training of networking and other computing-systems engineers – that is people who stitch digital systems together. This type of professional should have depth of knowledge in the areas conducive to digital systems integration, in addition to broad skills that will enable him or her to troubleshoot problems that arise, advocate for computer users needs, interact with other technical professionals and otherwise manage digital systems.

Historically, before the advent of formal IT education, other technical professionals were retrained to provide these services. In this self-education process, the particular background needed for a particular task was acquired on an ad-hoc basis. In a rather notorious example, in the 1980's an astrophysicist was working as a network engineer when he stumbled across a discrepancy in some billing records[3]. He then trained himself to troubleshoot this issue, and eventually to track the hackers that were at the root of the problem. Anecdotal examples of the breadth required of this type of professional abound in IT lore, as in an account given in Computerworld,[4] in which an IT troubleshooter assists a user with a monitor whose display was unstable. The troubleshooter discovers that a computer monitor's poor performance is due to electromagnetic fields from a high-voltage power main directly behind the monitor. The embarrassed user was an electrical engineer, who was aware of the location of the power main.

These examples, among many other anecdotes, show the large scope required of IT professionals. In addition, the need for a broad-based technical education is not likely to diminish, as the internet expands to include the embedded systems and devices that surround us. These systems usually have little resemblance to the desktop PCs and servers that IT emphasizes but are becoming part of the IT mainstream as more and more systems converge.

An uninformed observer may conclude from the forgoing description of the emerging IT movement, that hardware is necessarily an important part of all such programs' curriculum. In discussions with our peer institutions across the country, we have discovered this not to be the case. A recent survey, conducted via the ACM-SIGITE (Special Interest Group on Information Technology Education) list server illustrates this. This survey asked each institution to self assess their coverage of the following 8 topics: Networking, AC and DC fundamentals, Power (consumption, heat, etc.), Modulation techniques, Signal Propagation, Digital logic circuits, Hardware devices and drivers, and Computer architecture. In the survey, a scale from 1 to 5 was used, with 1 indicating no coverage, and 5 indicating through treatment of the topic.

