Computer engineering

From Wikipedia, the free encyclopedia

Computer Engineering (also called **Electronic and Computer Engineering** or **Computer Systems Engineering**) is a discipline that combines both Electrical Engineering and Computer Science.[1] Computer engineers usually have training in electrical engineering, software design and hardware-software integration instead of only software engineering or electrical engineering. Computer engineers are involved in many aspects of computing, from the design of individual microprocessors, personal computers, and supercomputers, to circuit design. This field of engineering not only focuses on how computer systems themselves work, but also how they integrate into the larger picture.[2]

Usual tasks involving computer engineers include writing software and firmware for embedded microcontrollers, designing VLSI chips, designing analog sensors, designing mixed signal circuit boards, and designing operating systems. Computer engineers are also suited for robotics research, which relies heavily on using digital systems to control and monitor electrical systems like motors, communications, and sensors.

Contents

Computer engineering as an academic discipline

The first accredited computer engineering degree program in the United States was established at Case Western Reserve University in 1971; as of October 2004 there were 170 ABET-accredited computer engineering programs in the US.[3]

Due to increasing job requirements for engineers, who can design and manage all forms of computer systems used in industry, some tertiary institutions around the world offer a bachelor's degree generally called computer engineering. Both computer engineering and electronic engineering programs include analog and digital circuit design in their curricula. As with most engineering disciplines, having a sound knowledge of mathematics and sciences is necessary for computer engineers.

In many institutions, computer engineering students are allowed to choose areas of in-depth study in their junior and senior year, as the full breadth of knowledge used in the design and application of computers is well beyond the scope of an undergraduate degree. The joint IEEE/ACM *Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering* defines the core knowledge areas of **computer engineering** as[4]

- Algorithms
- Computer architecture and organization
- Computer systems engineering
- Circuits and signals
- Database systems
- Digital logic

- Digital signal processing
- Electronics
- Embedded systems
- Human-computer interaction
- Interactive Systems Engineering
- Operating systems
- Programming fundamentals
- Social and Professional issues
- Software engineering
- VLSI design and fabrication

The breadth of disciplines studied in computer engineering is not limited to the above subjects but can include any subject found in engineering.

See also

- Association for Computing Machinery
- Electrical engineering
- List of electrical engineers
- Engineer
- Information technology
- Institute of Electrical and Electronics Engineers
- Computer
- Computer science
- Computer-aided software engineering

References

1. ^ IEEE Computer Society; ACM (12 December 2004). *Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*. pp. pg. iii. http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/CCCE-FinalReport-2004Dec12-Final.pdf. Retrieved on 2006-04-21. "Computer engineering has traditionally been viewed as a combination of both cool computer science (CS) and electrical engineering (EE)."
2. ^ Trinity College Dublin. "What is Computer Engineering". http://www.tcd.ie/Engineering/about/what_is_eng/computer_eng_intro.html. Retrieved on 2006-04-21., "Computer Engineers not only to understand how computer systems themselves work, but also how they integrate into the larger picture. Consider the car. A modern car contains many separate computer systems for controlling such things as the engine timing, the brakes and the air bags. To be able to design and implement such a car, the computer engineer needs a broad theoretical understanding of all these various subsystems & how they interact."
3. ^ IEEE Computer Society; ACM (12 December 2004). *Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*. pp. pg. 5. http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/CCCE-FinalReport-2004Dec12-Final.pdf. Retrieved on 2006-04-21. "In the United States, the first computer engineering program accredited by ABET (formerly the Accreditation Board for Engineering and Technology) was at Case Western Reserve University in 1971. As of October 2004, ABET has accredited over 170 computer engineering or similarly named programs."
4. ^ IEEE Computer Society; ACM (12 December 2004). *Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*. pp. pg. 12. http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/CCCE-FinalReport-2004Dec12-Final.pdf. Retrieved on 2006-04-21.

External links

- Computer Engineering Conference Calendar (http://wwwdsa.uqac.ca/~lsr/emcos/emcos-index.php?page=Computer+Engineering+Conference+Calendar)

Categories: Electronic engineering | Computer engineering
Hidden categories: All articles with unsourced statements | Articles with unsourced statements from April 2007

Chapter 2

Computer Engineering as a Discipline

This chapter presents some of the characteristics that distinguish computer engineering from other computing disciplines. It provides some background of the field and shows how it evolved over time. It also highlights some of the characteristics expected from its graduates, preparation for entering the curriculum, and student outcomes and assessment. The chapter also highlights the importance of graduates to have a proper sense of professionalism to ensure a proper perspective in the practice of computer engineering.

