A Novel Approach to an Embedded Systems Curriculum

Tyson S. Hall, Jared Bruckner, and Richard L. Halterman
School of Computing, Southern Adventist University
Collegedale, Tennessee 37315–0370, {tyson,bruckner,halterman}@southern.edu

***Abstract*— Universities around the world have modified their computer engineering curricula to include courses and modules focusing on the design of embedded systems including reconfigurable hardware, real-time processing, embedded processors, and firmware/software systems. This paper describes a different approach to embedded systems education—the development of an embedded systems track within an existing computer science program. This approach allows for an intentionally designed balance of computer science, computer engineering, mathematics, and science courses deemed necessary for a comprehensive embedded systems program.**

Index Terms—computer engineering education, computer science education, embedded systems, multi-disciplinary

EMBEDDED SYSTEMS EDUCATION

The field of embedded systems is a popular area of study for students and educators alike. Evidence for the popularity of embedded systems in academia is seen in the southeast where seventy percent of ABET-accredited computer engineering programs include a course focusing on embedded systems. Embedded systems courses are being integrated into computer science [1], electrical and computer engineering [2], [3], software engineering programs [4], and cross-disciplinary courses [5]. Current embedded systems courses span a number of topics including designing real-time systems [6], [7], low-power systems that contain reconfigurable hardware (FPGAs and FPAAs) [8]–[10], embedded processors, firmware/software [11], and digital signal processing [12]. Pedagogically, these courses allow students to explore the trade-offs between different design modalities (i.e., hardware versus software, digital versus analog circuitry, dedicated versus reconfigurable hardware, etc.) and investigate optimal solutions given a set of constraints placed on the system. Some novel courses are leveraging these multi-discipline characteristics of embedded systems to create unique project design experiences [13], [14]. While important to students' development, these individual courses cannot be expected to fully prepare students for careers in this new embedded systems discipline. A more extensive, program-level curriculum is needed to adequately educate students in embedded systems design.

Computer engineering programs at universities around the world have been modified to implement embedded systems courses and integrate multimodal design concepts throughout their curriculum [15], [16]. Several comptuer engineering programs include embedded systems concentrations [17]–[20], and only two programs are currently known to offer a complete four-year curriculum in embedded systems [21], [22]. Southern Adventist University has taken a non-traditional approach by implementing an embedded systems program track in the computer science program without the existence of a computer engineering program on campus. The novelty of this program curricula is found in its integration into the Computer Science degree program, nearly equal treatment of hardware and software topics, and implementation in a 124 semester hour degree program. This paper introduces the four-year embedded systems curriculum at Southern. The computer science heritage of this program along with Southern's strong liberal arts tradition are viewed as strengthening the embedded systems program and diminishing the tension between technical program requirements and university general education requirements.

General Education Area	Sem. Hours
Writing & Communication	9
History, Political, and Economic Systems	9
Religion	12
Language, Literature, Fine Arts	6
Behavioral, Family, Health Sciences	5
Recreational Activities	2

TABLE I
SUMMARY OF GENERAL EDUCATION REQUIREMENTS AT SOUTHERN ADVENTIST UNIVERSITY (IN SEMESTER HOURS)

ACADEMIC ENVIRONMENT

Southern Adventist University (Southern) is a private, accredited co-educational institution offering masters, baccalaureate, and associate degrees. The student body is approximately 2500 students with a variety of religious backgrounds although ninety-four percent are Seventh-day Adventists. Southern is located in southeast Tennessee just east of Chattanooga.

I. *The University*

Southern has a ninety-year history of liberal arts education. Although sixty percent of students on Southern's campus today major in areas other than those considered as liberal arts, this heritage is seen in the strong general education requirements still maintained by the univeristy. Computer science and embedded systems students are required to complete 43 semester hours of general education courses in addition to their major and cognate courses. As shown in Table I, students

Graduates will

1) Professionally practice in the field of computing or pursue advanced study in/using computing.
2) Demonstrate a computer scientist's professional responsibility to God, church, family, employer, colleagues, and society.
3) Effectively communicate in technical environments to increase knowledge, understanding, and professional awareness.
4) Possess a thorough grounding in the principles and practices of computing and apply this knowledge in professional practice including the design, implementation, and problem solving of practical software and/or hardware systems.
5) Pursue life-long learning including embracing new computing technologies, continuing professional development, and remaining active in the computing discipline.
6) Work effectively in both independent and team environments.