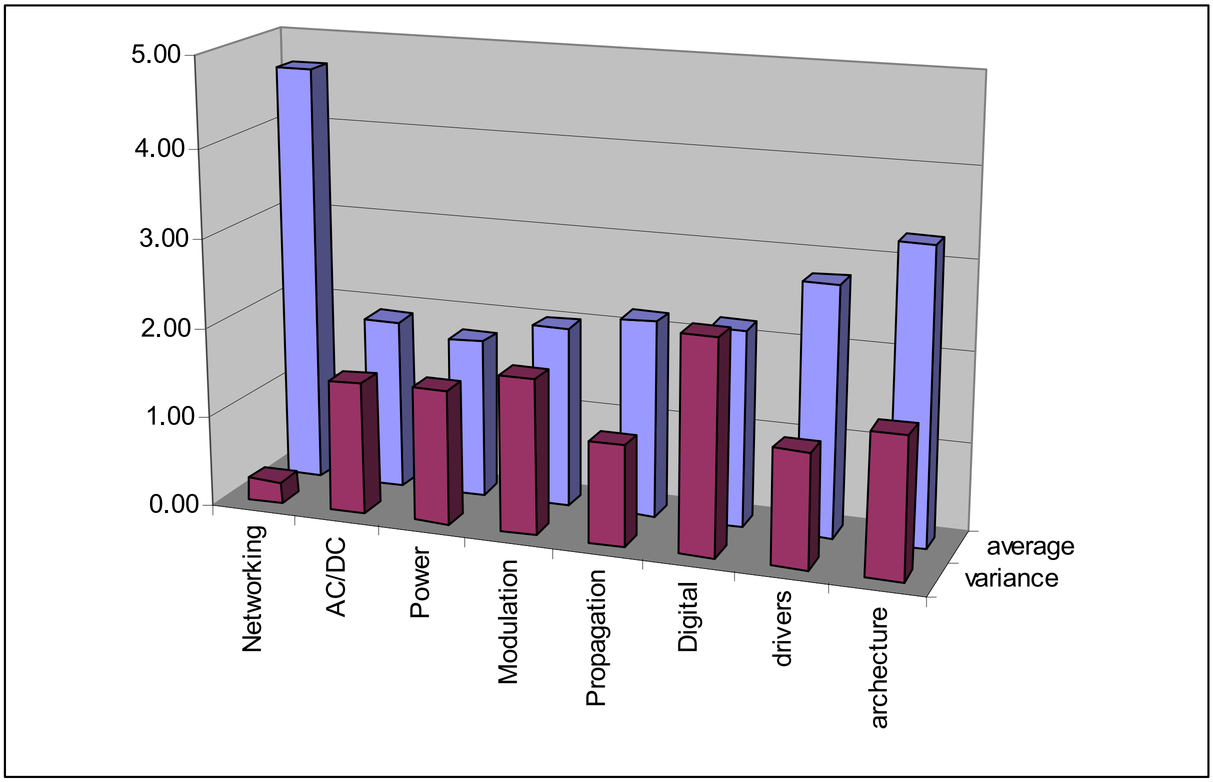

Figure 1 - IT institutions' hardware curriculum content

In order to better compare BYU's program to like programs, only the 11 responses where networking had a high priority (4 or 5) were included in the results. Note that BYU was not one of the polled institutions. With the exception of networking, the results show that there is little unanimity concerning hardware curriculum among IT educators. For each of the hardware topics listed, at least one program indicated that it had no coverage at all. In fact, a majority of institutions indicated no coverage for AC/DC electronics and for power concepts. Likewise, each hardware topic had strong support (4 or 5) from at least one institution. In summary, the IT community is far from unified regarding the need for hardware content in the curriculum.

In view of the fact that indeed, many of the IT professionals being trained today will be expected to function as digital systems engineers, educators are not serving them well if we neglect to adequately train them in computer system hardware. In fact, even the Computer Science discipline, which is generally thought of as being less systems-oriented than IT, expects that its students will get hardware training. For example, CC2001 specifies a minimum of 36 hours core coverage time of computer architecture, including six hours of digital logic and digital systems[5]. It should be unthinkable that IT, with its systems orientation, should do less.

There is, however, a problem with this point of view: There is so much that students should learn, and so little time in which they can be taught. Various educators can argue passionately that many different topics are essential to the IT practitioner, depending on the educator's perspective[6,7]. For example, many educators, including the authors, could argue that digital security should have a higher emphasis in IT education. So, how can we satisfy these divergent needs and desires within a limited curriculum?

This question is much larger than could be addressed in a single paper. However, this paper does show how the IT faculty at one institution, Brigham Young University, are providing a hardware background that will enable its graduates to become systems-capable IT professionals, while remaining within tight credit-hour constraints.

Goals of curriculum

When designing the hardware portion of the IT curriculum, the faculty soon realized that the hardware goals for an IT graduate are different than those for an electronics graduate. That is, there is no need for IT professionals to know how to design hardware. When design is required, the IT practitioner can contract with hardware design specialists. However, the IT professional will need sufficient knowledge to communicate hardware requirements to such specialists, and to troubleshoot the hardware when it malfunctions.

In addition, for IT professionals to achieve maximum performance from a system they are designing, they need a thorough grasp of hardware capabilities and constraints. For example, when setting up a wireless network, the IT worker should have some knowledge of electromagnetic propagation in order to properly place an access point's antenna. The IT professional should know how cable parameters and lengths can affect digital signals in systems and networks.

Finally, as a digital systems professional, the IT worker must be able to make hardware and software work well together. That is, the IT worker must be able to configure, and possibly create driver software for hardware.

Thus, the goals of the hardware portion of the IT curricula developed are to enable graduates to:

1. Troubleshoot digital system problems that cross the hardware/software boundary
2. Specify digital hardware requirements
3. Design systems, taking into account hardware capabilities
4. Configure digital system hardware
5. Facilitate hardware / software interoperability

The core areas of knowledge that will enable IT professionals to achieve these capabilities are:

1. Electrical principles (i.e. power, resistance, capacitance, etc.)
2. Signal propagation (in cables and the atmosphere)
3. Digital system architecture and building blocks
4. Devices and drivers

The curriculum that has been developed at BYU to address these goals is described below.

Implementation of curriculum

The hardware foundation portion of the IT curriculum is primarily contained in two courses – Digital Electronic Foundations and Computer Systems. Another course, Computer System Organization, presents a higher, systems view of hardware, as well as enabling students to create software to interface with hardware subsystems. Finally, the Digital Communications class gives students a good foundation in physical-layer communications concepts. Additional elective classes, such as Broadband Communications and Embedded Computer Systems, are available to students who want to gain more depth in specific hardware systems disciplines.