2.1 Background

Computer engineering is defined as the discipline that embodies the science and technology of design, construction, implementation, and maintenance of software and hardware components of modern computing systems and computer-controlled equipment. Computer engineering has traditionally been viewed as a combination of both computer science (CS) and electrical engineering (EE). It has evolved over the past three decades as a separate, although intimately related, discipline. Computer engineering is solidly grounded in the theories and principles of computing, mathematics, science, and engineering and it applies these theories and principles to solve technical problems through the design of computing hardware, software, networks, and processes.

Historically, the field of computer engineering has been widely viewed as "designing computers." In reality, the design of computers themselves has been the province of relatively few highly skilled engineers whose goal was to push forward the limits of computer and microelectronics technology. The successful miniaturization of silicon devices and their increased reliability as system building blocks has created an environment in which computers have replaced the more conventional electronic devices. These applications manifest themselves in the proliferation of mobile telephones, personal digital assistants, location-aware devices, digital cameras, and similar products. It also reveals itself in the myriad of applications involving embedded systems, namely those computing systems that appear in applications such as automobiles, large-scale electronic devices, and major appliances.

Increasingly, computer engineers are involved in the design of computer-based systems to address highly specialized and specific application needs. Computer engineers work in most industries, including the computer, aerospace, telecommunications, power production, manufacturing, defense, and electronics industries. They design high-tech devices ranging from tiny microelectronic integrated-circuit chips, to powerful systems that utilize those chips and efficient telecommunication systems that interconnect those systems. Applications include consumer electronics (CD and DVD players, televisions, stereos, microwaves, gaming devices) and advanced microprocessors, peripheral equipment, systems for portable, desktop and client/server computing, and communications devices (cellular phones, pagers, personal digital assistants). It also includes distributed computing environments (local and wide area networks, wireless networks, internets, intranets), and embedded computer systems (such as aircraft, spacecraft, and automobile control systems in which computers are embedded to perform various functions). A wide array of complex technological systems, such as power generation and distribution systems and modern processing and manufacturing plants, rely on computer systems developed and designed by computer engineers.

Technological advances and innovation continue to drive computer engineering. There is now a convergence of several established technologies (such as television, computer, and networking technologies) resulting in widespread and ready access to information on an enormous scale. This has created many opportunities and challenges for computer engineers. This convergence of technologies and the associated innovation lie at the heart of economic development and the future of many organizations. The situation bodes well for a successful career in computer engineering.

Chapter 4

Overview of the Computer Engineering Body of Knowledge

Developing any curriculum for undergraduate study in computer engineering should reflect the current needs of computer engineering students. The curriculum should also reflect current educational practice and suggest improvements where necessary. The discussion that follows attempts to accomplish this in preparing a body of knowledge commensurate with producing competent computer engineering graduates.

4.1 The Body of Knowledge

The Computer Engineering Task Force has sought to assemble a modern curriculum by first defining the primary disciplines that make up the body of knowledge for computer engineering. Some of these discipline areas contain material that should be part of *all* computer engineering curricula. These are the 18 knowledge areas, including two covering related mathematics topics, listed in Table 4.1. Other areas contain material that might, or might not, be part of such curricula, depending on the specific educational objectives of a program. Some of these are listed in Chapter 7, but are not described in detail in this report.

Table 4.1
CE2004 Discipline Areas Containing Core Material

CE-ALG*	Algorithms
CE-CAO	Computer Architecture and Organization
CE-CSE	Computer Systems Engineering
CE-CSG	Circuits and Signals
CE-DBS	Database Systems
CE-DIG	Digital Logic
CE-DSP	Digital Signal Processing
CE-ELE	Electronics
CE-ESY	Embedded Systems
CE-HCI*	Human-Computer Interaction
CE-NWK	Computer Networks
CE-OPS*	Operating Systems
CE-PRF*	Programming Fundamentals
CE-SPR*	Social and Professional Issues
CE-SWE*	Software Engineering
CE-VLS	VLSI Design and Fabrication
----------	------------------------------------
CE-DSC*	Discrete Structures
CE-PRS	Probability and Statistics

* Consult the CC2001 Computer Science report for more detail

After defining the above areas, each task force member designed and reviewed initial drafts defining the body of knowledge for one or more areas. In some cases, new members joined the task force to cover areas of expertise outside of those originally represented. Subsequently, a second task force member reviewed and revised each initial draft. After each revision, the entire task force reviewed the resulting draft for comment. At the completion of this process, the entire task force met as a group to review the draft body of knowledge, with follow-up modifications made as appropriate.