Fig. 1. These six program objectives are for the BS in Computer Science at Southern Adventist University.

receive a broad foundation in communication, history, religion, literature, and pyschology.

When designing any new degree program at Southern, consideration must be given to the strong liberal arts component. One can view this as a limitation on what can be taught within the major field, or it can be viewed as a very important part of an undergraduate's well-rounded education. While the authors take the latter view, the size of the general education component does constrain the number of hours that can be required within the major if students are expected to complete a Bachelor of Science (BS) degree in four years. Since all BS programs on campus keep total required course hours under 124 semester hours, some trade-offs must be made when designing the embedded systems degree program. In particular, the number of computer science courses, hardware/engineering courses, and math and science cognates must be constrained to keep the embedded system program track within the realm of a four-year BS degree.

II. The School of Computing

The School of Computing at Southern Adventist University offers four different undergraduate degrees spanning the areas of Computer Science, Computer Systems Administration, and Computer Information Systems. In 2003, the School embarked on the journey towards ABET accrediation of the Bachelor of Science (BS) in Computer Science degree. For the past several years, the School has undergone the necessary transformation in curriculum and assessment/evaluation processes to make this program accreditable. This environment of change and the possibility to have the embedded systems program seek accreditation along with the traditional Computer Science program has made the BS in Computer Science degree an ideal home for the embedded systems program.

There are six established program objectives for the BS in Computer Science program as shown in Fig. 1. These objectives are sufficient broad to as to be satisified by both the traditional Computer Science program and new embedded systems program.

DEVELOPING THE EMBEDDED SYSTEMS PROGRAM

Developing an embedded systems program from the ground up requires a number of different pedagogical and logistical discussions. In this section, the placement of the program within the university's organizational structure (is it housed in the engineering or computer science department?) is addressed. Next, the curricular structure and marketability of the resulting degree are discussed.

III. Engineering versus Science

The nature of an embedded systems program places it at the crossroads of electrical/computer engineering and computer science curricula. At the outset of planning a new program, three primary options can be considered:

1) Creating a computer engineering degree oriented towards embedded systems design,
2) Creating a BS in Embedded Systems degree, and
3) Creating a new program track to the existing BS in Computer Science degree.

Starting a computer engineering program at Southern was the most problematic option. Given the liberal art tradition and academic/political climate of the university, an engineering degree would be difficult to establish. Furthermore, the addition (and resulting expense) of multiple engineering faculty would meet resistance in administration. Also, accreditation would be complicated, because ABET accreditation from both EAC and CAC would be required to encompass the computer engineering and computer science degrees.

Adding a BS in Embedded Systems degree was deemed possible. Here, the cost of marketing a degree with a non-traditional name to both prespective students and employers/recruiters would have been high given that most current embedded systems programs have been implemented as concentrations within more traditional programs such as computer engineering or computer science. Employers are familiar with these programs and more accepting of concentrations within them. Establishing a separate embedded systems degree would, at least initially, have proved troublesome for graduates during their job search, because recruiters might not have immediately recognized or understood the degree program. Accreditation would also have been complicated since this degree would have been a standalone program.

The most natural home for the embedded systems program at Southern was within the BS Computer Science degree program. Since the computer science program was already well established at Southern, the cost and effort of implementing a new program track was minimized. Enhancing an established program was viewed very positively by the university administration whereas establishing a new, possibly competing degree program was veiwed skeptically. There were enough current computer science faculty to cover the courses that were common between the two program tracks and classroom sizes were able to handle the increased enrollment. Initially, one engineering faculty member was added with responsibility for the new courses specifically needed for the embedded systems program track.

IV. Learning Paradigm

Previous work available in engineering education literature has established the benefits cooperative and problem-based learning techniques [23]–[25]. The embedded systems program at Southern is designed to emphasize hands-on, problem-based learning. Every new course that has been added for the embedded systems program track is a combined lecture/laboratory course. In addition, course lectures are being designed from the very beginning to take advantage of active, cooperative learning and problem-based learning techniques.

To support this laboratory-intensive curriculum, a modern engineering laboratory has been created. Initially, sufficient laboratory space has been acquired by reorganizing current computer laboratories. The new embedded systems laboratory is comprised of five workstations. Most laboratory projects are completed in teams of two, allowing initial class sizes of 10 students. Additional laboratory workbenches can be added as student enrollment increases.