Considerable effort has been taken to review each topic taught, and judge its utility to IT practitioners, as well as *how* the knowledge might be applied. Since the IT department is new, several of the faculty have recent experience as digital systems engineers that has been put to use in this review. Note also that this is an evolving curriculum in its third year, which is just reaching a point that might be called stable.

A description of the curriculum in the core hardware courses is now presented. The first course in the curriculum, Digital Electronic Foundations, is divided in two completely separable halves, the first part on analog electronics, and the second on digital circuits. This separation achieves three things. First, as at most institutions with IT programs, many of our students transfer into our program from other institutions or programs, and they have had either analog or digital electronics before beginning study with us. By splitting the course in half, the students are able to take only the portion in which they are deficient. The second thing that is achieved by this bifurcated course, is that the analog portion is deemphasized, while still being sufficient preparation for the other courses in the curriculum. In effect the course teaches in ½ semester what is normally taught in 1 to 2 semesters. Admittedly, some material that is normally taught in electronics is not included, but the material omitted is of little use from a systems perspective. The third benefit achieved by combining digital material with analog in this course is that the students complete the course doing something enjoyable. Even the most jaded engineer must admit that digital logic is much less stressful to understand than analog. This enjoyable second half of the semester helps keep students enthusiastic about IT.

Below are listed the topics covered in the analog portion of this class. The particular aspects of digital systems to which each pertains is in brackets. In all cases, laboratory experiments using real hardware are used to reinforce the concepts.

- DC circuits and Ohm's Law [This gives an understanding of circuits, and how current flows within them. If they aren't taught, students don't know that no data flows in an incomplete circuit!]

- AC circuits [This includes frequency and phase, and what affects them. These are important concepts for understanding electrical signals on a cable or a bus.]

- Transients [Initial conditions when a system is turned on or off can wreak havoc if transients are not handled properly. Also, since voltages don't change instantaneously, transients control the data rates in our digital systems.]

- Inductors and Capacitors [Inductance is an important noise source in digital cables and systems. Capacitance limits digital signal response time, and thus data rates. Capacitors are useful as noise filters and in the power supplies used in computers.]

- Power [This is useful when considering uninterruptible power supplies, heat dissipation, and similar IT concepts.]

- Transformers [These are commonly used in power supplies, CRTs, and similar devices. Since these are some of the devices that most frequently have faults, it is good for IT practitioners to know these components.]

- Frequency Response and Resonance [Frequency response is a vital concept for understanding digital data transmission, particularly at various data rates. Resonance is used in various data transmitters and receivers.]

The second half of the Digital Electronic Foundations course, along with the Computer Systems course, is essentially similar to what CC2001 requires in the Computer Science curriculum. That is, the curriculum provides "understanding and appreciation of a computer system's functional components, their characteristics, their performance and their interactions."[5] As with CC2001, the topics include the fundamental digital building blocks (logic gates, flip-flops, counters, multiplexers, decoders, registers, and PLAs), Boolean logic and minimization, physical consideration (gate delays and fan-out), binary number representations and arithmetic, and CPU organization. As recognized by the Computer Science curriculum committee, this is excellent background material for computer professionals, including IT practitioners. In the IT curriculum, aspects related to digital communications and data busses and networks are emphasized (i.e. tri-state registers for use with busses, shift registers for serial networks).

The laboratory experiments associated with these courses include projects with discrete components as well as programmable logic devices. Dealing with such components gives excellent troubleshooting practice to the students. There is little to rival finding a broken wire in a project as a real world debugging experience, since such faults are common in industry too. Since most modern digital peripherals are implemented using some form of PLA, it is valuable for students to have experience with these components also. In industry, IT professionals who deal with any custom hardware or interfaces deal with PLAs, even if it be just specifying performance or troubleshooting non-performance issues.

The Digital Communications class starts with an understanding of the electromagnetic spectrum, and how it is allocated for multiple uses; an understanding of resonance and tuned circuits is essential to understand how this can be done. Another topic that depends on the prerequisite hardware courses is the concept of line drivers and power dissipation; the relationship between line (parasitic) capacitance, data rates, and power consumption is only comprehensible from a background of understanding capacitance and reactance. And finally, learning about serial/parallel and parallel/serial converters (serdes) is greatly aided by an understanding of flip-flops and shift registers.