Table 4.3
The Computer Engineering Body of Knowledge

Computer Engineering	*Knowledge Areas and Units*
CE-ALG Algorithms [30 core hours] CE-ALG0 History and overview [1] CE-ALG1 Basic algorithmic analysis [4] * CE-ALG2 Algorithmic strategies [8] * CE-ALG3 Computing algorithms [12] * CE-ALG4 Distributed algorithms [3] * CE-ALG5 Algorithmic complexity [2] * CE-ALG6 Basic computability theory *	**CE-CAO Computer Architecture and Organization** [63 core hours] CE-CAO0 History and overview [1] CE-CAO1 Fundamentals of computer architecture [10] CE-CAO2 Computer arithmetic [3] CE-CAO3 Memory system organization and architecture [8] CE-CAO4 Interfacing and communication [10] CE-CAO5 Device subsystems [5] CE-CAO6 Processor systems design [10] CE-CAO7 Organization of the CPU [10] CE-CAO8 Performance [3] CE-CAO9 Distributed system models [3] CE-CAO10 Performance enhancements
CE-CSE Computer Systems Engineering [18 core hours] CE-CSE0 History and overview [1] CE-CSE1 Life cycle [2] CE-CSE2 Requirements analysis and elicitation [2] CE-CSE3 Specification [2] CE-CSE4 Architectural design [3] CE-CSE5 Testing [2] CE-CSE6 Maintenance [2] CE-CSE7 Project management [2] CE-CSE8 Concurrent (hardware/software) design [2] CE-CSE9 Implementation CE-CSE10 Specialized systems CE-CSE11 Reliability and fault tolerance	**CE-CSG Circuits and Signals** [43 core hours] CE-CSG0 History and overview [1] CE-CSG1 Electrical Quantities [3] CE-CSG2 Resistive Circuits and Networks [9] CE-CSG3 Reactive Circuits and Networks [12] CE-CSG4 Frequency Response [9] CE-CSG5 Sinusoidal Analysis [6] CE-CSG6 Convolution [3] CE-CSG7 Fourier Analysis CE-CSG8 Filters CE-CSG9 Laplace Transforms
CE-DBS Database Systems [5 core hours] CE-DBS0 History and overview [1] CE-DBS1 Database systems [2] * CE-DBS2 Data modeling [2] * CE-DBS3 Relational databases * CE-DBS4 Database query languages * CE-DBS5 Relational database design * CE-DBS6 Transaction processing * CE-DBS7 Distributed databases * CE-DBS8 Physical database design *	**CE-DIG Digital Logic** [57 core hours] CE-DIG0 History and overview [1] CE-DIG1 Switching theory [6] CE-DIG2 Combinational logic circuits [4] CE-DIG3 Modular design of combinational circuits [6] CE-DIG4 Memory elements [3] CE-DIG5 Sequential logic circuits [10] CE-DIG6 Digital systems design [12] CE-DIG7 Modeling and simulation [5] CE-DIG8 Formal verification [5] CE-DIG9 Fault models and testing [5] CE-DIG10 Design for testability
CE-DSP Digital Signal Processing [17 core hours] CE-DSP0 History and overview [1] CE-DSP1 Theories and concepts [3] CE-DSP2 Digital spectra analysis [1] CE-DSP3 Discrete Fourier transform [7] CE-DSP4 Sampling [2] CE-DSP5 Transforms [2] CE-DSP6 Digital filters [1] CE-DSP7 Discrete time signals CE-DSP8 Window functions CE-DSP9 Convolution CE-DSP10 Audio processing CE-DSP11 Image processing	**CE-ELE Electronics** [40 core hours] CE-ELE0 History and overview [1] CE-ELE1 Electronic properties of materials [3] CE-ELE2 Diodes and diode circuits [5] CE-ELE3 MOS transistors and biasing [3] CE-ELE4 MOS logic families [7] CE-ELE5 Bipolar transistors and logic families [4] CE-ELE6 Design parameters and issues [4] CE-ELE7 Storage elements [3] CE-ELE8 Interfacing logic families and standard buses [3] CE-ELE9 Operational amplifiers [4] CE-ELE10 Circuit modeling and simulation [3] CE-ELE11 Data conversion circuits CE-ELE12 Electronic voltage and current sources CE-ELE13 Amplifier design CE-ELE14 Integrated circuit building blocks
CE-ESY Embedded Systems [20 core hours] CE-ESY0 History and overview [1] CE-ESY1 Embedded microcontrollers [6] CE-ESY2 Embedded programs [3] CE-ESY3 Real-time operating systems [3] CE-ESY4 Low-power computing [2] CE-ESY5 Reliable system design [2] CE-ESY6 Design methodologies [3] CE-ESY7 Tool support CE-ESY8 Embedded multiprocessors CE-ESY9 Networked embedded systems CE-ESY10 Interfacing and mixed-signal systems	**CE-HCI Human-Computer Interaction** [8 core hours] CE-HCI0 History and overview [1] CE-HCI1 Foundations of human-computer interaction [2] * CE-HCI2 Graphical user interface [2] * CE-HCI3 I/O technologies [1] * CE-HCI4 Intelligent systems [2] * CE-HCI5 Human-centered software evaluation * CE-HCI6 Human-centered software development * CE-HCI7 Interactive graphical user-interface design * CE-HCI8 Graphical user-interface programming * CE-HCI9 Graphics and visualization * CE-HCI10 Multimedia systems *