The laboratory has been stocked with state-of-the-art test and instrumentation equipment, standard laboratory supplies including a varity of workbench cables, common IC/discrete parts, and instructional robotics. To fund the initial acquisition of equipment for this laboratory, $125,000 was raised through internal university grants, corporate equipment grants, and private donations. The School of Computing also added $25,000 from its budget to complete the laboratory.

V. Ethics and Soft Sciences

The significant general education requirements at Southern provide students with a rich foundation in writing, public speaking, ethics, theology, history, and pyschology as shown in Table I. In addition, computer science students are also required to take a senior-level course taught by the computing faculty that integrates writing, public speaking, and ethics with the computing profession. This capstone class seeks to bridge students' liberal arts grounding with their technical/professional education. In this course, students write conference-style research papers, give technical lectures, present a research poster at a departmental poster session/contest, and lead in-depth class discussions of professional ethics, intellectual property, and privacy.

VI. After Graduation

One of the objectives for students leaving Southern with a Computer Science degree (see previous section) is for them to be able to enter professional practice. In general, there are an increasing number of jobs in the embedded systems arena. Additionally, the political and economic climate is shifting in southeast Tennessee. The greater Chattanooga community is actively courting high-tech manufacturing companies, and community leaders are advocating a more technology-driven local economy [26]–[29]. Also, Southern is centrally located between Oak Ridge, TN; Atlanta, GA; Huntsville, AL; and Research Triangle Park, NC—all locations with technology-rich economies.

It is also important to the faculty in the School of Computing to design the embedded systems program to lead directly to graduate study. Because of the dual nature of this program, it is ideal if graduates have the flexibility in choosing a computer science, computer engineering, or software engineering graduate program. The embedded system curriculum has been designed to lead into these graduate programs with a minimum number of additional prerequisites. Faculty advisors also work with students throughout their program of study to select elective courses that satisfy the various graduate program prerequisites.

There is no uniform admission requirements for masters degrees in electrical and computer engineering in terms of specific undergraduate coursework and/or number of required courses. Since the majority of Southern's students originate from the southeastern United States, several graduate electrical and computer engineering programs in this region have been used as case studies to determine student eligibility in graduate engineering programs. Graduate programs with designated computer engineering degrees are the most accessible engineering programs to graduates of Southern's embedded systems program. Students entering many of these masters-level programs will need three or fewer prerequisite courses [30], [31]. In most cases, additional prerequsities might be needed in a student's major area of concentration [31], [32]. Embedded systems students might pursue graduate study in the broader field of electrical engineering; however, admission to a Master in Electrical Engineering degree program could require as much as an additional year of prerequisite coursework [33].

Since embedded systems is a program track within the computer science major, a graduate can enter most computer science masters degree programs with few missing prerequisites. An examination of several representative graduate programs [34]–[38] confirms this expectation. Courses in theory of computation, algorithms, and programming languages are frequently listed as undergraduate prerequisites, but these are not part of the embedded systems program. Of the schools surveyed, one requires all three courses [36], two require two of the three [35], [37], another requires only one of the three [34], and one did not require any of the three [38]. In some graduate schools, one or two missing prerequisites can be taken as part of the masters coursework, so, depending on the institution of choice, an embedded systems graduate should be able to enter a computer science masters program with a two or fewer additional prerequisite undergraduate courses.

An embedded system graudate needs no additional course work to meet the admission requirements of most Master of Software Engineering (MSE) degree programs [39]–[42]. A common admissions requirement to MSE programs is work experience with large software projects [39], [40], which may be easier for a traditional computer science major to obtain. However, other MSE programs do not have such a requirement [41], [42].

PROGRAM CURRICULA

The embedded system program has been added to the BS in Computer Science degree program. This action resulted in a number of efficiencies in terms of faculty utilization, course requirements, and marketing (see Table II for the existing Computer Science curriculum). In designing the Computer Science–Embedded Systems program track, the majority of the first two-year Computer Science sequence was left intact as shown in Table III. This provided the necessary background in programming and software design required of embedded systems students.

In addition to the initial two-year course sequence, embedded systems students must also take Introduction to Engineering (already offered for students in the two-year pre-engineering program) and Digital Logic & Design. It should also be noted that embedded systems students register for CPHE 220: Computer Architecture instead of CPTR 220: Organization, Architecture, & Assembly Language. These courses are cross-listed, however, and differ only in the laboratory assignments given to each class. Therefore, the offering of this course does not require additional faculty hours.