The Computer System Organization course also builds on the basic hardware understanding developed in the core hardware classes. This course emphasizes understanding the various subsystems that comprise modern computers, and how they interoperate. In order to illustrate the concepts taught, students use assembly language to write firmware and drivers for the various components. This clearly teaches the relationship between software and real IO devices. There are other benefits as well: by using assembly language in the instruction process, this course teaches the importance of structured programming and appreciation of high-level languages.

Progress of Curriculum

Over the 3 years since the IT program's inception, the hardware content of the curriculum has been evolving. For example, one year the AC/DC content was revamped to use an integrated approach, where AC was taught from the beginning and DC was taught as a special, degenerate case of AC. The textbook publisher claimed that this was an especially time-efficient approach, which was of great interest to us given our constraints. However, careful analysis of identical questions on consecutive years' test results showed that student performance actually declined when using the integrated approach, all other factors (other than the actual student group) being equal. We returned to the more conventional approach of AC/DC instruction.

There have been aspects of the curriculum that have changed for the better. For example, it was decided that additional instruction on frequency effects, including elementary filters and Bode plots, would be useful for understanding digital communications concepts such as modulation and propagation. This material was added to the introductory electronics course. Unfortunately this course's content was already very ambitious, so something had to be removed to make room for the additions. In view of the facts that the goal of the course is to create IT professionals rather than electronic designers, the faculty eliminated more advanced circuit analysis techniques such as mesh and node analysis (which are rarely used in the age of computer simulations, anyway). Such tradeoffs are numerous, with the faculty always choosing to emphasize concepts important on a systems-design, rather than on a component-design level.

The IT program at BYU is three years old, so at this time we have no survey information that would indicate how our graduates view the utility of the hardware portion of the curriculum. A survey was administered to our seniors, as is summarized below. It should be kept in mind that these responses are from current students, and not practicing professionals, so the validity of the results is uncertain. The survey asked each student to "assess the need for various hardware-oriented topics in an IT program geared to train leaders and professionals" who will "create the next generation of IT systems." The students rated the topics on a scale from 1 to 5, with 5 indicating an essential subject. The topics polled are deliberately rather broad and imprecise, to allow subjects to freely consider hardware curricula in general, without any leading inferences. The results of this poll are shown below.