CE-NWK Computer Networks [21 core hours] CE-NWK0 History and overview [1] CE-NWK1 Communications network architecture [3] CE-NWK2 Communications network protocols [4] CE-NWK3 Local and wide area networks [4] CE-NWK4 Client-server computing [3] CE-NWK5 Data security and integrity [4] CE-NWK6 Wireless and mobile computing [2] CE-NWK7 Performance evaluation CE-NWK8 Data communications CE-NWK9 Network management CE-NWK10 Compression and decompression	**CE-OPS Operating Systems** [20 core hours] CE-OPS0 History and overview [1] CE-OPS1 Design principles [5] * CE-OPS2 Concurrency [6] * CE-OPS3 Scheduling and dispatch [3] * CE-OPS4 Memory management [5] * CE-OPS5 Device management * CE-OPS6 Security and protection * CE-OPS7 File systems * CE-OPS8 System performance evaluation *
CE-PRF Programming Fundamentals [39 core hours] CE-PRF0 History and overview [1] CE-PRF1 Programming Paradigms [5] * CE-PRF2 Programming constructs [7] * CE-PRF3 Algorithms and problem-solving [8] * CE-PRF4 Data structures [13] * CE-PRF5 Recursion [5] * CE-PRF6 Object-oriented programming * CE-PRF7 Event-driven and concurrent programming * CE-PRF8 Using APIs *	**CE-SPR Social and Professional Issues** [16 core hours] CE-SPR0 History and overview [1] CE-SPR1 Public policy [2] * CE-SPR2 Methods and tools of analysis [2] * CE-SPR3 Professional and ethical responsibilities [2] * CE-SPR4 Risks and liabilities [2] * CE-SPR5 Intellectual property [2] * CE-SPR6 Privacy and civil liberties [2] * CE-SPR7 Computer crime [1] * CE-SPR8 Economic issues in computing [2] * CE-SPR9 Philosophical frameworks *
CE-SWE Software Engineering [13 core hours] CE-SWE0 History and overview [1] CE-SWE1 Software processes [2] * CE-SWE2 Software requirements and specifications [2] * CE-SWE3 Software design [2] * CE-SWE4 Software testing and validation [2] * CE-SWE5 Software evolution [2] * CE-SWE6 Software tools and environments [2] * CE-SWE7 Language translation * CE-SWE8 Software project management * CE-SWE9 Software fault tolerance *	**CE-VLS VLSI Design and Fabrication** [10 core hours] CE-VLS0 History and overview [1] CE-VLS1 Electronic properties of materials [2] CE-VLS2 Function of the basic inverter structure [3] CE-VLS3 Combinational logic structures [1] CE-VLS4 Sequential logic structures [1] CE-VLS5 Semiconductor memories and array structures [2] CE-VLS6 Chip input/output circuits CE-VLS7 Processing and layout CE-VLS8 Circuit characterization and performance CE-VLS9 Alternative circuit structures/low power design CE-VLS10 Semi-custom design technologies CE-VLS11 ASIC design methodology

Mathematics Knowledge Areas and Units	
CE-DSC Discrete Structures [33 core hours] CE-DSC0 History and overview [1] CE-DSC1 Functions, relations, and sets [6] * CE-DSC2 Basic logic [10] * CE-DSC3 Proof techniques [6] * CE-DSC4 Basics of counting [4] * CE-DSC5 Graphs and trees [4] * CE-DSC6 Recursion [2] *	**CE-PRS Probability and Statistics** [33 core hours] CE-PRS0 History and overview [1] CE-PRS1 Discrete probability [6] CE-PRS2 Continuous probability [6] CE-PRS3 Expectation [4] CE-PRS4 Stochastic Processes [6] CE-PRS5 Sampling distributions [4] CE-PRS6 Estimation [4] CE-PRS7 Hypothesis tests [2] CE-PRS8 Correlation and regression

* Consult the CC2001 Report [ACM/IEEECS, 2001] for more detail on these knowledge units