The majority of the curricular differences between the Computer Science and Computer Science–Embedded Systems tracks occur in the junior and senior year sequences. Upper-division computer science courses and electives are replaced by hardware and embedded systems courses in the embedded systems track. Students are required to take Introduction to Signal Processing, Circuit Analysis, and Microcontroller Design, which are traditional electrical and computer engineering core curricula. The topical coverage for these courses is equivalent to their traditional computer engineering counterparts; however, design projects, homework assignments, and classroom examples are heavily based on embedded systems to create a unifying theme throughout the curriculum. To round out their computer science education, students must also take Principles of Networking, Operating Systems, and Senior Seminar. In addition, embedded systems students must take the senior-level Computer Interfacing course. This course integrates the concepts from both their computer science and computer hardware courses into a capstone design experience emphasizing embedded systems and hardware-software co-design.

The cognate requirements for the two tracks are similar; however, fifteen semester hours of the elective math and science courses in the computer science track are specified cognate requirements in the embedded system track. To stay within the generally accepted degree requirements at Southern, limitations were set on the total number of required cognate hours. These trade-offs primarily affected the math and science cognate requirements. Ideally, twelve to fifteen hours of additional math and science courses would be added to this program including Calculus III, Calculus Applications of Physics, and the second semester of a second general science sequence. While these courses are not required at this time, faculty advisors do encourage students to select these courses as a part of their free electives, especially for students considering graduate study in engineering.

ABET criteria for engineering programs requires at least 45 semester hours of engineering courses. The embedded systems curriculum must divide these technical hours between both computer science and computer engineering courses. Since the total degree hours could not be increased substantially and still fit within a reasonable four-year degree program, trade-offs in engineering versus computer science courses are required. On the computer science side, courses in database management and software engineering are not included in the required curriculum. On the engineering side, specific courses in microelectronic circuits, VLSI design, and advanced topics in one or more technical interest areas (i.e., advanced digital signal processing, computer architecture, IC design, etc.) are not required. While there are merits to including each of these courses in the curriculum, the nature of embedded systems requires a breadth of traditional coursework with sufficient depth needed in system-level design. Additional topical depth can be acquired in many different technical interest areas at the graduate level, and students with an affinity for a particular topic are encouraged to pursue graduate study in their areas of interest.

Required Courses		**Hours**
CPTR 103	Principles of Computing	3
CPTR 124	Fundamentals of Programming	4
CPTR 215	Fundamentals of Software Design	4
CPTR 220	Organization, Arch. & Assembly Language	4
CPTR 314	Data Structrues, Alg. & Know. Systems	4
CPTR 319	Database Management Systems	3
CPTR 365	Operating Systems	3
CPTR 405	Organization of Prog. Languages	3
CPTR 486	Senior Seminar	2
SENG 209	Intro. to Software Engineering	4
	Computer Science Electives	13

Required Cognates		**Hours**
MATH 181	Calculus I	3
MATH 182	Calculus II	4
MATH 200	Elementary Linear Algebra	2
MATH 215	Statistics	3
MATH 280	Discrete Mathematical Structures	3
	Two-Semester Science Sequence with Lab	8
	Approved Science Electives	4
	Approved Math or Science Elective	3

TABLE II

SUMMARY OF THE EXISTING COMPUTER SCIENCE CURRICULUM (IN SEMESTER HOURS)

CONCLUSION

The Computer Science–Embedded Systems program track provides an almost equal share of engineering and computer science courses. In addition, a solid foundation in math and science is required along with a strong background in communication, ethics, and the humanities. This degree provides students with a diverse educational experience. It also provides both the academic rigor to prepare students for advanced study in computer science or engineering and the practical skills to be productive in an industry environment upon graduation.

References

[1] D. L. Tarnoff, "Incorporating embedded system design into a CS curriculum," in *Proc. of the 2nd Annual Conference on Mid-South College Computing*, April 2004, pp. 131–140.

[2] J. O. Hamblen, "Using an FPGA–based SOC approach for senior design projects," in *Proc. of the International Conference on Microelectronic Systems Education*, 2003, pp. 18–19.

[3] F. Vahid, "Embedded system design: Ucr's undergraduate three-course sequence," in *Proc. of the 2003 IEEE International Conference on Microelectronic Systems Education (MSE'03)*, 2003.