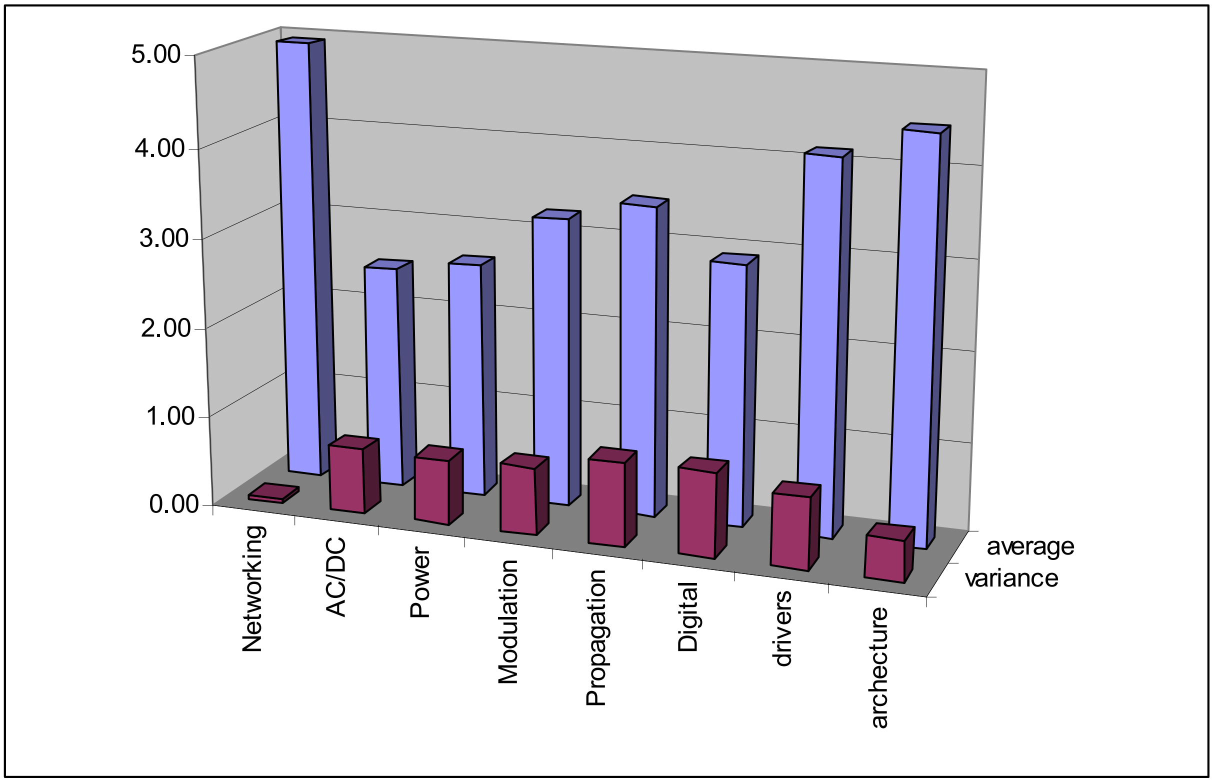

Figure 2 – Students' perceptions of importance of hardware curriculum

It comes as no surprise that the 22 seniors that completed this poll consider networking and computer architecture to be essential to their future. Device drivers and signal-oriented topics such as modulation also have some student support, although weak. Many students see little need for AC/DC concepts, power, and digital circuits, although it can be argued that these are at least needed in order to properly understand other topics listed that had more student support.

From this data, it is apparent that if faculty feel that hardware knowledge is necessary in IT professionals, additional effort needs to be made to explain this to IT students. We look forward to administering this poll to graduates, when we have a group that has the perspective of a few years of industry experience.

Summary

A digital-systems hardware curriculum has been developed that is appropriate for use in IT and related programs. The curriculum is abbreviated, spanning 4 courses for a total of 13 credit hours. Of these hours, 7 are for the core hardware curriculum, including analog and digital electronics. An additional 3 hours are used for communications concepts instruction, and the remaining 3 hours are computer systems oriented, including low-level programming of hardware. The intention of the curriculum is to create IT personnel who are truly digital systems professionals.

When the program matures to the point where there are graduates with several years of experience in industry, the success and utility of this type of systems training with be revisited through the survey process.

References

1. Armitage, William D., Karshmer, Arthur I., Florida's Universities tackle the IT Curriculum, *IT Professional*, September/October 2003, pp. 37-43

2. Ekstrom, J. J., Lunt, Barry M., Education at the Seams: Preparing Students to Stitch Systems Together; Curriculum and Issues for 4-Year IT Programs, *Proceedings of the 4th Conference on Information Technology Curriculum*, Oct. 2003, Lafayette Indiana, pp 196-200

3. Stoll, Clifford, Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage, Pocket Books, Oct. 2000

4. http://www.computerworld.com/departments/opinions/sharktank/0,4885,84442,00.html, 28 August 2003

5. http://www.computer.org/education/cc2001/final/index.htm

6. Bailey, Janet L., Stefaniak, Greg, Preparing the Information Technology Workforce for the New Millennium, *Proceedings of the 2000 ACM SIGCPR conference on Computer personnel research*, Chicago, Illinois, pp 1-7

7. Furse, Cynthia, The Global Wireless Education Consortium: An Innovative Organization; A New Wireless Curriculum, *IEEE Antennas and Propagation Magazine*, Vol. 43 No. 6, Dec. 2001 pp. 156-158

Biography

Michael G. Bailey

Michael G. Bailey has electrical engineering degrees Brigham Young University, the University of Southern California, and the Florida Institute of Technology. Along with 3 years in academia, he has 15 years of experience in the aerospace industry, where he gained a lively interest in Digital Signal Processing and High Performance Computing. Dr. Bailey also enjoys taking his family for adventures in the Utah wilderness.

C. Richard G. Helps

Richard Helps is the Program Chair of the Information Technology program at BYU. He spent ten years in industry as a control systems design engineer. He completed BS and MS degrees at the U of the Witwatersrand, South Africa and a further graduate degree at the University of Utah in Electrical Engineering. His scholarly interests are in embedded and real-time computing with its instrumentation and control aspects and also in technology education.

Barry M. Lunt

Barry M. Lunt is an Associate Professor of Information Technology at Brigham Young University in Provo, UT. Dr. Lunt received a B.S. and an M.S. degree in EET from BYU, and a Ph.D. in Occupational and Adult Education from Utah State University in Logan, UT. He has spent seven years in industry as a design engineer. His present research emphases are the physical design of electronic circuits and systems, IT curriculum, and engineering education.