[4] D. L. Tarnoff, "A CS/SE approach to a real-time embedded systems software development course," in *Proc. of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Feb. 2001, pp. 278–281.

[5] I. Verbauwhede and P. Schaumont, "Skiing the embedded systems mountain," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 3, pp. 529–548, Aug. 2005.

[6] M. W. Mutka and A. Bakic, "Teaching undergraduate computer science and engineering students techniques for the design and analysis of real-time applications," in *Proc. of the 28th Annual Frontiers in Education Conference*, vol. 3, Nov. 1998, pp. 1079–1084.

[7] M. J. Wirthlin, "Senior-level embedded system design project using fpgas," in *Proc. of the 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05)*, 2005.

[8] S. Merchant, G. D. Peterson, and D. Bouldin, "Improving embedded systems education: Laboratory enhancements using programmable systems on chip," in *Proc. of the 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05)*, 2005.

[9] K. Newman, J. O. Hamblen, and T. S. Hall, "An introductory digital design course using a low–cost autonomous robot," *IEEE Transactions on Education*, vol. 45, no. 3, pp. 289–296, Aug. 2002.

[10] T. S. Hall and P. Hasler, "Field–programmable analog arrays enable analog signal processing education," in *Proc. of the 2005 American Society for Engineering Education Southeastern Section Annual Meeting*, Chattanooga, TN, April 2005.

[11] J. W. McCormick, "We've been working on the railroad: a laboratory for real-time embedded systems," in *Proc. of the 36th SIGCSE Technical Symposium on Computer Science Education*, Feb. 2005, pp. 530–534.

[12] T. S. Hall and D. V. Anderson, "A framework for teaching real–time digital signal processing with field–programmable gate arrays," *IEEE Transactions on Education*, vol. 48, no. 3, Aug. 2005.

[13] J. W. Bruce, J. C. Harden, and R. B. Reese, "Cooperative and progressive design experience for embedded systems," *IEEE Transactions on Education*, vol. 47, no. 1, pp. 83–92, Feb. 2004.

[14] R. C. Hsu and W. C. Liu, "Project based learning as a pedagogical tool for embedded system education," in *Proc. of the 3rd International Conference on Information Technology: Research and Education*, June 2005, pp. 362–366.

[15] P. Koopman, H. Choset, R. Gandhi, B. Krogh, D. Marculescu, P. Narasimhan, J. M. Paul, R. Rajkumar, D. Siewiorek, A. Smailagic, P. Steenkiste, D. E. Thomas, and C. Wang, "Undergraduate embedded system education at carnegie mellon," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 3, pp. 500–528, Aug. 2005.

[16] A. L. Sangiovanni-Vincentelli and A. Pinto, "An overview of embedded system design education at berkeley," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 3, pp. 472–499, Aug. 2005.

[17] J. Sztipanovits, G. Biswas, K. Frampton, A. Gokhale, L. Howard, G. Karsai, T. J. Koo, X. Koutsoukos, and D. C. Schmidt, "Introducing embedded software and systems education and advanced learning in an engineering curriculum," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 3, pp. 549–568, Aug. 2005.

[18] *Computer engineering technical area options*, [Online], University of Texas at Austin, http://www.ece.utexas.edu/undergrad/cetech_all.html, March 2006.

[19] *Undergraduate degree program*, [Online], University of North Texas, http://www.cse.unt.edu/education/degree_programs.php, March 2006.

[20] G. C. Gannod, F. Golshani, B. Huey, Y.-H. Lee, S. Panchanathan, and D. Pheanis, "A consortium-based model for the development of a concentration track in embedded systems," in *Proc. of the 2002 American Society for Engineering Education Annual Conference and Exposition*, 2002.

[21] R. E. Seviora, "A curriculum for embedded system engineering," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 3, pp. 569–586, Aug. 2005.

[22] B. Haberman and M. Trakhtenbrot, "An undergraduate program in embedded systems engineering," in *Proc. of the 18th Conference on Software Engineering and Training (CSEET'05)*, 2005.

[23] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer, "Engineering design thinking, teaching, and learning," *Journal of Engineering Education*, vol. 94, pp. 103–120, Jan. 2005.

[24] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, Jan. 2005.

[25] K. A. Smith, S. D. Sheppard, D. W. Johnson, and R. T. Johnson, "Pedagogies of engagement: classroom-based practices," *Journal of Engineering Education*, vol. 94, no. 1, pp. 87–101, Jan. 2005.

[26] *Our mission*, [Online], http://www.chatc.org/index.php?page=mission, March 2006.

[27] *Charting Chattanooga's technology potential*, [Online], http://www.chattanoogachamber.com/newsandvideo/trend_november_03_tech_potential.asp, March 2006.

[28] *An award for innovation*, [Online], http://epaper.ardemgaz.com/Repository/getFiles.asp?Style=OliveXLib:ArticleToMail&Type=text/html&Path=ChatTFPress/2004/06/07&ID=Ar01301, June 2004.

[29] *Top ten reasons the Tennessee Valley Corridor is one of America's top ten technology hot spots*, [Online], http://www.tennvalleycorridor.org/index.html, March 2006.

[30] *Graduate admissions requirements for computer engineering*, [Online], Clemson University, http://www.ece.clemson.edu/ece/gradadmreq.shtml, March 2006.

[31] *Articulation requirements*, [Online], University of Florida, http://www.ece.ufl.edu/graduate/admission/articulation.html, March 2006.

[32] *Admission requirements*, [Online], Georgia Tech, http://www.ece.gatech.edu/academics/graduate/apply.html, March 2006.

[33] *Master of Science in Computer or Electrical Engineering Graduate Handbook*, [Online], University of Tennessee, Knoxville, http://www.ece.utk.edu/grad/handbook/ms_ee.html, March 2006.

[34] *Information for Graduate Majors*, [Online], University of Tennessee, Knoxville, http://www.cs.utk.edu/academics/graduate/curriculum/handout/2005/07/handout.pdf, March 2006.

[35] *Graduate Program Prerequisites*, [Online], University of Alabama at Birmingham, http://www.cis.uab.edu/graduate/prereq.php, March 2006.

[36] *Graduate Admissions Letter*, [Online], Georgia State University, http://www.cs.gsu.edu/degrees/grad/info.pdf, March 2006.

[37] *Requirements for the M.S. Degree in CS*, [Online], Vanderbilt University, http://eecs.vuse.vanderbilt.edu/programs/cs_masters.html, March 2006.

[38] *Graduate Admission Procedures and Standards*, [Online], Florida State University, http://www.cs.fsu.edu/prospective/grad/admissions.php, March 2006.

[39] *Admission policy*, [Online], http://www.omse.org/degree_admission/ policy.htm, March 2006.

[40] *Admission requirements*, [Online], Carnegie Mellon University, http://www.mse.cs.cmu.edu/Admission-Requirements.html, March 2006.

[41] *Software engineering foundation requirements*, [Online], George Mason University, http://ise.gmu.edu/ms-swe/swe-found.html, March 2006.

[42] *Computer science and software engineering graduate pre-requisites*, [Online], Auburn University, http://eng.auburn.edu/programs/comp/programs/graduate/pre-requisite.html, March 2006.

Required Courses		**Hours**
ENGR 121	Intro to Engineering	1
CPTR 103	Principles of Computing	3
CPTR 124	Fundamentals of Programming	4
CPTR 215	Fundamentals of Software Design	4
CPHE 200	**Digital Logic & Design**	**4**
CPHE 220**	Computer Architecture	4
CPTR 314	Data Structrues, Alg. & Know. Systems	4
CPHE 310	**Intro to Signal Processing**	**3**
CPHE 320	**Circuit Analysis**	**4**
CPTR 328	Principles of Networking	3
CPTR 365	Operating Systems	3
CPHE 380	**Microcontroller Design**	**4**
CPHE 410	**Computer Interfacing**	**4**
CPTR 486	Senior Seminar	2

Required Cognates		**Hours**
MATH 181	Calculus I	3
MATH 182	Calculus II	4
MATH 200	Elementary Linear Algebra	2
MATH 215	Statistics	3
MATH 280	Discrete Mathematical Structures	3
PHYS 21x	General Physics with Lab	8
	Approved General Science Elective with Lab	4
MATH 319	Differential Equations	3

* ***Bold*** *text denotes new courses added for the embedded systems track.*
** *CPHE 220 is cross-listed with CPTR 220. Lectures are co-taught, but laboratory assignments differ.*

TABLE III

SUMMARY OF THE NEW COMPUTER SCIENCE–EMBEDDED SYSTEMS CURRICULUM (IN SEMESTER HOURS)